Concepts and Methods from Artificial Intelligence in Modern Information Systems – Contributions to Data-driven Decision-making and Business Processes

DISSERTATION

zur Erlangung des Grades eines Doktors der Wirtschaftswissenschaft

eingereicht an der

Fakultät für Wirtschaftswissenschaften

der Universität Regensburg

vorgelegt von:

Alexander Schiller, M.Sc.

Berichterstatter:

Prof. Dr. Bernd Heinrich

Prof. Dr. Mathias Klier

Tag der Disputation: 13.12.2019

Acknowledgements

I would like to thank everyone who has supported me during the work on this thesis. In particular, I would like to express my deep gratitude towards Prof. Dr. Bernd Heinrich. He gave me the opportunity to begin working at his chair for information systems when I was enrolled in the master degree program in mathematics and guided me towards proficiency in the new research area. With respect to my thesis, he provided me with interesting ideas, extensive and constructive feedback, valuable advice, exceptional supervision and ample support. I would also like to express my sincere thanks to Prof. Dr. Mathias Klier for fruitful discussions, insightful comments and guidance and, in particular, precious assistance in communicating research.

In addition, I would like to thank my co-authors and colleagues – some of whom have become dear friends – for the superb collaboration. Due to them, the thesis could be written in a productive, inspiring and pleasant work atmosphere full of mutual support. I want to particularly point out Michael Szubartowicz, with whom sharing an office has been both prolific and enjoyable. Moreover, I would also like to offer my thanks to the students who worked with me and supported my research.

Last but not least, I would like to thank my family and friends for their ongoing encouragement. Most notably, I would like to take this opportunity to express my heartfelt gratitude towards my parents for their continuous support. A special thanks goes to my wonderful girlfriend Alina for enriching my life in a phenomenal way.

August 2019 *Alexander Schiller*

Summary of Contents

Explanatory note: To facilitate selective reading, each paper in the dissertation is treated as its own manuscript with respect to abbreviations, figure, table as well as general numbering and references and is thus self-contained.

Contents

Explanatory note: To facilitate selective reading, each paper in the dissertation is treated as its own manuscript with respect to abbreviations, figure, table as well as general numbering and references and is thus self-contained.

1 Introduction

In this chapter, first, a brief motivation for the dissertation is provided. It is followed by a discourse on its focal points, comprising a discussion of the addressed research questions. Subsequently, the structure of the dissertation including an overview of the contained papers is presented.

1.1 Motivation

Rapidly emerging new technologies and fast-paced changes omnipresent in today's world pose severe challenges for organizations, while at the same time offering groundbreaking opportunities for those able to capitalize on them. Some of the most notable recent technology-driven developments are outlined in the following. Afterwards, the aims of the dissertation and the role of artificial intelligence (AI) in the dissertation are clarified. Concluding the motivation, a synopsis and an outlook are given.

1.1.1 The Rising Availability of Data

A massive amount of data is becoming available to organizations through various sources such as user-generated content in the context of web 2.0 (e.g., on social media platforms, where users also produce content instead of just consuming), mobile transactions and digitization of analog sources (George et al., 2014). Moreover, data is intentionally being captured via sensors, for instance, in smartphones, Internet of Things-devices, vehicles or sensor networks (Krishnan and Cook, 2014). Furthermore, operational domains such as finance, bioinformatics and health care nowadays produce immense amounts of data (Kasemsap, 2016). For example, the increase of speed and frequency of customer interactions as well as the enormous number of transactions on the stock market lead to an explosion of financial data (Fang and Zhang, 2016; Sheng et al., 2017). Similarly, in health care, clinical data of diagnosis and treatment, genomic data as well as individual health records have accumulated drastically (Beam and Kohane, 2018; Wang et al., 2018; Zhang and Zhang, 2014). Additionally, a large number of open data repositories have become available for organizations to tap into (Attard et al., 2015; Bates, 2012; Zuiderwijk et al., 2018). All of these advancements have been recent and lead to a substantially rising availability of data to organizations.

The term used to describe this kind of data is "big data". It is commonly characterized by the "4Vs" volume, velocity, variety and veracity (Abbasi et al., 2016; Schroeck et al., 2012). Volume stands for the sheer size of the data. Velocity represents the high speed at which data is generated while variety describes the diversity of data with respect to structure, source and format. Veracity clarifies that this kind of data is often uncertain (i.e., of possibly poor data quality). Big data has fueled the increasing integration of data analysis into decision-making (Akter and Wamba, 2016; Janssen et al., 2017; Ngai et al., 2017; Provost and Fawcett, 2013; Zhou et al., 2016). Due the opportunities data-driven decision-making provides, it has received a lot of attention lately (Brynjolfsson and McElheran, 2016; Power, 2016; Provost and Fawcett,

2013). Studies suggest that data-driven decision-making improves returns and associate it with a 4-6% increase in productivity (Brynjolfsson et al., 2011; McAfee et al., 2012). For instance, in information systems, big data can be exploited to derive economically valuable insights about customers, competitors or the own organization and its processes (cf., e.g., Erevelles et al., 2016; van der Aalst, 2016a). To give an example, a marketer could exploit data analysis which shows in detail how customers react to different advertisements instead of selecting ads just based on experience and opinion on what will work (Provost and Fawcett, 2013). More examples include the targeting of customers in customer relationship management campaigns (Kumar and Reinartz, 2016), smart energy management (Zhou et al., 2016) and talent management (Witchalls, 2014).

1.1.1.1 The Surge in Uncertain Data

Gaining insights from big data is, however, significantly impeded by insufficient veracity – that is, poor data quality (Cai and Zhu, 2015; Ghasemaghaei and Calic, 2019a, 2019b; Hristova, 2016; Janssen et al., 2017; Witchalls, 2014). After all, its uncertainty indeed is a common characteristic of big data (Abbasi et al., 2016; Bendler et al., 2014; Lukoianova and Rubin, 2014). Yet, data of high quality is central to obtain meaningful results and avoid false conclusions – otherwise, the use of big data may result in costly "big error" (Liu et al., 2016). For instance, recent Gartner research reports that poor data quality is estimated to cost organizations an average of $15 million per year (Moore, 2018). IBM values the cost of poor data quality for the US economy at $3.1 trillion (IBM Big Data and Analytics Hub, 2016). This does not come as a surprise when considering that 84% of CEOs are not confident about the quality of the data they use for decision-making (KPMG, 2016; Rogers et al., 2017).

It is thus more critical than ever to be able to accurately assess and improve data quality. Efforts for this task have already been conducted since the last millennium (cf., e.g., Batini et al., 2009; Pipino et al., 2002; Wang, 1998). However, the rise of big data has significant implications for data quality assessment (Cai and Zhu, 2015; Lukoianova and Rubin, 2014). For instance, especially when a large amount of data is to be assessed, a comparison of data values to their real-world counterparts is infeasible as such comparisons tend to be time-consuming and cost-intensive (Zak and Even, 2017). In contrast, an aspect (a "data quality dimension"; Wand and Wang, 1996; Wang and Strong, 1996) that has gained in importance is data consistency – the degree to which assessed data is free of internal contradictions (Batini and Scannapieco, 2016; Redman, 1996). The assessment of consistency is generally based on rules and does not require a comparison of data values to their real-world counterparts (Batini and Scannapieco, 2016).

There are also data quality issues which are particularly aggravated by big data. A prominent example is the prevalence of duplicates (i.e., the same real-world entity being represented by multiple records; Draisbach and Naumann, 2011; Fan, 2015): Large and quickly expanding datasets as they are common in the big data era are particularly prone to containing such duplicates (Gruenheid et al., 2014; Vatsalan et al., 2017). In the context of information systems, duplicates can cause a variety of detrimental effects such as misjudgments of customers

(Bleiholder and Schmid, 2015), incorrect operational and strategic decisions (Helmis and Hollmann, 2009) and additional operative expenses (Draisbach, 2012). The assessment of datasets in regard to duplicates is thus increasing in relevance (Fan, 2015; Obermeier, 2019).

The results of data quality assessment can be incorporated in data-driven decision-making to alleviate problems caused by poor data quality (cf., e.g., Blake and Mangiameli, 2011; Feldman et al., 2018; Heinrich and Klier, 2015; Ofner et al., 2012; Watts et al., 2009). As discussed by Hristova (2016), this can happen either by directly integrating the data quality level in the decisions (e.g., as in Heinrich et al., 2009b) or indirectly by considering the data quality level in knowledge discovery techniques such as decision trees (e.g., as in Hristova, 2014). Especially with respect to big data, which often is heavily involved in decision-making, taking the data quality level in decision-making into account is crucial. Failure to do so can turn out to be very costly in case of erroneous decisions (Janssen et al., 2017; Shankaranarayanan and Cai, 2006). Yet, data quality assessment also needs to satisfy specific requirements to be adequate for proper decision support. For instance, a clear interpretability of the data quality assessment results is required to understand the actual meaning of the data quality level (Even and Shankaranarayanan, 2007; Heinrich et al., 2007a). Moreover, well-founded data quality assessment is required to assess changes in data quality, analyze which measures for data quality improvement are economically efficient and thus support an economically oriented data quality management (Even et al., 2010; Heinrich et al., 2009b; Wang, 1998).

1.1.1.2 The Emergence of Unstructured Data

As reflected in the "variety-v", diversity of data with respect to structure plays a vital role in big data. Data is called unstructured when it lacks a pre-defined data model, in contrast to structured data which possesses a formal schema and data models and is usually managed with a database system (Assunção et al., 2015). The emergence of unstructured data is another striking development organizations are facing: IDC expects the population on earth to create and replicate 163 Zettabytes (1 Zettabyte = 10^{21} bytes) of data in 2025, but the vast majority – and a rising percentage – of it will be unstructured (Gantz and Reinsel, 2013; Potnis, 2018). This development can, for instance, be explained with the pervasiveness of mobile devices in conjunction with the explosive growth of online social media (e.g., *Facebook*, *WhatsApp*, *Instagram*, *YouTube*, *Twitter*, *LinkedIn*, *WeChat*, *Yammer*). On these platforms, users frequently post images, audio clips, videos and textual content (e.g., from their smartphone), which is then available for analysis by organizations. Indeed, with respect to analysis, textual data has received particular attention (Aggarwal and Zhai, 2012; Allahyari et al., 2017; Weiss et al., 2015) as organizations are increasingly confronted with textual data not only from social media posts, but also in central business areas such as customer interaction (e.g., with customer reviews), internal and external communication (e.g., with emails), reporting (e.g., with documentations) or recruiting (e.g., with CVs). Yet, extracting information from textual data is much more difficult compared to structured data and, in particular, a manual analysis of large amounts of text is highly complex and time-consuming (Debortoli et al., 2016). However, complementary to

the emergence of textual data, great progress has been made in regard to analytical methods, with sophisticated approaches aiming to generate beneficial insights in an automated manner. The results from these data analyses can subsequently be used in business processes and decision-making (Ghasemaghaei and Calic, 2019b).

For instance, to extract valuable information from textual data, it is often essential to understand which topics the texts actually cover. To this end, topic modeling approaches (Alghamdi and Alfalqi, 2015; Hu et al., 2014) can be used to discover latent thematic structures in text collections and to, for example, identify thematically similar texts (Blei, 2012). Topic models consist of a number of topics, each of which is represented by terms firmly associated to the topic. Moreover, they determine how topics are distributed across the texts in the collection. In this way, the topics are representative for the content of the texts and allow for quick assessments of individual texts or the whole text collection. Topic modeling is widely applicable in organizations and has already successfully been employed to, for example, hotel critiques from social media (Guo et al., 2017) and consumer good reviews (Debortoli et al., 2016), enabling organizations to better understand what is important for their customers and to act accordingly. Further promising opportunities for organizations lie ahead as, for instance, nowadays a large number of CVs from social media (e.g., *LinkedIn*) or internet platforms such as *Indeed* can be accessed. These documents contain valuable data for human resource management processes and recruiting decisions in organizations.

In other cases, such as when analyzing customer reviews and associated ratings, it is indispensable to understand opinions and assessments expressed in text. A research field that has attracted tremendous attention and achieved significant progress in recent years in this regard is sentiment analysis. In general, approaches for sentiment analysis seek to identify the opinion expressed in a text regarding a certain entity (Liu, 2012; Medhat et al., 2014). Interest in sentiment analysis has been increasing considerably due to the wide range of applications (Agarwal et al., 2015). For instance, businesses benefit substantially from knowing the opinion of customer about their services and products. More generally, as Liu (2012) puts it, whenever a decision is made, one would like to know others' opinions and assessments. Nowadays, again, the explosive growth of online social media and user-generated data on the web facilitate obtaining a sizable number of user opinions expressed in text. To give an example, sentiment analysis has been applied heavily to *Twitter* data (e.g., Agarwal et al., 2011; Pak and Paroubek, 2010; Rosenthal et al., 2017) and was used to forecast product sales based on blog posts (Liu et al., 2007). A further interesting application case are platforms such as *Yelp*, *TripAdvisor* and *Amazon* where a large number of customers assess businesses (e.g., restaurants), locations, products and services in millions of publicly accessible reviews, comprising a textual part as well as an associated star rating. An analysis of these reviews enables organizations to gain a data-driven competitive advantage facilitated by a deeper understanding of customer opinions and assessments (e.g., Chatterjee, 2019).

1.1.2 A Complex, Dynamically Changing Environment

A further remarkable development is the dynamic change of the environment in which organizations operate today. They are confronted with an ever more globalized, complex and unpredictable world (Hamilton and Webster, 2018; Wetherly and Otter, 2014), shorter product life cycles (Bakker et al., 2014), increasing regulatory restrictions (Leuz and Wysocki, 2016), new customer demands (Jones et al., 2005) and faster times to market (Afonso et al., 2008). Additionally, scientific output is expanding at a quicker pace than ever (van Noorden, 2014), rapidly creating innovative and even disruptive technologies which require organizations to overhaul their operations and processes constantly or fall short of the competition.

Faced with such a development, business agility is essential to organizations (Lee et al., 2015). It can be defined as the ability to efficiently react and operate in a quickly changing, demanding environment (Couto et al., 2015; Gong and Janssen, 2012). Business processes, which are defined as an order of work activities with a beginning, an end, and clearly identified inputs and outputs (Davenport, 1993), are the center of value creation in an organization (Dumas et al., 2018). Hence, business process agility (business agility with respect to processes) and also business process flexibility (the ability to configure or adapt a process without completely replacing it; Hallerbach et al., 2010; Regev et al., 2007) are of particular importance (Chen et al., 2014; La Rosa et al., 2017; Mejri et al., 2018). Business processes are examined by the research field business process management (BPM; Dumas et al., 2018; van der Aalst, 2013; van der Aalst et al., 2016; Weske, 2012). They are represented by process models, which are crucial for the design, implementation and analysis of business processes (van der Aalst, 2013; vom Brocke and Mendling, 2018).

Process models have traditionally been constructed and monitored manually using tools such as the ARIS platform (Scheer, 2012). However, especially in a complex environment, a manual modeling is time-consuming and error-prone (Fahland et al., 2011; Mendling et al., 2008; Roy et al., 2014), inhibiting business process agility. Thus, research fields such as process mining (IEEE Task Force on Process Mining, 2012; van der Aalst, 2016b) and automated process model verification (Weber et al., 2008b; Weber et al., 2010) have emerged, striving to support process modelers in an automated manner during the analysis of processes. In line with these works, there have been efforts to develop an automated planning of process models using planning algorithms (Heinrich et al., 2012; Heinrich and Schön, 2015; Marrella, 2017, 2018). The aim of these approaches is the automated construction of feasible process models based upon specifications of a starting point of the process, semantically annotated actions and a set of goals.

Apart from representing the order in which actions in the business process are to be executed, process models also contain control flow patterns expressing how a business process can be executed (Russell et al., 2016; van der Aalst et al., 2003). The “basic” control flow patterns (capturing the elementary aspects of control flow) are sequence, parallel split, synchronization, exclusive choice and simple merge (Migliorini et al., 2011; Russell et al., 2016; van der Aalst

et al., 2003). For instance, a parallel split indicates that a single execution route is split into two or more concurrent sequences of action (Russell et al., 2016; van der Aalst et al., 2003). Automated planning of process models aims to also comprise the automated construction of control flow patterns (Heinrich et al., 2012, 2015; Heinrich and Schön, 2016).

In light of the dynamically changing environment outlined above, organizations frequently need to improve, revise and redesign their business processes and thus the respective process models (cf., e.g., Dumas et al., 2018; Vanwersch et al., 2016). This includes, on the one hand, fundamental and long-lasting transformations such as the alignment of processes to Basel III regulations in the financial industry (Allen et al., 2012; Härle et al., 2010). On the other hand, more short-term operative changes such as new product launches or process automation may also warrant a redesign of processes. To cite an example from La Rosa et al. (2017), the *Suncorp Group*, the largest insurer in Australia, frequently launches new insurance products. Whenever a new insurance product is launched, a corresponding process model is required. Since the required process model for the new insurance product is similar to already existing process models for established insurance products, an adaptation of such an existing model instead of constructing a new one from scratch may be advantageous with respect to effort and time. More generally, adapting process models instead of constructing from scratch is promising to improve business process agility in a dynamically changing environment.

Moreover, it needs to be taken into account that in today's complex environment, business processes rarely take place in isolation with just a single conducting actor. Rather, organizations concentrate on their core competences (e.g., using outsourcing; Oshri et al., 2015) and reduce their real net output ratio, leading to highly sophisticated value chains (Christopher, 2016; Timmer et al., 2014). To give an example, Dedrick et al. (2010) show how US-based *Apple's* iPod is assembled in China using hundreds of components which are shipped from around the world. In a similar vein, often multiple actors in a company (e.g., different individuals or business divisions) cooperate within intra-organizational processes (Ghrab et al., 2017). To appropriately represent the elaborate processes prevalent in today's complex environment and properly support process modelers, process models should thus account for multiple actors.

1.1.3 Aims of the Dissertation and the Role of AI

As these technology-driven developments bring up issues which have not yet been sufficiently treated in existing literature, the dissertation strives to provide contributions addressing corresponding research gaps (cf. Section 1.2). Besides its scientific relevance, the dissertation further seeks to propose concrete concepts and methods which support organizations in their transformation when adjusting to the developments in practice, in particular with respect to data-driven decision-making, business processes and business process management. To pursue these aims, concepts and methods from AI are employed. AI has a very broad scope, encompassing, for instance, fields such as machine learning, planning, quantifying uncertainty, natural language processing and decision-making under uncertainty (Barr and Feigenbaum, 2014; Nilsson, 2014; Russell and Norvig, 2016). It draws from a wide range of related areas such as economics,

mathematics, information systems, computer science, neuroscience, cybernetics, psychology, philosophy and linguistics (Russell and Norvig, 2016). Thus, several widespread definitions of AI exist. The dissertation follows the definition of Winston (1992) who calls AI "The study of the computations that make it possible to perceive, reason and act." (p. 14).

AI is a research area with a long history (Buchanan, 2005; Russell and Norvig, 2016) that has garnered a lot of traction and attention in recent years both in scientific literature (e.g., Lu et al., 2018; Nilsson, 2014; Russell et al., 2015a; Samek et al., 2017) as well as in public interest (Fast and Horvitz, 2017). As a matter of fact, big data is often referred to as an enabler and precondition for applying AI techniques (Fang et al., 2015; Najafabadi et al., 2015; O'Donovan et al., 2015; O'Leary, 2013; Rathore et al., 2016; Zhang et al., 2018). Nowadays, investment in AI surges from both companies and governments. Indeed, companies are estimated to have invested $26 billion to $36 billion in AI in 2016, and the numbers are growing rapidly (Bughin et al., 2017), demonstrating its relevance in practice. Similarly, governments strive to considerably increase expenditure for AI research. The EU commission seeks to reach at least €20 billion of investments by the end of 2020 (European Commission, 2018), and the German government has pledged initial €3 billion until 2025 (Federal Ministry for Economic Affairs and Energy, 2018). Still, these initiatives pale in comparison to USA's and especially China's AI efforts (Duranton et al., 2018; Foundation for Law & International Affairs, 2017). The total market for AI applications is estimated to reach a value of $127 billion by 2025 (Barton et al., 2017).

As Pan (2016) notes, AI has experienced major setbacks in the past, when state of research, available basis of data and computing power had not been as advanced as today. Even nowadays, in many areas, AI still needs to prove its potential (Makridakis, 2017). Yet, scholars and practitioners uniformly agree that AI will be of tremendous impact on economics and society (Agrawal et al., 2018; Barton et al., 2017; Bughin et al., 2017; Cockburn et al., 2018; Duranton et al., 2018; Russell et al., 2015b). Often, AI is assessed as "disruptive" and "fundamentally changing", and the opinions reach as far as to proclaiming an upcoming "AI revolution" (Makridakis, 2017). In many AI fields, recent progress has been rapid (Cockburn et al., 2018). The AI fields found to be particularly relevant and promising to cope with the aforementioned developments are outlined briefly in the following.

In AI, handling uncertainty, in particular with respect to uncertain data, is a highly common task (Kanal and Lemmer, 2014; Li and Du, 2017). To this end, often concepts from probability theory and economics (e.g., utility theory) are employed and furthered. Indeed, fields such as the quantification of uncertainty, probabilistic reasoning, decision-making under uncertainty and learning probabilistic models are at the very core of AI (Ghahramani, 2015; Russell and Norvig, 2016). In particular, decision-making under uncertainty allows to deal with challenges that often occur in data quality assessment such as considering different potential outcomes, estimating the value of data and acting under uncertainty (cf., e.g., Heinrich and Hristova, 2016;

Kanal and Lemmer, 2014). Hence, this AI field is well-suited to tackle (quantitative) data quality assessment and, in this way, support organizations in handling the rise of big data with uncertain veracity that they are facing.

The analysis of textual data is part of natural language processing (Bird et al., 2009; Manning and Schütze, 1999) and crucial to AI. More precisely, achieving proficiency with respect to automated understanding of text as well as extracting information from it is often cited as a major goal of AI (Bird et al., 2009; Cambria and White, 2014; Gangemi, 2013; Manning and Schütze, 1999; Russell and Norvig, 2016). After all, this extraction is a prerequisite for tasks such as information retrieval (Baeza-Yates et al., 2011) as well as understanding and participating in human communication (Russell and Norvig, 2016). In particular, sentiment analysis is seen as central to the advancement of AI since understanding emotions is key for emulating intelligence (Cambria et al., 2017). Thus, AI research in natural language processing provides a variety of concepts and methods for analyzing text from different lenses, ranging from neural networks for information retrieval (e.g., Liu et al., 2015; Shen et al., 2014) to statistical topic modeling (e.g., Blei, 2012; Cambria and White, 2014; Paisley et al., 2015) and lexical- or machine learning-based sentiment analysis (e.g., Cambria and White, 2014; Kolchyna et al., 2015; Taboada et al., 2011). Hence, this AI field is apt to assist organizations in dealing with the recent emergence of unstructured data and, in particular, to condense topics and opinions expressed in textual data.

Planning (defined as developing a plan of actions to reach a goal; Ghallab et al., 2016) is assessed to be a core component of AI (Russell and Norvig, 2016; Webber and Nilsson, 2014; Wilkins, 2014) because a plan allows to conduct rational actions. Thus, there has been significant research on concepts for AI planning, based on which a plethora of algorithms have been developed (cf., e.g., Geffner and Bonet, 2013; Ghallab et al., 2004, 2016; Haslum and Geffner, 2014). This knowledge base from AI planning has already been found to be very valuable for BPM (Marrella, 2017, 2018) and forms the foundations of automated planning of process models (e.g., Heinrich et al., 2012) which seeks to replace time-consuming and error-prone manual modeling. The further employment of AI planning in this area is thus promising to address a complex, dynamically changing environment and improve business process agility in organizations.

Decision-making under uncertainty, natural language processing and planning are all recurrent mentions of crucial AI fields (Barr and Feigenbaum, 2014; Miner et al., 2012; Nilsson, 2014; Russell and Norvig, 2016; Sigaud and Buffet, 2013). Against the background outlined above, it is argued that well-founded AI concepts and methods, in particular from these fields, are adequate and valuable to deal with technology-driven developments faced by organizations. In the dissertation, decision-making under uncertainty is used to address the surge in uncertain data. The emergence of unstructured data and the necessity to extract information from it is treated by natural language processing approaches. The increasingly complex, dynamically changing environment is dealt with using concepts and methods from planning. Vice versa, the

concepts and methods developed in the dissertation may also be prove to be useful in general AI research.

1.1.4 Synopsis and Outlook

To sum up, organizations today are facing a variety of challenging, technology-driven developments, three of the most notable ones being the surge in uncertain data, the emergence of unstructured data and a complex, dynamically changing environment. These developments require organizations to transform in order to stay competitive. AI with its fields decision-making under uncertainty, natural language processing and planning offers valuable concepts and methods to address the developments. The dissertation utilizes and furthers these contributions in focal points to address research gaps in existing literature and to provide concrete concepts and methods for the support of organizations in the transformation and improvement of data-driven decision-making, business processes and business process management. In particular, as motivated above, the focal points are the assessment of data quality, the analysis of textual data and the automated planning of process models. In regard to data quality assessment, probability-based approaches for measuring consistency and identifying duplicates as well as requirements for data quality metrics are suggested. With respect to analysis of textual data, the dissertation proposes a topic modeling procedure to gain knowledge from CVs as well as a model based on sentiment analysis to explain ratings from customer reviews. Regarding automated planning of process models, concepts and algorithms for an automated construction of parallelizations in process models, an automated adaptation of process models and an automated construction of multi-actor process models are provided. The focal points and the contribution of the dissertation are discussed in greater detail in the next section. Illustrating the discourse above, an overview of the research endeavor is given in Figure 1 (on the following page).

1.2 Focal Points and Research Questions

This section comprises a discussion of the focal points addressed by the dissertation, including brief summaries of the research background. Moreover, the central research questions are presented. Please note that the detailed discussion of related work is contained in the corresponding papers in the Sections 2, 3 and 4.

1.2.1 Focal Point 1: Assessment of Data Quality

Data quality is commonly defined as “the measure of the agreement between the data views presented by an information system and that same data in the real world” (Orr, 1998, p. 67). This is also the notion adopted in this dissertation. Data quality has been established as a multidimensional concept in the literature (Batini and Scannapieco, 2016; Pipino et al., 2002; Wand and Wang, 1996; Wang and Strong, 1996). This means that data quality can be seen from different lenses; for instance, a data value may be current, but at the same time inconsistent to another data value. The most frequently analyzed dimensions of data quality for data values are accuracy, currency, completeness and consistency as well as the deduplication of data (Batini and Scannapieco, 2016; Fan, 2015; Redman, 1996; Wang and Strong, 1996).

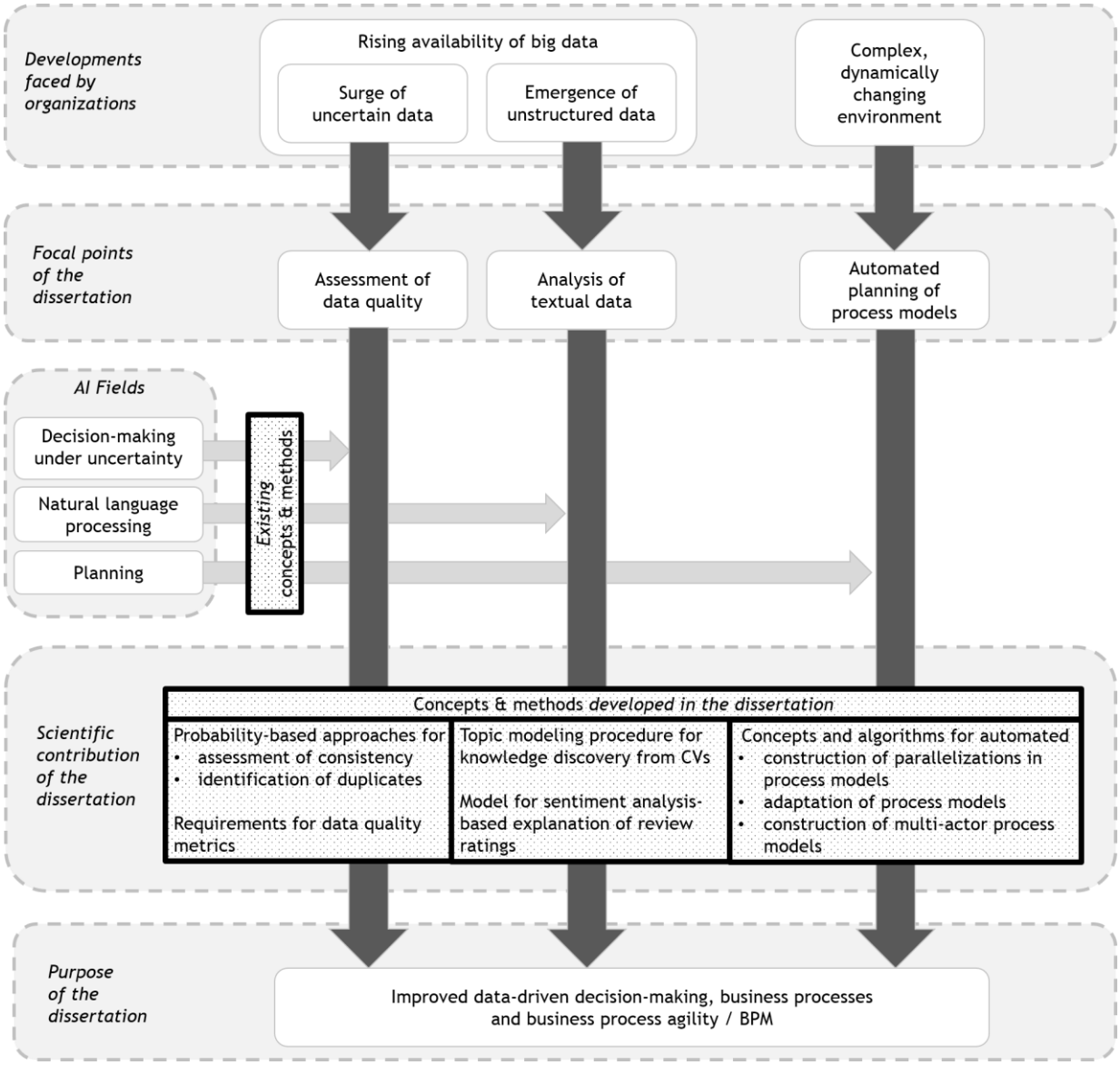

Figure 1. Overview of Research Endeavor

A commonly adopted framework for data quality management is the "Total Data Quality Management" methodology proposed by Wang (1998). It consists of four phases. The first phase is the "define"-phase, during which data quality requirements are identified and defined. The second phase, the "measure"-phase, comprises the development of data quality metrics as well as the assessment of data quality. In the third phase "analyze", the root causes for data quality issues are examined. Finally, in the "improve"-phase, key areas that allow for the improvement of data quality are identified. While all four phases of the methodology are imperative, in this dissertation, the focus is on the measure-phase. As already motivated, this is due to the fact that the assessment of data quality is essential for decision-making under uncertainty (cf., e.g., Janssen et al., 2017; Shankaranarayanan and Cai, 2006; Watts et al., 2009). Still, by supporting thorough assessment of data quality, the concepts and methods developed in this dissertation also offer starting points for analyzing data quality issues and, in particular, a well-founded

improvement of data quality. After all, they facilitate the assessment of the data quality level before and after conducting a data quality improvement measure.

Concepts and methods from probability theory are a key part for the quantification of uncertainty and decision-making under uncertainty in AI (Russell and Norvig, 2016) and, together with other concepts from the AI field decision-making under uncertainty, are also used in the dissertation, most notably within this focal point. Probability theory is a well-known and classical way to model uncertainty (Liu, 2015). For data quality assessment, a variety of concepts and methods from probability theory have already been used. For instance, for the assessment of currency, conditional probabilities (e.g., Heinrich et al., 2007a; Heinrich and Klier, 2009, 2011, 2015) and stochastic processes (Heinrich and Hristova, 2016; Zak and Even, 2017) have been employed. Similarly, Wechsler and Even (2012) address accuracy issues closely related to currency using Markov chains. In particular, probability-based metrics possess some distinct advantages. For instance, metric values which are expressed as probabilities have a concrete unit of measurement, are interval-scaled in [0, 1] and thus are clearly interpretable as a percentage.

Consistency has commonly been referred to as one of the most import data quality dimensions in data quality literature (Batini and Scannapieco, 2016; Blake and Mangiameli, 2009; Shankaranarayanan et al., 2012; Wand and Wang, 1996). It is also pivotal for practice, as 63% of financial institutions surveyed by Moges et al. (2011) confirmed inconsistency to be a main recurring data quality issue. In this dissertation, the focus is on semantic consistency, which is defined as the degree to which assessed data values are free of contradictions with respect to a rule set (cf., e.g., Batini and Scannapieco, 2016; Heinrich et al., 2007b; Mezzanzanica et al., 2012; Pipino et al., 2002; Redman, 1996). This focus is due to the fact that decision-making in organizations is usually based on data values. As rule sets can easily be applied to a large number of data values, the relevance of semantic consistency has even increased in the era of big data. Existing data quality metrics for semantic consistency are defined in such a way that each violation of a rule in the rule set indicates inconsistency (cf., e.g., Alpar and Winkelsträter, 2014; Hinrichs, 2002; Hipp et al., 2007). This is a highly limiting restriction, impeding the well-founded use of rules which are only fulfilled with a certain probability. Moreover, most existing metrics lack a clear interpretation of the metric values, which hampers their use for decision-making. A probability-based approach may circumvent these drawbacks of existing metrics. Hence, the first research question treated in this focal point is as follows:

RQ1: How can a novel probability-based approach for the assessment of the data quality dimension semantic consistency be defined such that it provides well-founded support for decision-making?

Duplicate detection is also one of the most extensively studied data quality research subjects (Christen, 2012; Elmagarmid et al., 2007; Fan, 2015; Winkler, 2006). In addition, its urgency in practice is further substantiated by the findings of Franz and von Mutius (2008), who show

that the cost of insufficient duplicate detection in a company's databases can easily reach millions of euros just based on inadequate customer communication. Big data has further amplified the necessity of duplicate detection as duplicates are more likely to occur with increasing volume and velocity (Gruenheid et al., 2014; Vatsalan et al., 2017). Moreover, big data often leads to the integration of multiple data sources, which is a frequent cause for duplicates (Fan, 2015). Probability-based duplicate detection, typically based on the framework by Fellegi and Sunter (1969), has been found to outperform other strategies for duplicate detection (Tromp et al., 2011). However, existing approaches for probability-based duplicate detection (e.g., Larsen and Rubin, 2001; Lehti and Fankhauser, 2006; Ravikumar and Cohen, 2004) are either based on limiting assumptions or suffer from restricted applicability. Additionally, none of these approaches takes the underlying causes for duplicates into account. These shortcomings interfere with the identification of a large number of duplicates and reduce the benefit of duplicate detection for decision-making. Thus, the second research question addressed in this focal point is:

RQ2: How can a novel probability-based approach for duplicate detection that considers the underlying causes for duplicates be defined such that it provides well-founded support for decision-making?

As already established, well-founded data quality assessment is required to support decision-making under uncertainty and an economically oriented management of data quality not just with respect to semantic consistency and duplicate detection, but in general. However, many existing data quality metrics are highly subjective (Cappiello et al., 2009) or specifically developed on an ad hoc basis for a problem at hand without consideration of practical use (Pipino et al., 2002). Thus, they lack an appropriate methodical foundation and their application may cause erroneous decisions and economic losses. To avoid such issues, researchers and practitioners have proposed requirements for data quality metrics (cf., e.g., Even and Shankaranarayanan, 2007; Hüner, 2011; Pipino et al., 2002). Yet, the verification of many of the suggested requirements is subjective and difficult, impeding their application in practice. Moreover, because existing literature suffers from the lack of a decision-oriented foundation, it does not agree on which requirements are indeed relevant to support decision-making under uncertainty and an economically oriented management of data quality. Therefore, the third research question discussed in this focal point is as follows:

RQ3: Based on a decision-theoretic foundation, which clearly defined requirements must a data quality metric satisfy to support both decision-making under uncertainty and an economically oriented management of data quality?

1.2.2 Focal Point 2: Analysis of Textual Data

In line with the emergence and increasing relevance of textual data, research on its analysis is progressing, aiming to enable the extraction of valuable insights. Notably, as Cambria and White (2014) put it, a shift from syntactical to semantic approaches is occurring, with the even-

tual aim to reach "natural language understanding" instead of natural language processing. Unlike purely syntactical techniques, semantic approaches focus on intrinsic meaning contained in text and operate on a concept-level. Among such works are advanced topic modeling and sentiment analysis approaches (Cambria and White, 2014). In particular, state-of-the-art topic modeling approaches such as latent Dirichlet allocation (Belford et al., 2018; Blei et al., 2003) relate words to each other to identify semantically coherent topics (Lau et al., 2014; Newman et al., 2010). Similarly, recent approaches for sentiment analysis semantically associate opinions and assessments with different aspects of the characterized entity, resulting in an aspect-based sentiment analysis (Liu, 2012; Pontiki et al., 2016; Schouten and Frasincar, 2016). Employing these kind of approaches, the concepts and methods developed in the dissertation contribute to the semantic analysis of textual data.

The research in this focal point is based on a variety of different concepts and methods from natural language processing intertwined with other (AI) concepts and methods. A plethora of different approaches from AI fields have been utilized for the analysis of textual data. For topic modeling, the most common approach (Belford et al., 2018) is the probabilistic machine learning-based natural language processing method latent Dirichlet allocation (Blei et al., 2003; Blei, 2012). Sentiment analysis is also founded on multiple different natural language processing concepts and methods (Cambria and White, 2014; Liu, 2012). The dissertation additionally engages a regression model, which can be seen as part of machine learning as well (Bishop, 2006; Russell and Norvig, 2016). Text pre-preprocessing, which often is necessary before invoking the actual method for analysis, involves, amongst other natural language processing routines, part-of-speech tagging, named entity recognition, lemmatization and dependency parsing (Collobert et al., 2011; Manning et al., 2014).

A pronounced manifestation of today's textual data ubiquity is businesses' accessibility to CVs harnessing social media (e.g., *LinkedIn*), job platforms (e.g., *Indeed*) and private homepages. Applying a specialized topic model procedure to a database of CVs collected from such sources should facilitate to discover knowledge from CVs and provide high-quality, fine-grained topics representing, for instance, skills, work expertise and abilities. This supports, for example, categorizing CVs, swiftly assessing a CV's contents and identifying candidates for job offers, while avoiding drawbacks associated with manual assessments of large textual databases (Debortoli et al., 2016). Still, despite proposals of topic modeling approaches for job offers and further related tasks (Gao and Eldin, 2014; Gorbacheva et al., 2016), a topic modeling procedure for this application context has not yet been suggested. Following existing (generic) literature for topic modeling (Blei, 2012; Debortoli et al., 2016) which does not take into account the characteristics of CVs leads suboptimal results. Additionally, such work does not report on how exactly to apply topic modeling to CVs and how to capitalize on its results, thus impeding promising use cases in the support of human resource management processes (e.g., proactive recruiting from the web). Hence, the first research question treated in this focal point is as follows:

RQ4: How can topic modeling be used to discover knowledge from CVs and provide support in human resource management processes?

The proliferation of online customer reviews is one of the most prominent examples of the flood of textual data available on the web. On most platforms such as *Yelp*, *TripAdvisor* and *Amazon*, the reviews consist of an ordinal-scaled rating (e.g., 1-5 stars) and a textual part in which customer opinions and assessments are expressed. Online customer reviews are a critical means to reduce information asymmetries about businesses, products and services (Chatterjee, 2019; Hu et al., 2008) and play a crucial role in the decision-making process of potential customers (Minnema et al., 2016; Phillips et al., 2017; Ye et al., 2011). In fact, a recent survey has revealed that 86% of customers read reviews for local businesses, and that 57% will only use a business if it is rated 4 or more stars (Murphy, 2018). The relevance of understanding why customers rate the way they do is thus evident. Conducting an aspect-based sentiment analysis to this end is promising, since often, the customer's opinion regarding multiple aspects (e.g., service and food in case of a restaurant review) is disclosed in the textual part of the review, and aspect-based sentiment analysis allows for a differentiated, in-depth assessment (Liu, 2012; Schouten and Frasincar, 2016). Moreover, in contrast to other approaches based on word counts (Fu et al., 2013) or factor loadings (Xiang et al., 2015), aspect-based sentiment analysis extracts the customer's opinions in an interpretable way (Zhu et al., 2011). Such an interpretability is also advantageous in practice, for instance by enabling a data-driven competitive advantage to businesses that are able to uncover the reasoning behind customers' assessments. Still, while approaches to explain the ratings in customer reviews based on textual data exist, these approaches do not consider aspect-based sentiments (e.g., Debortoli et al., 2016), do not address methodical issues associated with the ratings (e.g., Linshi, 2014) or do not evaluate how well they are actually able to explain the ratings (e.g., Ganu et al., 2009; Ganu et al., 2013). Thus, the second research question addressed in this focal point is as follows:

RQ5: How can aspect-based sentiments contained in the textual parts of online customer reviews be used to explain and interpret the associated overall star ratings?

1.2.3 Focal Point 3: Automated Planning of Process Models

Automated planning of process models is a part of BPM. BPM focuses on discovering, analyzing, implementing and optimizing business processes in an economically efficient way (Dumas et al., 2018; vom Brocke and Rosemann, 2015; Weske, 2012). To this end, concepts and methods from multiple research areas such as management sciences and information technology are employed (Dumas et al., 2018; van der Aalst, 2013). BPM involves different phases corresponding to the lifecycle of processes (e.g., Dumas et al., 2018; van der Aalst, 2013; Wetzstein et al., 2007). The cycle begins with a "process modeling"-phase (sometimes called "process design"-phase or "process discovery"-phase). It comprises the identification of business processes and the construction of the respective process models (Dumas et al., 2018; vom Brocke and Rosemann, 2015; Wetzstein et al., 2007). In the second phase, the "process implementation"-phase, process models are transformed into executable processes (van der Aalst, 2013;

vom Brocke and Rosemann, 2015; Wetzstein et al., 2007). The third phase is the "process execution"-phase (or "process enactment"-phase). This phase incorporates the actual initiation and running of business process instances (vom Brocke and Rosemann, 2015; Weske, 2012; Wetzstein et al., 2007). Finally, in the "analyze"-phase, the business processes are monitored and evaluated to allow for further improvement, for instance using process mining (Dumas et al., 2018; Weske, 2012; Wetzstein et al., 2007). As automated planning of process models is part of the process modeling-phase (Heinrich et al., 2015), the concepts and methods developed in the dissertation mainly contribute to this phase of the BPM lifecycle. However, some research fields in other phases deal with related tasks and thus may also benefit from the conducted research. For instance, the concepts and methods may also be of use for the development of corresponding approaches in automated (web) service composition and selection in the process implementation- and the process execution-phase. Moreover, they may prove useful for the research fields process mining and process model verification in the process analysis-phase.

To address challenges in automated planning of process models, concepts and methods from AI planning are employed and extended. AI planning methods allow to leverage a higher level of automation in BPM (Marrella, 2018). In particular, automated planning of process models can be understood as a specific planning problem with the objective to arrange process model components in a feasible order based on input data given in form of an initial state, a set of available actions and conditions for goal states. Thus, AI planning methods (e.g., Bertoli et al., 2006; Bertoli et al., 2010) and a large variety of concepts from AI planning such as belief states, state-transition systems and applicability (Bertoli et al., 2006; Ghallab et al., 2016; Russell and Norvig, 2016) are commonly used in automated planning of process models (e.g., Heinrich et al., 2015; Heinrich and Schön, 2015, 2016). These concepts and methods serve as valuable foundation for extensions in the dissertation, which in turn are compatible with existing works.

A central challenge in automated planning of process models is to not only plan sequences of actions but also control flow patterns (Russell et al., 2016; van der Aalst et al., 2003) representing the control flow of a process (Heinrich et al., 2012). By splitting up and synchronizing process execution in two or more concurrent sequences of action, the patterns parallel split and synchronization capture fundamental aspects of processes and thus are assessed to be essential (Russell et al., 2016; Soffer et al., 2015; van der Aalst and ter Hofstede, 2005). Moreover, actions conducted in parallel are omnipresent in practice (He et al., 2008; Russell et al., 2016) and offer to, for instance, reduce execution times of processes (Alrifai et al., 2012). Yet, while approaches for the automated construction of the patterns exclusive choice and simple merge have already been proposed (Heinrich et al., 2009a; Heinrich et al., 2015; Heinrich and Schön, 2016), constructing parallel splits and synchronizations has remained an unsolved issue. Existing works in related BPM research fields (e.g., (web) service composition) provide thought-provoking starting points for research, but suffer from shortcomings such as the inability to handle complex parallelizations. Thus, the first research question discussed in this focal point is as follows:

RQ6: How can the control flow patterns parallel split and synchronization be constructed in an automated manner, including complex parallelizations?

As the need of improving, revising and redesigning business processes and process models becomes more frequent, adapting process models plays an increasingly important role. These adaptations can be carried out manually (van der Aalst et al., 2009; Weber et al., 2008a). However, as argued above, manual modeling is time-consuming and subject to human errors, particularly when process instances are not available for modelers' analysis. In contrast, an automated adaptation of process models based on automated planning may allow to rapidly construct process models for processes which are to be changed in the future, with the changes not yet realized. Hence, such an adaptation of process models is promising to improve business process agility and flexibility. Yet, existing approaches for an automated adaptation of process models often rely on execution logs of already executed process instances (cf., e.g., Fahland and van der Aalst, 2012), inhibiting their use for adapting process models in advance. Those approaches that do adapt process models in advance only cover changes to actions and just adapt a part of the process model (Eisenbarth et al., 2011; Eisenbarth, 2013; Lautenbacher et al., 2009). Therefore, the process models adapted by these approaches are not complete (i.e., they do not contain all feasible paths), providing limited support to process modelers. This leads to the second research question treated in this focal point:

RQ7: How can process models be adapted to needs for change in advance in an automated manner, such that the resulting process models are correct and complete?

In today's increasingly complex inter- and intra-organizational business processes, usually each conducting actor (e.g., suppliers, partnering companies, departments, employees) has its individual starting point, follows own distinctive goals and cooperates with other actors (Becker et al., 2013; Ghrab et al., 2017; Stadtler et al., 2015). Conventional process models are not particularly well-suited to represent such processes (Pulgar and Bastarrica, 2017). Instead, these peculiarities of multi-actor processes should be reflected conceptually, thus also enabling the construction of multi-actor process models beneficial in practice. Existing concepts (e.g., swimlanes in modeling languages such as BPMN and UML; Object Management Group, 2013, 2015; Shapiro et al., 2012), however, are mostly annotations which are limited in their ability to represent individual starting points and goals as well as actions conducted by multiple actors. Furthermore, these concepts tend to result in cluttered process models which are hard to understand (Pulgar and Bastarrica, 2017). Similarly, current approaches for automated planning of process models do not support actor-specific initial and goal states and cannot efficiently deal with actions that can be executed by multiple actors. This is despite the fact that an automated planning approach is promising to handle the complexity of multi-actor processes. The third research question taken up in this focal point is thus as follows:

RQ8: How can a conceptual foundation to represent multi-actor process models be specified, facilitating the construction of feasible multi-actor process models by means of an automated planning approach?

1.3 Structure of the Dissertation

The dissertation contains eight papers, which address the research questions presented in the previous section. The following Figure 2 gives an overview of the papers. For each paper, the discussed research question, its title, its authors (which had been ordered alphabetically in each case), its outlet and its status with regard to acceptance are provided. Additionally, the focal point of each paper is disclosed in the first column and illustrated in form of the row background color.

Focal Point	Research Question	Title	Authors	Outlet	Current Status
Assessment of data quality	RQ1	Assessing Data Quality – A Probability-based Metric for Semantic Consistency	Bernd Heinrich Mathias Klier Alexander Schiller Gerit Wagner	Decision Support Systems (DSS)	accepted and published in Issue 110 (2018)
	RQ2	Event-driven Duplicate Detection – A Probability-based Approach	Bernd Heinrich Mathias Klier Andreas Obermeier Alexander Schiller	Proceedings of the European Conference on Information Systems (ECIS)	accepted and published in the 2018 proceedings; winner of the Claudio Ciborra Award for the most innovative paper of ECIS 2018
	RQ3	Requirements for Data Quality Metrics	Bernd Heinrich Diana Hristova Mathias Klier Alexander Schiller Michael Szubartowicz	Journal of Data and Information Quality (JDIQ)	accepted and published in Volume 9, Issue 2 (2018)
Analysis of textual data	RQ4	Knowledge Discovery from CVs: A Topic Modeling Procedure	Alexander Schiller	Proceedings of the Internationale Tagung Wirtschaftsinformatik (WI)	accepted and published in the 2019 proceedings
	RQ5	Explaining the Stars: Aspect-based Sentiment Analysis of Online Customer Reviews	Markus Binder Bernd Heinrich Mathias Klier Andreas Obermeier Alexander Schiller	Proceedings of the European Conference on Information Systems (ECIS)	accepted and published in the 2019 proceedings
Automated planning of process models	RQ6	Automated Planning of Process Models: The Construction of Parallel Splits and Synchronizations	Bernd Heinrich Felix Krause Alexander Schiller	Decision Support Systems (DSS)	accepted and published in Issue 125 (2019)
	RQ7	Adapting Process Models via an Automated Planning Approach	Bernd Heinrich Alexander Schiller Dominik Schön Michael Szubartowicz	Journal of Decision Systems (JDS)	under review
	RQ8	The Cooperation of Multiple Actors within Process Models: An Automated Planning Approach	Bernd Heinrich Alexander Schiller Dominik Schön	Journal of Decision Systems (JDS)	accepted and published in Volume 27, Issue 4 (2018)

Figure 2. Overview of Papers contained in the Dissertation

The remainder of the dissertation is structured as follows (cf. Figure 3). Subsequent to this introduction, the Sections 2, 3 and 4 present the three focal points with the corresponding papers and thus the main contribution of the dissertation. Here, apart from the papers themselves, brief

summaries of each paper are provided. They clarify how the research questions have been addressed and suggest implications for decision-making and business processes. Moreover, they sum up which (AI) concepts and methods have been used and developed. Finally, Section 5 concludes the dissertation with a discussion of major findings and directions for further research.

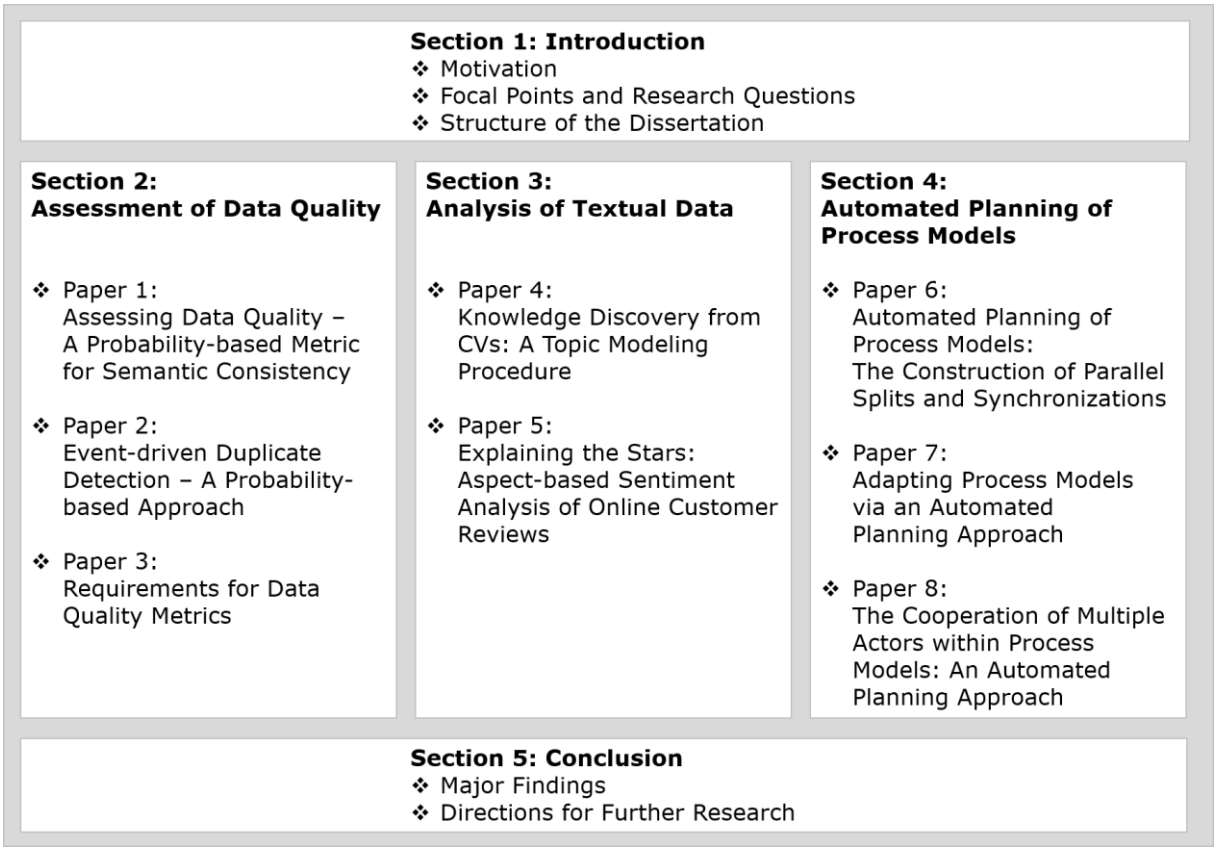

Figure 3. Structure of the Dissertation

1.4 References

Abbasi, A., S. Sarker and R. H. L. Chiang (2016). "Big data research in information systems: Toward an inclusive research agenda" *Journal of the Association for Information Systems (JAIS)* 17 (2), I.

Afonso, P., M. Nunes, A. Paisana and A. Braga (2008). "The influence of time-to-market and target costing in the new product development success" *International Journal of Production Economics* 115 (2), 559–568.

Agarwal, A., B. Xie, I. Vovsha, O. Rambow and R. Passonneau (2011). "Sentiment analysis of twitter data". In: *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pp. 30–38.

Agarwal, B., N. Mittal, P. Bansal and S. Garg (2015). "Sentiment analysis using common-sense and context information" *Computational Intelligence and Neuroscience (CIN)* 2015, 30.

Aggarwal, C. C. and C. Zhai (2012). *Mining text data:* Springer Science & Business Media.

Agrawal, A., J. Gans and A. Goldfarb (2018). *Prediction Machines: The simple economics of artificial intelligence:* Harvard Business Press.

Akter, S. and S. F. Wamba (2016). "Big data analytics in E-commerce: a systematic review and agenda for future research" *Electronic Markets (EM)* 26 (2), 173–194.

Alghamdi, R. and K. Alfalqi (2015). "A survey of topic modeling in text mining" *International Journal of Advanced Computer Science and Applications (IJACSA)* 6 (1).

Allahyari, M., S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez and K. Kochut (2017). "A brief survey of text mining: Classification, clustering and extraction techniques" *arXiv preprint arXiv:1707.02919*.

Allen, B., K. K. Chan, A. Milne and S. Thomas (2012). "Basel III: Is the cure worse than the disease?" *International Review of Financial Analysis (IRFA)* 25, 159–166.

Alpar, P. and S. Winkelsträter (2014). "Assessment of data quality in accounting data with association rules" *Expert Systems with Applications* 41 (5), 2259–2268.

Alrifai, M., T. Risse and W. Nejdl (2012). "A hybrid approach for efficient Web service composition with end-to-end QoS constraints" *ACM Transactions on the Web (TWEB)* 6 (2), 7.

Assunção, M. D., R. N. Calheiros, S. Bianchi, M. A. S. Netto and R. Buyya (2015). "Big Data computing and clouds: Trends and future directions" *Journal of Parallel and Distributed Computing* 79, 3–15.

Attard, J., F. Orlandi, S. Scerri and S. Auer (2015). "A systematic review of open government data initiatives" *Government Information Quarterly (GIQ)* 32 (4), 399–418.

Baeza-Yates, R., Ribeiro, Berthier de Araújo Neto and others (2011). *Modern information retrieval:* ACM Press & Addison-Wesley.

Bakker, C., F. Wang, J. Huisman and M. Den Hollander (2014). "Products that go round: exploring product life extension through design" *Journal of Cleaner Production* 69, 10–16.

Barr, A. and E. A. Feigenbaum (2014). *The handbook of artificial intelligence:* Butterworth-Heinemann.

Barton, D., J. Woetzel, J. Seong and Q. Tian (2017). *Artificial Intelligence: Implications for China.* McKinsey Global Institute.

Bates, J. (2012). "This is what modern deregulation looks like: co-optation and contestation in the shaping of the UK's Open Government Data Initiative" *The Journal of Community Informatics (CI)* 8 (2), 1–20.

Batini, C., C. Cappiello, C. Francalanci and A. Maurino (2009). "Methodologies for data quality assessment and improvement" *ACM Computing Surveys (CSUR)* 41 (3), 16.

Batini, C. and M. Scannapieco (2016). *Data and information quality:* Springer.

Beam, A. L. and I. S. Kohane (2018). "Big data and machine learning in health care" *JAMA* 319 (13), 1317–1318.

Becker, J., D. Beverungen, R. Knackstedt, M. Matzner, O. Müller and J. Pöppelbuß (2013). "Designing Interaction Routines in Service Networks" *Scandinavian Journal of Information Systems (SJIS)* 25 (1), 37–68.

Belford, M., B. Mac Namee and D. Greene (2018). “Stability of topic modeling via matrix factorization” *Expert Systems with Applications* 91, 159–169.

Bendler, J., S. Wagner, T. Brandt and D. Neumann (2014). “Taming uncertainty in big data” *Business & Information Systems Engineering (BISE)* 6 (5), 279–288.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2006). “Strong planning under partial observability” *Artificial Intelligence* 170 (4), 337–384.

Bertoli, P., M. Pistore and P. Traverso (2010). “Automated composition of web services via planning in asynchronous domains” *Artificial Intelligence* 174 (3), 316–361.

Bird, S., E. Klein and E. Loper (2009). *Natural language processing with Python: analyzing text with the natural language toolkit:* O’Reilly Media, Inc.

Bishop, C. M. (2006). *Pattern recognition and machine learning:* Springer.

Blake, R. and P. Mangiameli (2011). “The effects and interactions of data quality and problem complexity on classification” *Journal of Data and Information Quality (JDIQ)* 2 (2), 8.

Blake, R. H. and P. Mangiameli (2009). “Evaluating the semantic and representational consistency of interconnected structured and unstructured data”. In: *Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009)*.

Blei, D. M. (2012). “Probabilistic Topic Models” *Communications of the ACM* 55 (4), 77–84.

Blei, D. M., A. Y. Ng and M. I. Jordan (2003). “Latent dirichlet allocation” *Journal of Machine Learning Research (JMLR)* 3 (Jan), 993–1022.

Bleiholder, J. and J. Schmid (2015). “Datenintegration und Deduplizierung”. In *Daten-und Informationsqualität: Auf dem Weg zur Information Excellence. 2. Auflage*, 121–140: Vieweg + Teubner.

Brynjolfsson, E., L. M. Hitt and H. H. Kim (2011). “Strength in numbers: How does data-driven decisionmaking affect firm performance?” *SSRN* (1819486).

Brynjolfsson, E. and K. McElheran (2016). “The rapid adoption of data-driven decision-making” *American Economic Review (AER)* 106 (5), 133–139.

Buchanan, B. G. (2005). “A (very) brief history of artificial intelligence” *AI Magazine* 26 (4), 53.

Bughin, J., E. Hazan, S. Ramaswamy, M. Chui, T. Allas, P. Dahlström, N. Henke and M. Trench (2017). *Artificial intelligence: The next digital frontier.* McKinsey Global Institute.

Cai, L. and Y. Zhu (2015). “The challenges of data quality and data quality assessment in the big data era” *Data Science Journal* 14.

Cambria, E., D. Das, S. Bandyopadhyay and A. Feraco (2017). *A practical guide to sentiment analysis:* Springer.

Cambria, E. and B. White (2014). “Jumping NLP curves: A review of natural language processing research” *IEEE Computational Intelligence Magazine (CIM)* 9 (2), 48–57.

Cappiello, C., M. Comuzzi and others (2009). “A utility-based model to define the optimal data quality level in IT service offerings”. In: *Proceedings of the 17th European Conference on Information Systems (ECIS 2009)*.

Chatterjee, S. (2019). "Explaining customer ratings and recommendations by combining qualitative and quantitative user generated contents" *Decision Support Systems (DSS)* 119, 14–22.

Chen, Y., Y. Wang, S. Nevo, J. Jin, L. Wang and W. S. Chow (2014). "IT capability and organizational performance: the roles of business process agility and environmental factors" *European Journal of Information Systems (EJIS)* 23 (3), 326–342.

Christen, P. (2012). *Data matching. Concepts and techniques for record linkage, entity resolution, and duplicate detection:* Springer-Verlag.

Christopher, M. (2016). *Logistics & supply chain management:* Pearson UK.

Cockburn, I. M., R. Henderson and S. Stern (2018). *The impact of artificial intelligence on innovation.* National Bureau of Economic Research.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa (2011). "Natural language processing (almost) from scratch" *Journal of Machine Learning Research (JMLR)* 12 (Aug), 2493–2537.

Couto, E. S., F. C. Lopes and R. D. Sousa (2015). "Can IS/IT governance contribute for business agility?" *Procedia Computer Science* (64), 1099–1106.

Davenport, T. H. (1993). *Process innovation: reengineering work through information technology:* Harvard Business Press.

Debortoli, S., O. Müller, I. A. Junglas and J. vom Brocke (2016). "Text mining for information systems researchers: an annotated topic modeling tutorial" *Communications of the Association for Information Systems (CAIS)* 39, 7.

Dedrick, J., K. L. Kraemer and G. Linden (2010). "Who profits from innovation in global value chains?: A study of the iPod and notebook PCs" *Industrial and Corporate Change* 19 (1), 81–116.

Draisbach, U. (2012). *Partitionierung zur effizienten Duplikaterkennung in relationalen Daten:* Springer Vieweg.

Draisbach, U. and F. Naumann (2011). "A generalization of blocking and windowing algorithms for duplicate detection". In: *2011 International Conference on Data and Knowledge Engineering (ICDKE)*, pp. 18–24.

Dumas, M., M. La Rosa, J. Mendling and H. A. Reijers (2018). *Fundamentals of business process management (2nd edition):* Springer.

Duranton, S., J. Erlebach and M. Pauly (2018). *Mind the (AI) gap.* Boston Consulting Group Gamma.

Eisenbarth, T. (2013). "Semantic process models: Transformation, adaptation, resource consideration". Dissertation. University of Augsburg.

Eisenbarth, T., F. Lautenbacher and B. Bauer (2011). "Adaptation of process models – A semantic-based approach" *Journal of Research and Practice in Information Technology (JRPIT)* 43 (1), 5–23.

Elmagarmid, A. K., P. G. Ipeirotis and V. S. Verykios (2007). "Duplicate record detection. A survey" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 19 (1), 1–16.

Erevelles, S., N. Fukawa and L. Swayne (2016). “Big Data consumer analytics and the transformation of marketing” *Journal of Business Research (JBR)* 69 (2), 897–904.

European Commission (2018). *Member States and Commission to work together to boost artificial intelligence "Made in Europe" (Press Release).*

Even, A. and G. Shankaranarayanan (2007). “Utility-driven assessment of data quality” *ACM SIGMIS Database* 38 (2), 75–93.

Even, A., G. Shankaranarayanan and P. D. Berger (2010). “Evaluating a model for cost-effective data quality management in a real-world CRM setting” *Decision Support Systems (DSS)* 50 (1), 152–163.

Fahland, D., C. Favre, J. Koehler, N. Lohmann, H. Völzer and K. Wolf (2011). “Analysis on demand: Instantaneous soundness checking of industrial business process models” *Data & Knowledge Engineering (DKE)* 70 (5), 448–466.

Fahland, D. and W. M. P. van der Aalst (2012). “Repairing Process Models to Reflect Reality” *Business Process Management* 7481, 229–245.

Fan, W. (2015). “Data quality: from theory to practice” *ACM SIGMOD Record* 44 (3), 7–18.

Fang, B. and P. Zhang (2016). “Big data in finance”. In *Big data concepts, theories, and applications*, pp. 391–412: Springer.

Fang, H., Z. Zhang, C. J. Wang, M. Daneshmand, C. Wang and H. Wang (2015). “A survey of big data research” *IEEE Network* 29 (5), 6–9.

Fast, E. and E. Horvitz (2017). “Long-term trends in the public perception of artificial intelligence”. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Federal Ministry for Economic Affairs and Energy (2018). *Federal Government adopts Artificial Intelligence Strategy (Press Release).*

Feldman, M., A. Even and Y. Parmet (2018). “A methodology for quantifying the effect of missing data on decision quality in classification problems” *Communications in Statistics-Theory and Methods* 47 (11), 2643–2663.

Fellegi, I. P. and A. B. Sunter (1969). “A theory for record linkage” *Journal of the American Statistical Association (JASA)* 64 (328), 1183–1210.

Foundation for Law & International Affairs (2017). *China’s New Generation of Artificial Intelligence Development Plan (Press Release).*

Franz, T. and C. von Mutius (2008). *Kundendatenqualität – Ein Schlüssel zum Erfolg im. Kundendialog.* Zürich, Switzerland: Swiss CRM Forum 2008.

Fu, B., J. Lin, L. Li, C. Faloutsos, J. Hong and N. Sadeh (2013). “Why people hate your app: Making sense of user feedback in a mobile app store”. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1276–1284.

Gangemi, A. (2013). “A comparison of knowledge extraction tools for the semantic web”. In: *Extended Semantic Web Conference (ESWC 2013)*, pp. 351–366.

Gantz, J. and D. Reinsel (2013). *The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east.* IDC.

Ganu, G., N. Elhadad and A. Marian (2009). "Beyond the stars: improving rating predictions using review text content". In: *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, pp. 1–6.

Ganu, G., Y. Kakodkar and A. Marian (2013). "Improving the quality of predictions using textual information in online user reviews" *Information Systems* 38 (1), 1–15.

Gao, L. and N. Eldin (2014). "Employers' expectations: A probabilistic text mining model" *Procedia Engineering* 85, 175–182.

Geffner, H. and B. Bonet (2013). "A concise introduction to models and methods for automated planning" *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8 (1), 1–141.

George, G., M. R. Haas and A. Pentland (2014). *Big data and management:* Academy of Management Briarcliff Manor, NY.

Ghahramani, Z. (2015). "Probabilistic machine learning and artificial intelligence" *Nature* 521 (7553), 452.

Ghallab, M., D. Nau and P. Traverso (2004). *Automated planning: theory & practice:* Elsevier.

Ghallab, M., D. Nau and P. Traverso (2016). *Automated planning and acting:* Cambridge University Press.

Ghasemaghaei, M. and G. Calic (2019a). "Can big data improve firm decision quality? The role of data quality and data diagnosticity" *Decision Support Systems (DSS)* 120, 38–49.

Ghasemaghaei, M. and G. Calic (2019b). "Does big data enhance firm innovation competency? The mediating role of data-driven insights" *Journal of Business Research (JBR)* 104, 69–84.

Ghrab, S., I. Saad, G. Kassel and F. Gargouri (2017). "A Core Ontology of Know-How and Knowing-That for improving knowledge sharing and decision making in the digital age" *Journal of Decision Systems (JDS)* 26 (2), 138–151.

Gong, Y. and M. Janssen (2012). "From policy implementation to business process management: Principles for creating flexibility and agility" *Government Information Quarterly (GIQ)* 29, S61-S71.

Gorbacheva, E., A. Stein, T. Schmiedel and O. Müller (2016). "The role of gender in business process management competence supply" *Business & Information Systems Engineering (BISE)* 58 (3), 213–231.

Gruenheid, A., X. L. Dong and D. Srivastava (2014). "Incremental record linkage" *Proceedings of the VLDB Endowment* 7 (9), 697–708.

Guo, Y., S. J. Barnes and Q. Jia (2017). "Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation" *Tourism Management* 59, 467–483.

Hallerbach, A., T. Bauer and M. Reichert (2010). "Capturing variability in business process models: the Provop approach" *Journal of Software Maintenance and Evolution: Research and Practice* 22 (6-7), 519–546.

Hamilton, L. and P. Webster (2018). *The international business environment:* Oxford University Press.

Härle, P., E. Lüders, T. Pepanides, S. Pfetsch, T. Poppensieker and U. Stegemann (2010). *Basel III and European banking: Its impact, how banks might respond, and the challenges of implementation.* McKinsey & Company.

Haslum, P. and H. Geffner (2014). "Heuristic planning with time and resources". In: *Proceedings of the 6th European Conference on Planning*.

He, Q., J. Yan, H. Jin and Y. Yang (2008). "Adaptation of web service composition based on workflow patterns". In: *Proceedings of the International Conference on Service-Oriented Computing (ICSOC 2008)*, pp. 22–37.

Heinrich, B., M. Bolsinger and M.-A. Bewernik (2009a). "Automated planning of process models: the construction of exclusive choices". In *Proceedings of the 30th International Conference on Information Systems (ICIS 2009)*.

Heinrich, B. and D. Hristova (2016). "A quantitative approach for modelling the influence of currency of information on decision-making under uncertainty" *Journal of Decision Systems (JDS)* 25 (1), 16–41.

Heinrich, B., M. Kaiser and M. Klier (2007a). "How to measure data quality? A metric-based approach". In: *Proceedings of the 28th International Conference on Information Systems (ICIS 2007)*.

Heinrich, B., M. Kaiser and M. Klier (2007b). "Metrics for measuring data quality - foundations for an economic data quality management". In: *Proceedings of the 2nd International Conference on Software and Data Technologies (ICSOFT 2007)*.

Heinrich, B. and M. Klier (2009). "A novel data quality metric for timeliness considering supplemental data". In: *Proceedings of the 17th European Conference on Information Systems (ECIS 2009)*.

Heinrich, B. and M. Klier (2011). "Assessing data currency—a probabilistic approach" *Journal of Information Science (JIS)* 37 (1), 86–100.

Heinrich, B. and M. Klier (2015). "Metric-based data quality assessment—Developing and evaluating a probability-based currency metric" *Decision Support Systems (DSS)* 72, 82–96.

Heinrich, B., M. Klier and M. Kaiser (2009b). "A procedure to develop metrics for currency and its application in CRM" *Journal of Data and Information Quality (JDIQ)* 1 (1), 5.

Heinrich, B., M. Klier and S. Zimmermann (2012). "Automated Planning of Process Models –Towards a Semantic-based Approach". In *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pp. 169–194: IGI Global.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models. Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

Heinrich, B. and D. Schön (2015). "Automated Planning of Context-aware Process Models". In: *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*.

Heinrich, B. and D. Schön (2016). "Automated Planning of Process Models: The Construction of Simple Merges". In: *Proceedings of the 24th European Conference on Information Systems (ECIS 2016)*.

Helmis, S. and R. Hollmann (2009). *Webbasierte Datenintegration. Ansätze zur Messung und Sicherung der Informationsqualität in heterogenen Datenbeständen unter Verwendung eines vollständig webbasierten Werkzeuges:* Springer Vieweg.

Hinrichs, H. (2002). "Datenqualitätsmanagement in Data Warehouse-Systemen". Dissertation. University of Oldenburg.

Hipp, J., M. Müller, J. Hohendorff and F. Naumann (2007). "Rule-Based Measurement Of Data Quality In Nominal Data". In: *Proceedings of the 12th International Conference on Information Quality (ICIQ 2007)*, pp. 364–378.

Hristova, D. (2014). "Considering Currency in Decision Trees in the Context of Big Data". In: *Proceedings of the 35th International Conference on Information Systems (ICIS 2014)*.

Hristova, D. (2016). "Quantitative Approaches for Modeling Information Quality in Information Systems". Dissertation. University of Regensburg.

Hu, N., L. Liu and J. J. Zhang (2008). "Do online reviews affect product sales? The role of reviewer characteristics and temporal effects" *Information Technology and Management* 9 (3), 201–214.

Hu, Y., J. Boyd-Graber, B. Satinoff and A. Smith (2014). "Interactive topic modeling" *Machine Learning* 95 (3), 423–469.

Hüner, K. M. (2011). "Führungssysteme und ausgewählte Maßnahmen zur Steuerung von Konzerndatenqualität". Dissertation. University of St. Gallen.

IBM Big Data and Analytics Hub (2016). *Extracting business value from the 4 V's of big data*. URL: http://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data (visited on 07/19/2017).

IEEE Task Force on Process Mining (2012). "Process mining manifesto". In: *Business Process Management Workshops*, pp. 169–194.

Janssen, M., H. van der Voort and A. Wahyudi (2017). "Factors influencing big data decision-making quality" *Journal of Business Research (JBR)* 70, 338–345.

Jones, E., S. P. Brown, A. A. Zoltners and B. A. Weitz (2005). "The changing environment of selling and sales management" *Journal of Personal Selling & Sales Management* 25 (2), 105–111.

Kanal, L. N. and J. F. Lemmer (2014). *Uncertainty in artificial intelligence:* Elsevier.

Kasemsap, K. (2016). "Mastering big data in the digital age". In *Effective big data management and opportunities for implementation*, pp. 104–129: IGI Global.

Kolchyna, O., T. T. P. Souza, P. Treleaven and T. Aste (2015). "Twitter sentiment analysis: Lexicon method, machine learning method and their combination" *arXiv preprint arXiv:1507.00955*.

KPMG (2016). *Now or never - 2016 Global CEO Outlook.* KPMG.

Krishnan, N. C. and D. J. Cook (2014). "Activity recognition on streaming sensor data" *Pervasive and Mobile Computing* 10, 138–154.

Kumar, V. and W. Reinartz (2016). “Creating enduring customer value” *Journal of Marketing* 80 (6), 36–68.

La Rosa, M., van der Aalst, Wil MP, M. Dumas and F. P. Milani (2017). “Business process variability modeling: A survey” *ACM Computing Surveys (CSUR)* 50 (1), 2.

Larsen, M. D. and D. B. Rubin (2001). “Iterative automated record linkage using mixture models” *Journal of the American Statistical Association (JASA)* 96 (453), 32–41.

Lau, J. H., D. Newman and T. Baldwin (2014). “Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality”. In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pp. 530–539.

Lautenbacher, F., T. Eisenbarth and B. Bauer (2009). “Process model adaptation using semantic technologies”. In: *13th Enterprise Distributed Object Computing Conference Workshops*, pp. 301–309.

Lee, O.-K., V. Sambamurthy, K. H. Lim and K. K. Wei (2015). “How does IT ambidexterity impact organizational agility?” *Information Systems Research (ISR)* 26 (2), 398–417.

Lehti, P. and P. Fankhauser (2006). “Unsupervised duplicate detection using sample non-duplicates” *Journal on Data Semantics VII*, 136–164.

Leuz, C. and P. D. Wysocki (2016). “The economics of disclosure and financial reporting regulation: Evidence and suggestions for future research” *Journal of Accounting Research* 54 (2), 525–622.

Li, D. and Y. Du (2017). *Artificial intelligence with uncertainty:* CRC press.

Linshi, J. (2014). *Personalizing Yelp star ratings: A semantic topic modeling approach.* Yelp Dataset Challenge Winner. URL: https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf (visited on 06/24/2019).

Liu, B. (2012). “Sentiment analysis and opinion mining” *Synthesis Lectures on Human Language Technologies* 5 (1), 1–167.

Liu, B. (2015). *Uncertainty theory:* Springer.

Liu, J., J. Li, W. Li and J. Wu (2016). “Rethinking big data: A review on the data quality and usage issues” *ISPRS Journal of Photogrammetry and Remote Sensing* 115, 134–142.

Liu, X., J. Gao, X. He, L. Deng, K. Duh and Y.-Y. Wang (2015). “Representation learning using multi-task deep neural networks for semantic classification and information retrieval”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Liu, Y., X. Huang, A. An and X. Yu (2007). “ARSA: a sentiment-aware model for predicting sales performance using blogs”. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 607–614.

Lu, H., Y. Li, M. Chen, H. Kim and S. Serikawa (2018). “Brain intelligence: go beyond artificial intelligence” *Mobile Networks and Applications* 23 (2), 368–375.

Lukoianova, T. and V. L. Rubin (2014). “Veracity roadmap: Is big data objective, truthful and credible?”. In: *Proceedings of the 24th ASIS SIG/CR Classification Research Workshop*.

Makridakis, S. (2017). “The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms” *Futures* 90, 46–60.

Manning, C., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky (2014). “The Stanford CoreNLP natural language processing toolkit”. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics System Demonstrations*, pp. 55–60.

Manning, C. D. and H. Schütze (1999). *Foundations of statistical natural language processing:* MIT Press.

Marrella, A. (2017). “What automated planning can do for business process management”. In: *Proceedings of the 15th International Conference on Business Process Management*.

Marrella, A. (2018). “Automated Planning for Business Process Management” *Journal on Data Semantics* 8 (2), 1–20.

McAfee, A., E. Brynjolfsson, T. H. Davenport, D. J. Patil and D. Barton (2012). “Big data: the management revolution” *Harvard Business Review* 90 (10), 60–68.

Medhat, W., A. Hassan and H. Korashy (2014). “Sentiment analysis algorithms and applications: A survey” *Ain Shams Engineering Journal* 5 (4), 1093–1113.

Mejri, A., S. Ayachi-Ghannouchi and R. Martinho (2018). “A quantitative approach for measuring the degree of flexibility of business process models” *Business Process Management Journal (BPMJ)* 24 (4), 1023–1049.

Mendling, J., H. M.W. Verbeek, B. F. van Dongen, van der Aalst, Wil MP and G. Neumann (2008). “Detection and prediction of errors in EPCs of the SAP reference model” *Data & Knowledge Engineering (DKE)* 64 (1), 312–329.

Mezzanzanica, M., M. Cesarini, F. Mercorio and R. Boselli (2012). “Towards the Use of Model Checking for Performing Data Consistency Evaluation and Cleansing”. In: *Proceedings of the 17th International Conference on Information Quality (ICIQ)*, pp. 163–177.

Migliorini, S., M. Gambini, M. La Rosa and A. H. M. ter Hofstede (2011). “Pattern-based evaluation of scientific workflow management systems”. Technical Report. Queensland University of Technology.

Miner, G., J. Elder IV, A. Fast, T. Hill, R. Nisbet and D. Delen (2012). *Practical text mining and statistical analysis for non-structured text data applications:* Academic Press.

Minnema, A., T. H. A. Bijmolt, S. Gensler and T. Wiesel (2016). “To keep or not to keep: effects of online customer reviews on product returns” *Journal of Retailing* 92 (3), 253–267.

Moges, H.-T., K. Dejaeger, W. Lemahieu and B. Baesens (2011). “Data quality for credit risk management: new insights and challenges”. In *Proceedings of the 16th International Conference on Information Quality (ICIQ 2011)*, pp. 632–646.

Moore, S. (2018). *How to Create a Business Case for Data Quality Improvement*. URL: https://www.gartner.com/smarterwithgartner/how-to-create-a-business-case-for-data-quality-improvement/ (visited on 03/25/2019).

Murphy, R. (2018). *Local Consumer Review Survey 2018*. URL: https://www.brightlocal.com/research/local-consumer-review-survey/ (visited on 06/23/2019).

Najafabadi, M. M., F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald and E. Muharemagic (2015). “Deep learning applications and challenges in big data analytics” *Journal of Big Data* 2 (1), 1.

Newman, D., J. H. Lau, K. Grieser and T. Baldwin (2010). “Automatic evaluation of topic coherence”. In: *Proceedings of the 8th North American Chapter of the Association for Computational Linguistics*, pp. 100–108.

Ngai, E. W. T., A. Gunasekaran, S. F. Wamba, S. Akter and R. Dubey (2017). “Big data analytics in electronic markets” *Electronic Markets (EM)* 27 (3), 243–245.

Nilsson, N. J. (2014). *Principles of artificial intelligence:* Morgan Kaufmann.

O’Donovan, P., K. Leahy, K. Bruton and D. T. J. O’Sullivan (2015). “Big data in manufacturing: a systematic mapping study” *Journal of Big Data* 2 (1), 20.

O’Leary, D. E. (2013). “Artificial intelligence and big data” *IEEE Intelligent Systems* 28 (2), 96–99.

Obermeier, A. (2019). “Anomaly-Based Duplicate Detection: A Probabilistic Approach”. In: *Proceedings of the International Conference on Design Science Research in Information Systems and Technology (DESRIST 2019)*, pp. 221–236.

Object Management Group (2013). *OMG Unified Modeling Language TM (OMG UML). Version 2.5*. URL: http://www.omg.org/spec/UML/2.5/Beta2/PDF (visited on 10/16/2014).

Object Management Group (2015). *OMG Unified Modeling Language TM (OMG UML). Version 2.5*. URL: http://www.omg.org/spec/UML/2.5 (visited on 05/04/2016).

Ofner, M. H., B. Otto and H. Österle (2012). “Integrating a data quality perspective into business process management” *Business Process Management Journal (BPMJ)* 18 (6), 1036–1067.

Orr, K. (1998). “Data quality and systems theory” *Communications of the ACM* 41 (2), 66–71.

Oshri, I., J. Kotlarsky and L. P. Willcocks (2015). *The Handbook of Global Outsourcing and Offshoring 3rd Edition:* Springer.

Paisley, J., C. Wang, D. M. Blei and M. I. Jordan (2015). “Nested hierarchical Dirichlet processes” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (2), 256–270.

Pak, A. and P. Paroubek (2010). “Twitter as a corpus for sentiment analysis and opinion mining”. In: *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC 2010)*.

Pan, Y. (2016). “Heading toward artificial intelligence 2.0” *Engineering* 2 (4), 409–413.

Phillips, P., S. Barnes, K. Zigan and R. Schegg (2017). “Understanding the impact of online reviews on hotel performance: an empirical analysis” *Journal of Travel Research* 56 (2), 235–249.

Pipino, L. L., Y. W. Lee and R. Y. Wang (2002). “Data quality assessment” *Communications of the ACM* 45 (4), 211–218.

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. de Clercq and others (2016). "Semeval-2016 task 5: Aspect based sentiment analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pp. 19–30.

Potnis, A. (2018). *Illuminating Insight for Unstructured Data at Scale.* IDC.

Power, D. J. (2016). "Data science: supporting decision-making" *Journal of Decision Systems (JDS)* 25 (4), 345–356.

Provost, F. and T. Fawcett (2013). "Data science and its relationship to big data and data-driven decision making" *Big Data* 1 (1), 51–59.

Pulgar, J. and M. C. Bastarrica (2017). "Transforming Multi-role Activities in Software Processes into Business Processes". In *Business Process Management Workshops: BPM 2016 International Workshops*: Springer International Publishing.

Rathore, M. M., A. Ahmad, A. Paul and S. Rho (2016). "Urban planning and building smart cities based on the internet of things using big data analytics" *Computer Networks* 101, 63–80.

Ravikumar, P. and W. W. Cohen (2004). "A hierarchical graphical model for record linkage". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 454–461.

Redman, T. C. (1996). *Data quality for the information age:* Artech House, Inc.

Regev, G., I. Bider and A. Wegmann (2007). "Defining business process flexibility with the help of invariants" *Software Process: Improvement and Practice* 12 (1), 65–79.

Rogers, B., E. Maguire and A. Nishi (2017). *The Data Differentiator. How Improving Data Quality Improves Business.* Forbes Insights.

Rosenthal, S., N. Farra and P. Nakov (2017). "SemEval-2017 task 4: Sentiment analysis in Twitter". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*.

Roy, S., A. S.M. Sajeev, S. Bihary and A. Ranjan (2014). "An empirical study of error patterns in industrial business process models" *IEEE Transactions on Services Computing* 7 (2), 140–153.

Russell, N., W. M. P. van der Aalst and A. H. M. ter Hofstede (2016). *Workflow Patterns. The Definitive Guide:* MIT Press.

Russell, S., D. Dewey and M. Tegmark (2015a). "Research priorities for robust and beneficial artificial intelligence" *AI Magazine* 36 (4), 105–114.

Russell, S., S. Hauert, R. Altman and M. Veloso (2015b). "Ethics of artificial intelligence" *Nature* 521 (7553), 415–416.

Russell, S. J. and P. Norvig (2016). *Artificial intelligence: a modern approach:* Pearson Education Limited.

Samek, W., T. Wiegand and K.-R. Müller (2017). "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models" *arXiv preprint arXiv:1708.08296*.

Scheer, A.-W. (2012). *ARIS—business process modeling:* Springer Science & Business Media.

Schouten, K. and F. Frasincar (2016). "Survey on aspect-level sentiment analysis" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 28 (3), 813–830.

Schroeck, M., R. Shockley, J. Smart, D. Romero-Morales and P. Tufano (2012). *Analytics: the real-world use of big data: How innovative enterprises extract value from uncertain data.* IBM Institute for Business Value and Said Business School at the University of Oxford. Technical Report.

Shankaranarayanan, G. and Y. Cai (2006). "Supporting data quality management in decision-making" *Decision Support Systems (DSS)* 42 (1), 302–317.

Shankaranarayanan, G., B. Iyer and D. Stoddard (2012). "Quality of Social Media Data and Implications of Social Media for Data Quality". In: *Proceedings of the 17th International Conference on Information Quality (ICIQ)*, pp. 311–325.

Shapiro, R., S. A. White, C. Bock, N. Palmer, M. zur Muehlen, Brambilla. Marco and D. Gagné (2012). *BPMN 2.0 handbook second edition. Methods, concepts, case studies and standards in business process management notation.* 2nd ed.: Future Strategies.

Shen, Y., X. He, J. Gao, L. Deng and G. Mesnil (2014). "Learning semantic representations using convolutional neural networks for web search". In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374.

Sheng, J., J. Amankwah-Amoah and X. Wang (2017). "A multidisciplinary perspective of big data in management research" *International Journal of Production Economics* 191, 97–112.

Sigaud, O. and O. Buffet (2013). *Markov decision processes in artificial intelligence:* John Wiley & Sons.

Soffer, P., Y. Wand and M. Kaner (2015). "Conceptualizing routing decisions in business processes. Theoretical analysis and empirical testing" *Journal of the Association for Information Systems (JAIS)* 16 (5), 345.

Stadtler, H., C. Kilger and H. Meyr (eds.) (2015). *Supply Chain Management and Advanced Planning:* Springer Berlin Heidelberg.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll and M. Stede (2011). "Lexicon-based methods for sentiment analysis" *Computational Linguistics* 37 (2), 267–307.

Timmer, M. P., A. A. Erumban, B. Los, R. Stehrer and G. J. de Vries (2014). "Slicing up global value chains" *Journal of Economic Perspectives* 28 (2), 99–118.

Tromp, M., A. C. Ravelli, G. J. Bonsel, A. Hasman and J. B. Reitsma (2011). "Results from simulated data sets. Probabilistic record linkage outperforms deterministic record linkage" *Journal of Clinical Epidemiology* 64 (5), 565–572.

van der Aalst, W. (2016a). "Data science in action". In *Process Mining*, pp. 3–23: Springer.

van der Aalst, W. (2016b). *Process Mining:* Springer Berlin Heidelberg.

van der Aalst, W. M. P. (2013). "Business process management: a comprehensive survey" *ISRN Software Engineering* 2013.

van der Aalst, W. M. P., M. La Rosa and F. M. Santoro (2016). "Business Process Management - Don't Forget to Improve the Process!" *Business & Information Systems Engineering (BISE)* 58 (1), 1–6.

van der Aalst, W. M. P., M. Pesic and H. Schonenberg (2009). "Declarative workflows: Balancing between flexibility and support" *Computer Science-Research and Development* 23 (2), 99–113.

van der Aalst, W. M. P. and A. H. M. ter Hofstede (2005). "YAWL. Yet another workflow language" *Information Systems* 30 (4), 245–275.

van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns" *Distributed and Parallel Databases* 14 (1), 5–51.

van Noorden, R. (2014). *Global scientific output doubles every nine years*. URL: http://blogs.nature.com/news/2014/05/global-scientific-output-doubles-every-nine-years.html (visited on 06/27/2019).

Vanwersch, R. J. B., K. Shahzad, I. Vanderfeesten, K. Vanhaecht, P. Grefen, L. Pintelon, J. Mendling, G. G. van Merode and H. A. Reijers (2016). "A critical evaluation and framework of business process improvement methods" *Business & Information Systems Engineering (BISE)* 58 (1), 43–53.

Vatsalan, D., Z. Sehili, P. Christen and E. Rahm (2017). "Privacy-preserving record linkage for big data: Current approaches and research challenges". In *Handbook of Big Data Technologies*, pp. 851–895: Springer.

vom Brocke, J. and J. Mendling (eds.) (2018). *Business Process Management Cases. Digital Innovation and Business Transformation in Practice:* Springer International Publishing.

vom Brocke, J. and M. Rosemann (eds.) (2015). *Handbook on business process management 1: Introduction, methods, and information systems:* Springer Publishing Company, Incorporated.

Wand, Y. and R. Y. Wang (1996). "Anchoring data quality dimensions in ontological foundations" *Communications of the ACM* 39 (11), 86–95.

Wang, R. Y. (1998). "A product perspective on total data quality management" *Communications of the ACM* 41 (2), 58–66.

Wang, R. Y. and D. M. Strong (1996). "Beyond accuracy: What data quality means to data consumers" *Journal of Management Information Systems (JMIS)*, 5–33.

Wang, Y., L. Kung and T. A. Byrd (2018). "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations" *Technological Forecasting and Social Change* 126, 3–13.

Watts, S., G. Shankaranarayanan and A. Even (2009). "Data quality assessment in context: A cognitive perspective" *Decision Support Systems (DSS)* 48 (1), 202–211.

Webber, B. L. and N. J. Nilsson (2014). *Readings in artificial intelligence:* Morgan Kaufmann.

Weber, B., M. Reichert and S. Rinderle-Ma (2008a). "Change patterns and change support features-enhancing flexibility in process-aware information systems" *Data & Knowledge Engineering (DKE)* 66 (3), 438–466.

Weber, I., J. Hoffmann and J. Mendling (2008b). "Semantic business process validation". In: *Proceedings of the 3rd International Workshop on Semantic Business Process Management (SBPM'08) at EWCS'08*.

Weber, I., J. Hoffmann and J. Mendling (2010). "Beyond soundness: on the verification of semantic business process models" *Distributed and Parallel Databases* 27 (3), 271–343.

Wechsler, A. and A. Even (2012). "Using a Markov-Chain model for assessing accuracy degradation and developing data maintenance policies". In: *Proceedings of the 18th Americas Conference on Information Systems (AMCIS 2012)*.

Weiss, S. M., N. Indurkhya and T. Zhang (2015). *Fundamentals of predictive text mining:* Springer.

Weske, M. (2012). *Business process management - concepts, languages, architectures, 2nd Edition:* Springer.

Wetherly, P. and D. Otter (2014). *The business environment: themes and issues in a globalizing world:* Oxford University Press.

Wetzstein, B., Z. Ma, A. Filipowska, M. Kaczmarek, S. Bhiri, S. Losada, J.-M. Lopez-Cob and L. Cicurel (2007). "Semantic Business Process Management: A Lifecycle based Requirements Analysis". In: *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) in conjunction with the 3rd European Semantic Web Conference (ESWC 2007)*.

Wilkins, D. E. (2014). *Practical planning: extending the classical AI planning paradigm:* Elsevier.

Winkler, W. E. (2006). *Overview of record linkage and current research directions.* U.S. Bureau of the Census.

Winston, P. H. (1992). *Artificial Intelligence*. *3rd Edition:* Addison-Wesley.

Witchalls, C. (2014). *Gut & gigabytes: Capitalising on the art & science in decision making.* PwC.

Xiang, Z., Z. Schwartz, J. H. Gerdes Jr and M. Uysal (2015). "What can big data and text analytics tell us about hotel guest experience and satisfaction?" *International Journal of Hospitality Management* 44, 120–130.

Ye, Q., R. Law, B. Gu and W. Chen (2011). "The influence of user-generated content on traveler behavior: An empirical investigation on the effects of e-word-of-mouth to hotel online bookings" *Computers in Human Behavior* 27 (2), 634–639.

Zak, Y. and A. Even (2017). "Development and evaluation of a continuous-time Markov chain model for detecting and handling data currency declines" *Decision Support Systems (DSS)* 103, 82–93.

Zhang, J. and B. Zhang (2014). "Clinical research of traditional Chinese medicine in big data era" *Frontiers of Medicine* 8 (3), 321–327.

Zhang, Q., L. T. Yang, Z. Chen and P. Li (2018). "A survey on deep learning for big data" *Information Fusion* 42, 146–157.

Zhou, K., C. Fu and S. Yang (2016). "Big data driven smart energy management: From big data to big insights" *Renewable and Sustainable Energy Reviews* 56, 215–225.

Zhu, J., H. Wang, M. Zhu, B. K. Tsou and M. Ma (2011). "Aspect-based opinion polling from customer reviews" *IEEE Transactions on Affective Computing* 2 (1), 37–49.

Zuiderwijk, A., R. Shinde and M. Janssen (2018). "Investigating the attainment of open government data objectives: Is there a mismatch between objectives and results?" *International Review of Administrative Sciences* (Paper ID 0020852317739115).

2 Assessment of Data Quality

This section contains three papers concerning the first focal point of the dissertation, the assessment of data quality, comprising the treatment of the research questions RQ1-RQ3. In particular, Section 2.1 covers a probability-based metric for assessing the data quality dimension semantic consistency (RQ1). In Section 2.2, a probability-based approach for duplicate detection considering the underlying causes for duplicates is proposed (RQ2). In Section 2.3, clearly defined requirements for data quality metrics to support both decision-making under uncertainty as well as an economically oriented management of data quality are suggested (RQ3).

2.1 Paper 1: Assessing Data Quality – A Probability-based Metric for Semantic Consistency

Current Status	Full Citation
accepted and published (04/2018) in Issue 110 of *Decision Support Systems*	Heinrich, B., M. Klier., A. Schiller and G. Wagner (2018). "Assessing data quality – A probability-based metric for semantic consistency". *Decision Support Systems (DSS)* 110, 95-106.

Summary

This paper addresses RQ1 by proposing a novel probability-based metric for semantic consistency. For the assessment, the metric uses uncertain rules, taking into account the probability with which a rule is expected to be fulfilled and applying these rules to the data to be analyzed. The more the actual rule fulfillment deviates from the expected rule fulfillment, the less likely is the semantic consistency of the data. More precisely, the determined metric value represents the probability that the data to be assessed is free of internal contradictions with respect to the uncertain rules. A formal metric definition and different possibilities for the instantiation of the metric are presented. The practical applicability and effectiveness of the metric are evaluated in a real-world setting, analyzing a customer dataset of an insurance company and identifying a serious consistency issue subsequently acknowledged by the insurer. Further analyses indicate the economic efficiency of the metric and reveal that in contrast to the presented approach, existing metrics for consistency are not able to determine the consistency issue in the insurer's data.

The work builds heavily on concepts and methods from probability theory, in particular employing statistical tests, the concepts of expected value and p-value as well as Bernoulli-distributed random variables to quantify uncertainty and engage decision-making under uncertainty. The metric itself is defined as the two-sided p-value of a binomial distribution. This enables the interpretation of the metric values as probabilities, which in turn facilitates well-founded support for data-driven decision-making and, in particular, their integration into expected value calculus. Applying the metric to customer data allows to identify a specific consistency issue and, due to the interpretability of the metric values, to pinpoint which records are problematic (i.e., probably erroneous) and which ones to treat as trustworthy. In this way, future data-driven decision-making by the insurer is supported, for instance by improved targeting in customer campaigns.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style.

The paper as published by Elsevier is available at: https://doi.org/10.1016/j.dss.2018.03.011

Abstract:

We present a probability-based metric for semantic consistency using a set of uncertain rules. As opposed to existing metrics for semantic consistency, our metric allows to consider rules that are expected to be fulfilled with specific probabilities. The resulting metric values represent the probability that the assessed dataset is free of internal contradictions with regard to the uncertain rules and thus have a clear interpretation. The theoretical basis for determining the metric values are statistical tests and the concept of the p-value, allowing the interpretation of the metric value as a probability. We demonstrate the practical applicability and effectiveness of the metric in a real-world setting by analyzing a customer dataset of an insurance company. Here, the metric was applied to identify semantic consistency problems in the data and to support decision-making, for instance, when offering individual products to customers.

Keywords: data quality, data quality assessment, data quality metric, data consistency

1 Introduction

Making use of large amounts of internal and external data becomes increasingly important for companies to gain competitive advantage and enable data-driven decisions in businesses (Ngai et al., 2017). However, data quality problems still impede companies to generate the best value from data (Moges et al., 2016; Witchalls, 2014). Overall, poor data quality amounts to an average financial impact of $9.7 million per year and organization as reported by recent Gartner research (Moore, 2018). In particular, 63% of the respondents of a survey by Moges et al. (2011, p. 639) indicated that "inconsistency (value and format) and diversity of data sources are main recurring challenges of data quality".

Data quality can be defined as the "agreement between the data views presented by an information system and that same data in the real world" (Orr, 1998, p. 67). In this regard, data quality is a multidimensional construct comprising different dimensions such as accuracy, consistency, completeness, and currency (Batini and Scannapieco, 2016; Redman, 1996; Zak and Even, 2017). In the following, we focus on consistency, in particular semantic consistency, as one of the most important dimensions (Blake and Mangiameli, 2011; Shankaranarayanan et al., 2012; Wand and Wang, 1996). We define semantic consistency as the degree to which assessed data is free of internal contradictions (cf. also Batini and Scannapieco, 2016; Heinrich et al., 2007; Redman, 1996).

Contradictions are usually determined based on a set of rules (Batini and Scannapieco, 2006; Heinrich et al., 2007; Mezzanzanica et al., 2012). Thereby, a rule represents a proposition consisting of two logical statements, where the first statement (antecedent) implies the second (consequent). For instance, in a database containing master data about customers in Western Europe, such a rule may be $year\ of\ birth = 2003 \rightarrow marital\ status = single$. Stored customer data regarding a married customer born in 2003 would contradict this rule, indicating a consistency

problem.

Existing data quality metrics for semantic consistency are based on rules which are considered as "true by definition" (cf. Section 2). This means that the rules have to be true for all of the assessed data and any violation indicates inconsistent data. Examples for such rules are provided in Figure 1, which also shows some selected records of a customer database serving as a basis for our discussion:

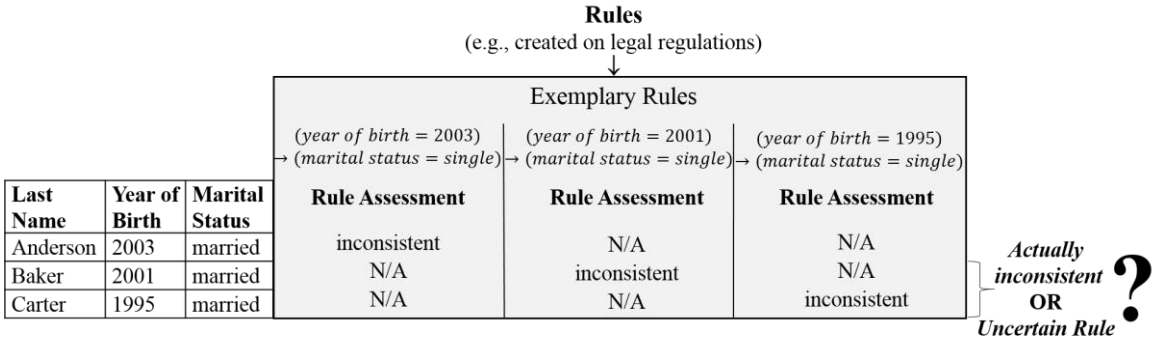

Figure 1. "True by Definition" Rules used for assessing Semantic Consistency

Due to the fact that in Western Europe marriage is only legally allowed for people of age 16 and older, for an assessment in the year 2018, a value for *year of birth* of 2003 (antecedent of the first rule in Figure 1) implies the value *single* for *marital status* (consequent of the first rule), which is a typical example for a "true by definition" rule. In this case, the value *married* for *marital status* of the first record is assessed as inconsistent. However, for the assessment of semantic consistency it can be necessary to also consider rules such as $year\ of\ birth = 2001 \rightarrow marital\ status = single$ (second rule in Figure 1). Here, one has to distinguish: On the one hand, violations of such a rule – assessed in 2018 – may indeed indicate an erroneous value which could have resulted from a random or systematic data error (cf. Alkharboush and Yuefeng Li, 2010; Fisher et al., 2009). On the other hand, violations may stem from the fact that the rule is not "true by definition", but only fulfilled with a specific probability. For example, some customers may indeed have married at the age of 16. Therefore, a violation of this rule does not necessarily imply that such data is inconsistent and of low quality. This also holds for other years of birth (e.g., $year\ of\ birth = 1995$; third rule in Figure 1) or, in general, for other antecedents and consequents or applications where rules cannot be considered as "true by definition". Hence, we are confronted with rules with uncertain consequent, to which we refer as uncertain rules in the following. To the best of our knowledge, none of the existing approaches aiming to measure semantic consistency has considered such relevant uncertain rules yet.

Thus, to (1) consider uncertain rules in a well-founded way and (2) ensure a clear interpretation of the resulting metric values, we propose a data quality metric for semantic consistency based on probability theory. To address uncertain rules, the metric delivers an indication rather than

a statement under certainty regarding the degree to which assessed data is free of internal contradictions. We argue that the well-founded methods of probability theory are adequate and valuable to deal with uncertain rules. More precisely, the theoretical basis for determining the metric values are statistical tests and the concept of the p-value, allowing the clear interpretation of the metric values as probabilities.

The remainder of the paper is structured as follows. In the next section, we discuss related work and the research gap. Then, we present a probability-based metric for semantic consistency and outline possible ways to instantiate this metric. In the fourth section, we illustrate the case of an insurance company to demonstrate the practical applicability and effectiveness of the metric. Finally, we briefly summarize the findings and conclude with a discussion of limitations and directions for further research.

2 Related Work

The data quality dimension consistency is seen "as a multi-faceted dimension" (Blake and Mangiameli, 2009, p. 3) which can be defined in terms of representational consistency, integrity, and semantic consistency (Blake and Mangiameli, 2009). Since these three aspects stem from different domains, they overlap in some cases. Representational consistency requires that data are "presented in the same format and are compatible with previous data" (Blake and Mangiameli, 2009; Wang and Strong, 1996). Integrity is often defined as entity, referential, domain, column, and user-defined integrity (Blake and Mangiameli, 2009; Lee et al., 2004). Entity integrity requires that data values considered as primary keys are unique and different from NULL. Referential integrity states that, given two relations, if an attribute is a primary key in one of them and is contained as a foreign key in the other one, the non-NULL data values from the second relation must be contained in the first one (Lee et al., 2004). Domain and column integrity require data values to be part of a predefined domain (e.g., $income \in \mathbb{R}^{+}$) and user-defined integrity requires the satisfaction of a set of general rules. Finally, semantic consistency refers to the absence of contradictions between different *data values* based on a rule set (Blake and Mangiameli, 2009; Heinrich et al., 2007; Lee et al., 2006; Liu and Chi, 2002; Mecella et al., 2002; Mezzanzanica et al., 2012; Redman, 1996; Scannapieco et al., 2005). Generally, semantic consistency is equivalent to user-defined integrity.

In this paper, we focus on semantic consistency due to two reasons. First, assuring semantic consistency is crucial for decision support, as decision-making is typically based on data values. Second, both representational consistency and integrity have already been extensively studied in literature (Blake and Mangiameli, 2009, 2011). Semantic consistency, however, is a field of research which gains more and more importance in the course of growing data volumes and their thorough analysis.

Underlining this importance, literature discusses several data quality problems and root causes which lead to inconsistencies with respect to data values (Kim et al., 2003; Laranjeiro et al.,

2015; Oliveira et al., 2005; Rahm and Do, 2000; Singh and Singh, 2010). These root causes are typically categorized in two ways. First, referring to the steps in the data management process (i.e., data entry/capturing, data transformation, data aggregation, data processing, etc.). And second, whether inconsistencies are caused by a single source or by multiple sources. Given this, a common and highly relevant root cause for inconsistencies are error-prone operative data entries via one single source (cf. Rahm and Do, 2000; Singh and Singh, 2010). This may be, for example, a call center employee, the person himself referred to in the considered record (e.g., a customer entering master data via a web application) or a damaged data capturing device (e.g., a malfunctioning sensor). In all these scenarios, inconsistencies regarding, for instance, two data values of a customer record may arise. In the case of a call center employee or the customer himself, it is possible that only one of the two data values is correctly entered or changed. The second data value, however, may be entered or changed erroneously (or not at all). For instance, the value for $year\ of\ birth$ may be correctly entered as 2003, the value for $marital\ status$, however, may be erroneously entered as $married$. Similarly, parts of a customer's address may be entered incorrectly, leading to an inconsistency. A second prevalent root cause concerns the steps data aggregation and integration in the data management process with respect to multiple sources (e.g., different databases; cf. Rahm and Do, 2000; Singh and Singh, 2010). Here, contradictory data values of, for instance, customer records may arise in scenarios in which the same customers are stored in multiple databases of departments and units of a company (e.g., after a merger). Contradictions may result from the integration of attributes or their values, for example when databases are integrated for a coordinated and comprehensive customer management. For instance, in one database, the $marital\ status$ of a customer may be stored as $single$, but in a second database, the value for $name\ of\ spouse$ of the same customer may not be equal to NULL, indicating that the customer is $married$. Faulty business rules used for data transformation and leading to contradicting data values (cf. Singh and Singh, 2010) constitute another important scenario and root cause among many others, stressing the relevance of semantic consistency.

In the following, for reasons of simplicity, we will use the term consistency instead of semantic consistency. To provide an overview of existing works on metrics for consistency, we concentrate on metrics that are (i) formally defined (e.g., by a closed-form mathematical function) and (ii) result in a numerical metric value representing the consistency of the data values to be assessed. In that sense, we do not consider approaches that aim to identify potentially (in)consistent data values without providing numerical metric values for (in)consistency (e.g., Bronselaer et al., 2016; Fan et al., 2013; Mezzanzanica et al., 2012). Table 1 presents existing metrics for consistency satisfying (i) and (ii). They follow the idea that consistency of data values can be determined based on the number of fulfilled rules, with a higher number of fulfilled rules implying higher consistency.

We discuss these metrics with regard to (1) the way they assess consistency and (2) the interpretation of the resulting metric values. Related to (1) the first three rows of Table 1 with the

light grey background contain metrics that assign weights to the fulfillment and violation of rules. The next two rows with the white background provide metrics assessing consistency purely as "true" or "false" regarding the fulfillment and violation of rules. The last two rows with the dark grey background contain metrics relying on conditional functional dependencies (CFDs; Bohannon et al., 2007; Cong et al., 2007).

		Source	Metric
"True by definition" rules	Assign weights to the fulfillment/ violation of rules	Alpar and Winkelsträter (2014); Hipp et al. (2001); Hipp et al. (2007)	t: record; N: number of relevant rules for t; L: number of irrelevant rules for t; w_n^-, w_n^+, w_l^0: weights; $r_n(t) = \begin{cases} 0, \text{if } t \text{ fulfills rule } r_n \\ 1 \text{ else} \end{cases}$; $h_n(t) = \begin{cases} 0, \text{if rule } r_n \text{ is relevant for } t \\ 1 \text{ else} \end{cases}$; $cons(t) = \sum_{n=1}^{N}(w_n^- r_n(t) + w_n^+(1 - r_n(t)))(1 - h_n(t)) + \sum_{l=1}^{L} h_l(t)\, w_l^0$
		Hinrichs (2002)	g: data value; N: number of relevant rules for g; w_n: weights; $r_n(g) = \begin{cases} 0, \text{if } g \text{ fulfills rule } r_n \\ 1 \text{ else} \end{cases}$; $cons(g) = \frac{1}{\sum_{n=1}^{N} w_n r_n(g) + 1}$
		Kübart et al. (2005)	t: record; N: number of relevant rules for t; $w_n^- \geq 0$: weights; $r_n(t) = \begin{cases} 0, \text{if } t \text{ fulfills rule } r_n \\ 1 \text{ else} \end{cases}$; $incons(t) = \sum_{n=1}^{N} w_n^- r_n(t)$
	Assess only fulfill-ment/ violation of rules	Cordts (2008); Pipino et al. (2002)	g: data value; N: number of relevant rules for g; $r_n(g) = \begin{cases} 0, \text{if } g \text{ fulfills rule } r_n \\ 1 \text{ else} \end{cases}$; $cons(g) = 1 - \frac{\sum_{n=1}^{N} r_n(g)}{N}$
		Heinrich et al. (2007); Heinrich and Klier (2015a)	g: data value; N: number of relevant rules for g; $r_n(g) = \begin{cases} 0, \text{if } g \text{ fulfills rule } r_n \\ 1 \text{ else} \end{cases}$; $cons(g) = \prod_{n=1}^{N}(1 - r_n(g))$
	Use CFDs	Abboura et al. (2016)	a: attribute; N: number of relevant CFDs for a $cons(a) = \prod_{n=1}^{N} support(r_n) \cdot conf(r_n)$
		Wang et al. (2016)	D_B: database; S: number of tuples in D_B; $C_{min}(D_B)$: minimum set of tuples in D_B such that $D_B \backslash C_{min}(D_B)$ fulfills all CFDs $incons(D_B) = \frac{\lvert C_{min}(D_B) \rvert}{S}$

Table 1. Existing Metrics for Consistency

All metrics in the first three rows of Table 1 (Alpar and Winkelsträter, 2014; Hinrichs, 2002; Hipp et al., 2001; Hipp et al., 2007; Kübart et al., 2005) assign weights to the fulfillment and violation of rules. The considered rules correspond to association rules. For a given set of records, association rules are implications of the form $X \rightarrow Y$ that satisfy specified constraints regarding minimum support and minimum confidence (cf. Agrawal et al., 1993; Srikant and Agrawal, 1996). Thereby, rule support $supp(X \rightarrow Y)$ is defined as the fraction of records that fulfill both antecedent X and consequent Y of the rule; rule confidence $conf(X \rightarrow Y)$ denotes the fraction of records fulfilling the antecedent X that also fulfill the consequent Y (cf. Agrawal et al., 1993). An example of an association rule is $year\ of\ birth = 1995 \rightarrow marital\ status = single$. If 80% of the records in the database with $year\ of\ birth = 1995$ also fulfill $marital\ status = single$ and 5% of the records fulfill both $year\ of\ birth = 1995$ and $marital\ status = single$, it follows that the support of the rule is 5% while its confidence is 80%. To treat the violation of distinct association rules r_n as differently severe when assessing consistency, for example Alpar and Winkelsträter (2014) and Hipp et al. (2007) use the rule confidence $conf(r_n)$ to determine respective weights. In particular, the idea of these authors is to assign a weight of $conf(r_n)$ to the fulfillment of a rule and a weight of $-conf(r_n)$ to its violation. In order to determine consistency concerning several rules and a set of data values, the weights are calibrated and summed up.

While these approaches treat the violation of distinct rules as differently severe (i.e., depending on rule confidence), they assess the fulfillment of a rule to always be an indicator for high consistency by assigning a positive weight (i.e., $conf(r_n)$) to rule fulfillments and vice versa. As only association rules above a chosen minimum threshold for confidence based on the dataset to be assessed are determined, rules below this threshold remain unconsidered in these approaches. However, rules with a lower confidence are also highly relevant for assessing consistency, as they can be an important indicator for inconsistent data. For example, a rule (confidence) stating that 30% of 17-year-olds are stored as being married would certainly help to identify inconsistencies because a much smaller percentage of 17-year-olds is actually married in Western Europe. In addition, solely using the rule confidence based on the assessed data can lead to misleading results if a large part of the data to be assessed is erroneous: For instance, if 90% of all 17-year-olds are erroneously stored as being married in a database, a corresponding association rule and its rule confidence is determined (given a minimum rule confidence of e.g. 80%). On this basis, however, the 10% of 17-year-olds which are accurately stored as *not* being married would be considered as inconsistent. More generally, these approaches assess all rules with high confidence as "true by definition" and penalize violations against them as inconsistent.

To conclude, these metrics provide first, promising steps concerning the treatment of violations of distinct rules as differently severe. However, rules with low confidence are ignored and rules with high confidence are seen as "true by definition". Further, the resulting values of these metrics suffer from a lack of clear interpretation (cf. (2)). Indeed, it remains unclear what a

particular metric value actually means, obstructing its use for decision support. This is due to the summation of the (calibrated) weights (representing the rule confidences as "measures of consistency"). To illustrate this, we again consider the example of a customer database. A customer record may fulfill some association rules (e.g., the values for *zip code* and *city*) and violate others (e.g., the values for *marital status* and *year of birth)*. The respective calibrated weights are summed up, but the result of the summation is a real number with no clear interpretation (e.g., in terms of a probability whether the considered record is consistent). Furthermore, the metric values are, in general, not interval-scaled and do not have a defined minimum and maximum. This may seriously hinder their usefulness for decision support: For example, in a second assessment of the customer data at a later point in time, the mined association rules and their confidence can differ from the first assessment. Then, a higher (or lower) metric value of the same, unchanged record in the second assessment does not necessarily represent higher (or lower) actual consistency. In fact, the consistency of the record may still be the same.

The metrics in the next two rows of Table 1 with the white background (Cordts, 2008; Heinrich et al., 2007; Heinrich and Klier, 2015a; Pipino et al., 2002) assess the consistency of data values only by "true" or "false" statements regarding the fulfillment and violation of the considered rules. On this basis, they provide a clear interpretation of the metric values in terms of the percentage of data values consistent with respect to the considered rules (cf. (2)). These approaches, however, treat all rules equally as "true by definition" and thus have similar limitations as the metrics discussed above.

Finally, the metrics provided in the last two rows of Table 1 with the dark grey background (Abboura et al., 2016; Wang et al., 2016) assess consistency by using CFDs. A CFD is a pair $(X \rightarrow Y, T_i)$ consisting of a functional dependency $X \rightarrow Y$ (an implication of sets of attributes) and a certain tableau T_i (with $i \in \{1, 2, \dots, N\}$) which specifies values for the attributes in X and Y (cf. Bohannon et al., 2007 for details). To give an example, stating that records with $year\ of\ birth = 1995$ also fulfill $marital\ status = single$ can be represented by the following CFD: $(year\ of\ birth \rightarrow marital\ status, T_1)$, with T_1 containing a row which includes 1995 as value for $year\ of\ birth$ and $single$ as value for $marital\ status$. A probabilistic CFD is a pair consisting of a CFD and its confidence, where support and confidence of a probabilistic CFD are defined analogously to association rules (Golab et al., 2008). Abboura et al. (2016) define the consistency of an attribute to be the product of the support of a (probabilistic) CFD multiplied by its confidence. The product is taken over all CFDs relevant for the considered attribute. Thus, analogous to the approaches in the first three lines of Table 1, the approach assesses the considered CFDs as "true by definition" and penalizes violations against them as inconsistent. This results in similar problems as outlined above. Additionally, the metric values do not provide a clear interpretation (cf. (2)). Wang et al. (2016) propose to determine a minimum subset of tuples in a database which – if corrected – would lead to the database fulfilling all CFDs. Then, the inconsistency of the database is measured by the ratio of the size of this minimum subset in relation to the size of the whole database.

Overall, existing metrics interpret their rules used for assessing consistency as "true by definition" resulting in several limitations. In particular, they do not deal with uncertain rules. Moreover, metrics which treat the violation of distinct rules as differently severe do not ensure a clear interpretation of the metric values. In the next section we address this research gap.

3 Probability-based Metric for Consistency

In this section, we present our metric for semantic consistency. First, we outline the general setting and the basic idea. Then, we describe methodological foundations which serve as a basis when defining the metric in the following subsection. Finally, we outline possible ways to instantiate the metric.

3.1 General Setting and Basic Idea

We consider the common relational database model and a database D_B to be assessed. A relation consists of a set of attributes $\{a_1, a_2, \dots, a_m\}$ and a set of records $T = \{t_1, t_2, \dots, t_n\}$. The data value of record t_j regarding attribute a_i is denoted by $\phi(t_j, a_i)$. In line with existing literature (cf. Section 2), we use a rule set R to assess consistency. Rules are propositions of the form $r: A \rightarrow C$, where A (antecedent) and C (consequent) are logical statements addressing either single attributes in D_B or relations between them. As opposed to existing approaches, we do not treat rules as "true by definition". Rather, we aim to consider uncertain rules that are expected to be fulfilled with specific probabilities.

This allows to determine metric values which represent the probability that the assessed dataset is free of internal contradictions with regard to these uncertain rules. More precisely, for a data value $\phi(t_j, a_i)$ in D_B and an uncertain rule r, we interpret consistency as the probability that $\phi(t_j, a_i)$ is free of contradictions with regard to r. A metric that results in a probability guarantees that the metric takes values in $[0;\ 1]$ and the metric values have a clear interpretation.

The following running example from our application context (cf. Section 4) illustrates the idea of our metric: An insurer strives to conduct a product campaign targeting only married customers younger than 20 years. If the data stored in the customer database is of low quality, wrong decisions and economic losses may result. For instance, if a customer younger than 20 years is erroneously stored as *married* in the database, contacting him with a product offer will generate costs and may lead to lower customer satisfaction. In case the insurer aims to assess the consistency of its customer database before conducting the campaign, existing metrics for consistency would consider the rule $r_1: year\ of\ birth > 1998 \rightarrow marital\ status = single$. This rule is selected because it is fulfilled by most people that are younger than 20 years (e.g., 95%), which goes along with a high rule confidence. However, such metrics would assess data regarding a married customer who is younger than 20 years as inconsistent. Thus, the determined metric values could not provide any support within the campaign.

Our metric, in contrast, additionally considers the rule r_2: $year\ of\ birth > 1998 \rightarrow marital\ status = married$ and the probabilities with which r_1 and r_2 are expected to be fulfilled (e.g., based on census data). In particular, our approach evaluates the actual fulfillment of r_1 and r_2 in the customer database in comparison to the expected distribution of rule fulfillment. For example, the number of married people that are younger than 20 years is generally low, meaning that r_2 is expected to be fulfilled only with a low frequency (e.g., 4.1%). Thus, if r_2 is fulfilled similarly infrequently in the customer database (e.g., 4.2%), the corresponding data of married customers is assessed to have a high probability of being consistent.

This interpretation of metric values as probabilities is viable because statistical tests and the concept of the p-value form the methodological foundation for determining the metric values (cf. Section 3.2). Moreover, by assessing consistency as a probability, the metric values for each customer can be integrated in decision support, for instance, into the calculation of expected values. Such a calculation may reveal that targeting a married customer younger than 20 years within the campaign is only beneficial if the consistency of the data of this customer – represented by a probability – is greater than 0.8. Thus, applying the rule r_2, the metric can be used to determine whether this threshold is met (note that this threshold is totally different from rule confidence, as confidence of r_2 is only 4.2%).

3.2 Methodological Foundations

3.2.1 Uncertain Rules

A rule r: $A \rightarrow C$ consists of logical statements A and C, with A and C describing single attributes or relations between different attributes in D_B. The simplest form of a logical statement S is defined as (Chiang and Miller, 2008; Fan et al., 2013):

$$\begin{array}{l} < attribute >< operator >< attribute > \\ \text{or} \\ < attribute >< operator >< constant > \end{array} \quad (1)$$

Here, $< attribute >$ is one of the attributes a_i and $< operator >$ is a binary operator such as $=,\ \geq,\ >,\ \neq$ or $substring\_of$. Simple logical statements can be linked by conjunction (AND, ∧), disjunction (OR, ∨) or negation (NOT, ¬) to form more complex logical statements. For instance, in the running example, we may have a rule of the following form:

$$r_3: year\ of\ birth > 1998\ \wedge gender = female \rightarrow marital\ status = single \quad (2)$$

Here, $year\ of\ birth > 1998$, $gender = female$, $marital\ status = single$, and $year\ of\ birth > 1998\ \wedge gender = female$ are logical statements. To determine whether a logical statement S is true or false for a record t of D_B, it can be applied to t by replacing each attribute a_i contained in S by $\phi(t, a_i)$. In other words, the corresponding data values of the record are inserted. We further define the set of records in D_B rendering S true as $fulfilling\ records(D_B, S) \coloneqq \{t \in T |\ S(t) \text{ is } true\}$.

As an example, we can apply the antecedent $year\ of\ birth > 1998 \wedge gender = female$ and the consequent $marital\ status = married$ of the rule r_3 to a record t of the database D_B with $\phi(t, year\ of\ birth) = 2000$, $\phi(t, gender) = female$ and $\phi(t, marital\ status) = married$. As $2000 > 1998$, $female = female$ and $married = married$, it follows $A(t)\ true$ and $C(t)\ true$. Thus, $t \in fulfilling\ records(D_B, A)$ and $t \in fulfilling\ records(D_B, C)$.

We call a rule $r: A \rightarrow C$ *relevant* for a record $t \in T$ if $t \in fulfilling\ records(D_B, A)$. If r is relevant for t, we say that t *fulfills* r, if $t \in fulfilling\ records(D_B, A \wedge C)$, and that t *violates* r otherwise. As mentioned above, we consider uncertain rules and not just rules which are "true by definition". To be more precise, an *uncertain rule* in our context is defined as:

$$(r: A \rightarrow C, p(r)) \tag{3}$$

An uncertain rule $(r: A \rightarrow C, p(r))$ has two components. It comprises a rule r containing the logical statements A (antecedent) and C (consequent) as well as a number $p(r) \in [0;\ 1]$ representing the probability with which r is expected to be fulfilled. The probability $p(r)$ allows to specify the uncertainty of the rule r. In contrast to existing approaches, this allows to consider rules that are unlikely to be fulfilled as well as almost certain rules or rules which are "true by definition" (i.e., the special case $p(r) = 1$) for the assessment of consistency. It is different from the confidence of an association rule as it is not based on the relative frequency of rule fulfillment in the dataset to be assessed. Moreover, the probability $p(r)$ is not used for selecting rules (e.g., with a high probability of being fulfilled), but rather for assessing consistency (the determination of uncertain rules will be outlined in Section 3.4.1).

3.2.2 Using Uncertain Rules for the Assessment of Consistency

Let $r: A \rightarrow C$ be a rule in the rule set R and let D_B be the dataset to be assessed. The rule r is expected to be fulfilled with probability $p(r)$. Hence, if the records in $fulfilling\ records(D_B, A)$ are consistent with regard to r, the application of r to such a record t can be seen as a Bernoulli trial with success probability $p(r)$, where success is defined as $t \in fulfilling\ records(D_B, A \wedge C)$. This is, because applying r to a record in $fulfilling\ records(D_B, A)$ has only two possible outcomes: The rule can either be fulfilled (with probability $p(r)$) or violated (with probability $1 - p(r)$). Thus, the Bernoulli trial can be represented by a random variable $r(t)$ resulting in $r(t) \sim Bern(p(r))$:

$$r(t) := \begin{cases} 1 & \text{if } t \in fulfilling\ records(D_B, A \wedge C) \\ 0 & \text{if } t \notin fulfilling\ records(D_B, A \wedge C), t \in fulfilling\ records(D_B, A) \end{cases} \tag{4}$$

Similarly, r can then be applied to all records t in D_B with $t \in fulfilling\ records(D_B, A)$ and the results can be summed up by the random variable $X(r) := \sum_{t \in fulfilling\ records(D_B, A)} r(t)$. As a sum of independent Bernoulli-distributed random variables, $X(r)$ follows a binomial distribution with parameters $|fulfilling\ records(D_B, A)|$ and $p(r)$:

$X(r) \sim B(|fulfilling\ records(D_B, A)|, p(r))$. An illustration for such a distribution with parameters 100 and 0.5 is presented in Figure 2.

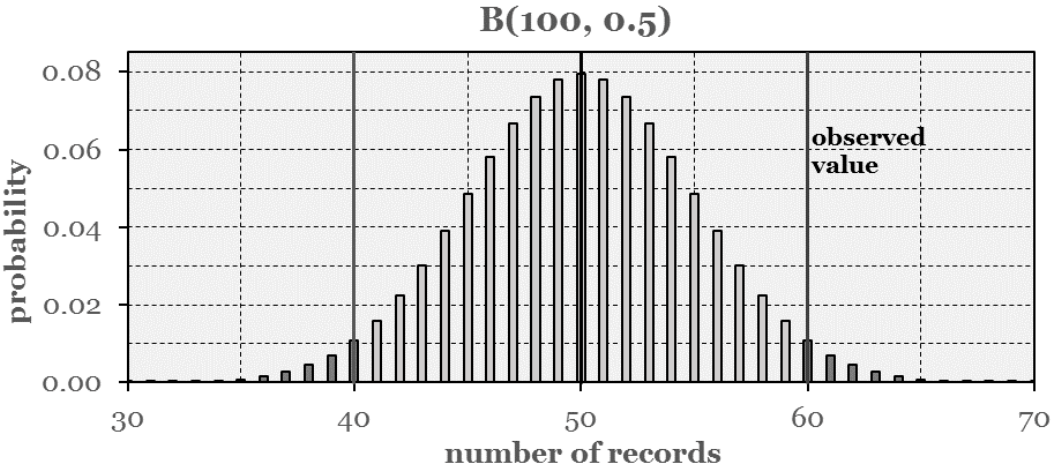

Figure 2. Binomial Distribution

If the records in $fulfilling\ records(D_B, A)$ are consistent with regard to r and $p(r)$, it follows that $|fulfilling\ records(D_B, A \wedge C)|$ is distributed as the successes of $X(r)$. Thus, to determine the consistency of the records in $fulfilling\ records(D_B, A)$, the actual value of $|fulfilling\ records(D_B, A \wedge C)|$ is contrasted with the distribution of $X(r)$. In Figure 2, we observe $|fulfilling\ records(D_B, A \wedge C)| = 60$ and expected value $E[X(r)] = 50$, resulting in an indication of inconsistency.

Based on this idea, we develop a probability-based metric for consistency founded on the well-known concept of the (two-sided) p-value in hypothesis testing. Let $p'(r)$ be the relative frequency with which the rule r is fulfilled by a relevant record in the dataset D_B. If the relevant records are consistent with regard to r, then $p'(r)$ should correspond to $p(r)$ (e.g., 0.5 in Figure 2). Thus, in statistical terms, measuring consistency implies testing the null hypothesis $H_0: p'(r) = p(r)$ against the alternative hypothesis $H_1: p'(r) \neq p(r)$ for the binomially distributed random variable $X(r) \sim B(|fulfilling\ records(D_B, A)|, p(r))$. A two-sided alternative hypothesis is used because both too many and too few fulfillments of r indicate inconsistency: The more $|fulfilling\ records(D_B, A \wedge C)|$ deviates from $E[X(r)]$, the more the consistency of D_B decreases in regard to r.

This intuitive understanding is formalized by the two-sided p-value. It represents the probability that a value occurs under the null hypothesis which is equal to or more extreme than the observed value. For example, in Figure 2, $E[X(r)] = 50$ and observed value $|fulfilling\ records(D_B, A \wedge C)| = 60$. Since the distribution is symmetric, values ≥ 60 and values ≤ 40 are equal to or more extreme than the observed value. Following this, the two-sided p-value is calculated by summing up the probabilities $p(X(r) \geq 60)$ and $p(X(r) \leq 40)$, represented by the dark grey bars.

In our case, the observed value is $|fulfilling\ records(D_B, A \wedge C)|$ and the expected value is $E[X(r)]$. Thus, the p-value represents the probability that, under the null hypothesis, the random variable $X(r)$ yields a value equal to or more extreme than $|fulfilling\ records(D_B, A \wedge C)|$. Hence, it represents the probability that the assessed records in D_B are free of contradictions with regard to the rule r. The two-sided p-value of the random variable $X(r) \sim B(|fulfilling\ records(D_B, A)|, p(r))$ with respect to the observed value $|fulfilling\ records(D_B, A \wedge C)|$ is denoted as follows:

$$p\text{-}value(X(r) \sim B(|fulfilling\ records(D_B, A)|, p(r)), |fulfilling\ records(D_B, A \wedge C)|) \quad (5)$$

Note that we are aware of the discussion regarding the p-value (cf., e.g., Goodman, 2008) and since this is not the main focus of our paper, we follow the above standard interpretation. The outlined methodological foundations allow for a formal definition of our metric in the next subsection and ensure a clear interpretation of the metric values.

3.3 Definition of the Metric for Consistency

Let D_B be a database, $t_j \in T$ be a record in D_B, a_i be an attribute in D_B, and $r: A \rightarrow C$ with $p(r) \in [0; 1]$ be an uncertain rule such that a_i is part of r and $t_j \in fulfilling\ records(D_B, A \wedge C)$. We define the consistency of the data value $\phi(t_j, a_i)$ with regard to r as:

$$Q_{Cons}(\phi(t_j, a_i), r: A \rightarrow C) := p\text{-}value(X(r) \sim B(|fulfilling\ records(D_B, A)|, p(r)), |fulfilling\ records(D_B, A \wedge C)|) \quad (6)$$

This definition ensures that only attributes which are part of the antecedent or consequent and records which fulfill the rule are considered. The metric value $Q_{Cons}(\phi(t_j, a_i), r: A \rightarrow C)$ represents the probability that, if the relevant records are consistent with regard to r, the random variable $X(r)$ yields a value which is equal to or more extreme than $|fulfilling\ records(D_B, A \wedge C)|$.

The metric in Definition (6) measures consistency with regard to a single rule. If multiple rules can be used to assess the consistency of a specific data value, these rules can be aggregated for the assessment. This can be achieved by using conjunctions (AND, ∧). For example, let $r_1: A_1 \rightarrow C_1$ and $r_2: A_2 \rightarrow C_2$ be two rules available for the assessment of the data value $\phi(t_j, a_i)$. Then, it holds that $t_j \in fulfilling\ records(A_1 \wedge C_1 \wedge A_2 \wedge C_2)$ while a_i is part of A_1 or C_1 and part of A_2 or C_2, respectively. Thus, instead of the single rules r_1 and r_2, the aggregated rule $r_3: A_1 \wedge A_2 \rightarrow C_1 \wedge C_2$ can be considered and used to assess consistency in a well-founded manner by means of Definition (6). Analogously, an iterative aggregation can be applied if more than two rules are available.

Definition (6) allows the identification of data values which are likely to be inconsistent due to both random and systematic data errors. On the one hand, random data errors may lead to erroneous data values, thus contradicting a rule in the rule set R. On the other hand, systematic data

errors may occur which usually bias the data values "in one direction" and thus cause $|fulfilling\ records(D_B, A \wedge C)|$ to differ considerably from $E[X(r)]$ for a rule $r: A \rightarrow C$ in R. Thus, for both random and systematic data errors, the considered p-value is low. As a result, both types of errors lead to low metric values indicating inconsistency of the corresponding data values with regard to R.

The metric in Definition (6) assesses consistency on the level of data values. On this basis, aggregated metric definitions for records, attributes, relations, and the whole database D_B can be determined. To do so, the weighted arithmetic mean of the metric values of the corresponding data values can be used similarly to, for example, Heinrich and Klier (2011). This allows the assessment of consistency on different data view levels and to support decisions relying on, for instance, the consistency of D_B as a whole.

3.4 Metric Instantiation

In this subsection, we describe how to instantiate our metric. In particular, we describe how uncertain rules can be obtained and how the metric values can be calculated.

3.4.1 Obtaining Uncertain Rules

For the application of the metric, it is crucial to determine an appropriate set of uncertain rules R and the corresponding values $p(r)$ for each uncertain rule $r \in R$. Generally, there are different possibilities to determine this rule set. We briefly describe the following three ways: (i) Analyzing a reference dataset, (ii) Conducting a study, and (iii) Surveying experts.

Ad (i): A promising option is to use a quality assured reference dataset D_R (in case such exists). This reference dataset D_R needs to be representative for the data of interest in D_B to allow the determination of meaningful uncertain rules. Such a reference dataset may, for example, be reliable historical data owned by the organization itself. With more and more external data being provided by recent open data initiatives, reliable publicly available data from public or scientific institutions (e.g., census data, government data, data from federal statistical offices and institutes) can be analyzed as well. The German Federal Statistical Office, for instance, offers detailed data about the population of Germany and thus for many attributes of typical master data (e.g., of customers). Further examples are traffic data as well as healthcare databases providing detailed (anonymized) data about diseases and patients. From such a reference dataset D_R, it is possible to determine uncertain rules for the assessment of D_B directly and with a high degree of automation. In the following, we exemplarily discuss three possible ways for determining uncertain rules based on a reference dataset.

First, an association rule mining algorithm (Agrawal et al., 1993; Kotsiantis and Kanellopoulos, 2006) can be applied to D_R. The resulting association rules can subsequently be used as input for the metric. Applying an association rule mining algorithm in this context differs from existing works using association rules for the assessment of consistency (e.g., Alpar and Winkelsträter, 2014). In our context the rules and their confidence are not determined based on the

dataset to be assessed itself, but on a reference dataset, which prevents possibly misleading results in case part of the dataset to be assessed is erroneous. Moreover, using an association rule mining algorithm in our context means that uncertain rules with a rule confidence below a chosen threshold for minimum rule confidence are not excluded. Such rules with low confidence are beneficial for assessing consistency with the metric presented in this paper and, thus, should also be mined. This can be achieved using common association rule mining algorithms (e.g., the Apriori algorithm; Agrawal and Srikant, 1994).

Still, it is possible that for a specific data value, no association rule can be used to assess consistency because the data value is not part of an antecedent or consequent in any rule. Thus, we suggest further ways to determine or enhance a set of uncertain rules based on a reference dataset.

As a second way, we propose the use of so-called *column rules*, which can also be determined in an automated manner. Using column rules to assess the consistency of D_B means that dependencies between different attributes are not considered. These rules consist of a tautological antecedent $\top$ (i.e., the logical statement A is always true) and $a_l = \phi(t_m, a_l)$ as a consequent for all records t_m in D_R and attributes a_l of D_R. This results in the rule set of the form $R_c = \{r: \top \rightarrow a_l = \phi(t_m, a_l)\}$, where the probability of a rule represents the relative frequency of occurrence of $\phi(t_m, a_l)$ in D_R. For example, for a record t in D_R with $\phi(t, year\ of\ birth) = 1997$ and $\phi(t, marital\ status) = single$, $r_1: \top \rightarrow year\ of\ birth = 1997$ and $r_2: \top \rightarrow marital\ status = single$ would be added to R_c.

Third, so-called *row rules* can also be used. Row rules are very strict with regard to their fulfillment, as all of the data values of a record need to match. These rules with tautological antecedent $A = \top$ and $\bigwedge_{a_l}(a_l = \phi(t_m, a_l))$ as consequent for all t_m in D_R can be generated in an automated manner as well. This leads to the rule set of the form $R_r = \{r: \top \rightarrow \bigwedge_{a_l}(a_l = \phi(t_m, a_l))\}$, where the probability of a rule represents the relative frequency of occurrence of t_m in D_R. To give an example, for a record t in D_R with $\phi(t, year\ of\ birth) = 1997$ and $\phi(t, marital\ status) = single$ (and no other attributes in D_R), the rule $r_3: \top \rightarrow year\ of\ birth = 1997 \wedge marital\ status = single$ would be added to R_r.

These three ways for obtaining uncertain rules based on a reference dataset D_R were presented because of their general applicability. A large variety of further uncertain rules can be determined, for example by considering fixed attributes in the antecedent or by using different operators. Depending on D_B and the specific application, any of these possibilities (or a combination of them) can be favorable as the dependencies between attributes may vary. For instance, in a context where dependencies of attributes do not have to be analyzed at all, using column rules is promising. Another example is provided in Section 4, where uncertain rules based on a reference dataset from the German Federal Statistical Office are determined. In any of these ways, the relative frequency with which r is fulfilled in D_R can be calculated and used as $p(r)$.

Thereby, based on D_R both rules and corresponding probabilities of fulfillment can be determined with a high degree of automation. This allows a use of multiple rule sets to focus on different aspects of the data to be assessed or to analyze the specific reasons for inconsistencies in the data (cf. Section 4).

When using a reference dataset D_R for determining rules, the number $|fulfilling\ records(D_R, A \wedge C)|$ of records in D_R fulfilling a rule needs to be sufficiently large to ensure reliable metric values with respect to this rule. To be more precise, the statistical significance of $p(r)$ needs to be assured. If an association rule mining algorithm is used, a suitable minimum support can be fixed to exclude rules based on a non-significant proportion of records. In any case, a statistical test can be applied in order to determine the minimal number of records required such that a rule has a significant explanatory power (cf. Section 4). Moreover, to provide a statistically reliable basis and to circumvent the aforementioned issue, rules can be aggregated (e.g., by using a disjunction). In this way, robust estimations of $p(r)$ can be obtained, allowing the determination of reliable metric values.

Ad (ii): If neither internal nor external reference data is available, conducting a study is a further possibility. For example, if a customer database is to be assessed, a random sample of the customers can be drawn and surveyed. The survey results can be used to determine appropriate uncertain rules by analyzing the customers' statements. Moreover, the corresponding values of $p(r)$ for each rule r can be obtained by analyzing how many of the surveyed customers fulfill the rule. Thus, the input parameters for the metric are provided. As a result of the survey, one obtains quality assured data of the surveyed customers and can also assess the consistency of the data of customers not part of the survey.

Ad (iii): Another possibility is to use an expert-based approach (similar to Mezzanzanica et al., 2012; Baker and Olaleye, 2013; Meyer and Booker, 2001). Here, the idea is to survey qualified individuals. For rules in a customer database of an insurer taking into the account the attributes *number of insurance relationships*, *insurance group* and *fee paid*, insurance experts could be surveyed. Another example concerns very rare events such as insurance exclusions without reimbursement, for which not enough (reference) data is available. The experts can assess which rules are suitable to describe the expected structure of the considered data values and can specify the respective values of $p(r)$ for each rule.

3.4.2 Calculating the Metric Values

Based on a set of uncertain rules R with values $p(r)$ for each $r \in R$, the metric values can be calculated in an automated manner. The values $|fulfilling\ records(D_B, A)|$ and $|fulfilling\ records(D_B, A \wedge C)|$ can be determined efficiently via simple database queries. In addition, based on the value of $p(r)$, the corresponding binomial distribution can be instantiated. Then, the (two-sided) p-value with regard to $|fulfilling\ records(D_B, A \wedge C)|$ can be calculated in order to obtain the metric values.

In the literature, several different approaches to calculate the two-sided p-value have been proposed (Dunne et al., 1996). These include doubling the one-sided p-value and clipping to one, summing up the probabilities less than or equal to the probability of the observed result, and more elaborate ways. In practical applications, for non-symmetric distributions, the approaches to calculate the two-sided p-value may lead to slightly different results. However, the larger the sample size (in our case $|fulfilling\ records(D_B, A)|$), the smaller the differences between the results of the different approaches are. This is due to the fact that for $p(r) \in (0;\ 1)$, the binomial distribution converges to the (symmetric) normal distribution (de Moivre-Laplace theorem).

4 Evaluation

In this section we evaluate (E1) the practical applicability as well as (E2) the effectiveness (Prat et al., 2015) of our metric for consistency in a real-world setting. First, we discuss the reasons for selecting the case of a German insurer and describe the assessed customer dataset. Then, we show how the metric could be instantiated for this case. Subsequently, we present and discuss the results of the application. Finally, we compare the results with those of existing metrics for consistency.

4.1 Case Selection and Dataset

The relevance of managing customer data at a high data quality level is well acknowledged (cf. e.g., Even et al., 2010; Heinrich and Klier, 2015b). The metric was applied in cooperation with one of the major providers of life insurances in Germany. High data quality of customer master data is critical for the insurer and plays a particularly important role in the context of customer management. However, the staff of the insurer suspected data quality issues due to negative customer feedback (e.g., in the context of product campaigns). Customers claimed to have a marital status different from the focused target group of campaigns. Thus, they either were not interested in the product offerings or were not even eligible to participate. To analyze these issues, we aimed to assess the consistency of the customers' marital status depending on their age.

This setting seemed particularly suitable for showing the applicability and effectiveness of our metric for the following reasons: First, the marital status of a customer is a crucial attribute for the insurer, because insurance tariffs and payouts often vary depending on marital status. Indeed, for example a customer whose marital status is erroneously stored as *widowed* may receive unwarranted life insurance payouts. Additionally, the marital status also significantly influences product offerings, as customers with different marital statuses tend to have varying insurance needs. In fact, as mentioned above, customers may even only be eligible for a particular insurance if they have a specific marital status. Second, interpretable metric values are of particular importance in this setting, for instance to facilitate the aforementioned product offerings. Third, using traditional rules which are "true by definition" is not promising here as except

for children, who are always *single*, no marital status is definite or impossible for customers. For example, a 60-year-old customer may be *single, married, divorced, widowed,* etc., each with specific probability.

To conduct the analyses described above, the insurer provided us with a subset of its customer database. The analyzed dataset contains five attributes storing data about customers of the insurer born from 1922 onwards and represents the state of the customer data from 2016. The subset consists of 2,427 records which had a value for both the attribute *marital status* and the attribute *date of birth*. Each record represents a specific customer of the insurer. The *marital status* of the customers was stored as a numerical value representing the different statuses *single, married, divorced, widowed, cohabiting, separated* and *civil partnership*. As the marital statuses *cohabiting* and *separated* are not recognized by German law (Koordinierungsstelle für IT-Standards, 2014), we matched these statuses to the respective official statuses *single* and *married*. The *date of birth* was stored in a standard date format. On this basis, customers' age could easily be calculated and stored as an additional attribute *age*. Moreover, an attribute *gender* was available both in the customer dataset as well as in the data used for the instantiation of the metric (cf. following subsection). As gender may have a significant impact on marital status as well, we also included this attribute in our analysis. Each of the 2,427 records contained a value for *gender*, classifying the respective customer as either *male* or *female*.

4.2 Instantiation of the Metric for Consistency

In Section 3.4.1, we described possibilities to obtain a set of uncertain rules for the instantiation of our metric. In our setting, we were able to use publicly available data from the German Federal Statistical Office as a reference dataset and thus chose option (i). The German Federal Statistical Office provides aggregated data regarding the number of inhabitants of Germany having a specific marital status. We used the most recent data available, which is based on census data from 2011 and was published in 2014 (German Federal Statistical Office, 2014). The data is broken down by age (in years) as well as gender and includes all Germans regardless of their date of birth, containing in particular the data of the insurer's customers. Overall, the data from the German Federal Statistical Office seems to be an appropriate reference dataset for our setting and could be used to determine meaningful uncertain rules and the probabilities $p(r)$ for each rule r.

As it was our aim to examine consistency of the marital status of customers depending on their age and gender, both attributes *age* and *gender* were part of the antecedent of the rules while the attribute *marital status* was contained in the consequent. To determine a rule set, we proceeded as follows: First, for each marital status *m*, each gender *g* and each possible value of age $a \in \mathbb{N}$, we specified rules of the following form:

$$r_{m,g}^{a}\colon (age = a) \wedge (gender = g) \rightarrow marital\ status = m \tag{7}$$

Second, we calculated the probabilities $p(r_{m,g}^{a})$ based on the data from the German Federal

Statistical Office. Third, starting at an age of 0 years, we systematically aggregated these rules to rules of the form:

$$(age \geq a_1) \wedge (age < a_2) \wedge (gender = g) \rightarrow marital\ status = m \quad (8)$$

Here, $a_1, a_2 \in \mathbb{N}$ (with $a_1 < a_2$) specify an age group. The aggregation of the rules $r_{m,g}^{a}$ was performed to increase the number of records each rule was relevant for. However, age groups also have to be homogeneous and thus, the differences in probabilities of rule fulfillment within an age group were required to not exceed a specific threshold. More precisely, for a given value of a_1, the value a_2 was determined to be the maximum of all values $j \in \mathbb{N}$ for which $\left|p\left(r_{m,g}^{j}\right) - p\left(r_{m,g}^{k}\right)\right| \leq 0.1$ held for all $a_1 \leq k \leq j$. In this way the following rule $\tilde{r}$ for single men between 42 and 49 was obtained:

$$\tilde{r}: (age \geq 42) \wedge (age < 50) \wedge (gender = male) \rightarrow marital\ status = single \quad (9)$$

Afterwards, for each rule $r \in R$ the probabilities $p(r)$ were calculated based on the data from the German Federal Statistical Office. For example, as approximately 26.4% of men between 42 and 49 are single according to the German Federal Statistical Office, this resulted in $p(\tilde{r})$=0.264. Moreover, a statistical test to the significance level of 0.05 was applied to ensure that each rule is based on a statistically significant number of relevant records in both the reference dataset and the customer dataset. Rules not fulfilling the test were excluded from further analysis to guarantee reliable metric results. This way, 37 different rules and corresponding probabilities were determined.

Each customer record of the insurer belonged to one of the age groups and had the value *male* or *female* for the attribute *gender* and the value *single*, *married*, *divorced*, *widowed* or *civil partnership* for the attribute *marital status* as represented by our rule set. Accordingly, a metric value could be determined for the value of the attribute *marital status* of each of these records. For instance, to assess the consistency of the marital status *single* of a 46-year-old male customer t, $\tilde{r}$ was used. The calculation of the metric value by means of Definition (6) yielded a consistency of 0.888:

$$Q_{Cons}(\phi(t, single), \tilde{r}) = p\text{-}value(X(\tilde{r}) \sim B(57, 0.264), 14) = 0.888 \quad (10)$$

To calculate the two-sided p-value, we doubled the one-sided and clipped to one (Dunne et al., 1996).

4.3 Application of the Metric for Consistency and Results

Having instantiated the metric, we applied the metric to the 2,427 customer records by means of a Java implementation. The results for the marital status *widowed* seemed particularly interesting and alarming. Indeed, in contrast to the other marital statuses, analyses for this marital status revealed that the metric values were very low across all customer records. In fact, for the 1,160 records with a marital status of *widowed*, the metric value was always below 0.001 (cf. Table 2).

Gender	Age Group	Relative Frequency of Rule Fulfillment (Insurer Dataset)	Probability of Corresponding Rule (Statistical Office)	Value of the Metric for Consistency
male	0-74	0.139	0.012	0.000
	75-81	0.713	0.132	0.000
	>=82	0.744	0.313	0.000
female	0-60	0.096	0.016	0.000
	61-68	0.435	0.143	0.000
	69-73	0.676	0.248	0.000
	74-77	0.898	0.359	0.000
	78-80	0.950	0.483	0.000
	81-84	0.921	0.610	0.000
	>=85	0.918	0.754	0.000

Table 2. Results of the Metric for Consistency per Age Group and Gender

This means that for each record, the difference between actual rule fulfillment and expected rule fulfillment was so large that it is very unlikely to have occurred by chance. To be more precise, this probability was less than 0.001 for each record. Thus, with the results being based on a large number of records, consistency of *widowed* was assessed as very low with high statistical significance. This led to the conclusion that a previously undetected systematic bias had to be present in the customer data. In general, various different reasons could have led to this bias (e.g., a systematic data error such as a large number of young customers erroneously captured and stored as *widowed*). The bias was likely to cause serious problems for the insurer (e.g., due to negative effects on insurance tariffs and product offerings). We thus decided to investigate this issue further by analyzing each age group and gender based on the respective metric values, focusing on all rules with *marital status = widowed* in the consequent.

Table 2 illustrates the results of this analysis for all age groups. The first two columns display which customers were taken into account (rule antecedent). The third column shows the relative frequency of fulfillment of the respective rule (i.e., the proportion of customers in this age group and of this gender which had the marital status *widowed*). The penultimate column specifies the probability of the respective rule based on the data of the German Federal Statistical Office which was determined during the instantiation of the metric. Finally, the last column shows the corresponding metric value for consistency. Obviously, for a marital status of *widowed*, the bias in the data was so strong that the metric value was below 0.001 in each case. For example, the dataset included 107 female customers of age between 61 and 68 with marital status *widowed*, which results in a relative frequency of rule fulfillment of 0.435. The corresponding rule was:

$$(age \geq 61) \wedge (age < 69) \wedge (gender = female) \rightarrow marital\ status = widowed \qquad (11)$$

The probability of this rule, however, was determined to be just 0.143 based on the data of the German Federal Statistical Office (i.e., 14.3% of female customers within that age group were

expected to be *widowed*). Measuring consistency as the probability that the assessed data is free of internal contradictions with regard to this rule results in a metric value of 0.000 (rounded). This means that the actual rule fulfillment was so different from the expected rule fulfillment that it is very likely that a systematic bias was present in the customer data.

The results in Table 2 indicate that the relative frequency of rule fulfillment was considerably higher than the probability of the corresponding rule in each row. This means that a much larger number of customers than to be expected was considered as widowed by the insurer. A systematic bias of this magnitude in the insurer's customer data could result in severe economic losses for the insurer. Thus, we aimed to find the reason(s) for this potential data quality issue.

We suspected that a data capturing problem or a data integration problem might have occurred during some time in the past, resulting in many customers being erroneously stored as *widowed*. To analyze this presumption, we took the additional attribute *month of acquisition* of the dataset into account. It represents the month in which a person first became customer of the insurer by a standard date format. Of the 2,427 records, 931 records had a *month of acquisition* in the recent years 2013-2016, while 786 records exhibited a *month of acquisition* further in the past (until November 1951) and 710 records had a missing value for this attribute. We chose 2013 as threshold because the insurer data was structured differently from this year on. We created a new rule set including *month of acquisition* in the antecedent. This rule set was determined analogously to the procedure above (with slightly different age groups due to *month of acquisition*). For example, the rule for a widowed female customer in age group 55-68 acquired by the insurer in 2013-2016 was then given by:

$$(age \geq 55) \wedge (age < 69) \wedge (gender = female) \\ \wedge (month\ of\ acquisition \in [2013, 2016]) \rightarrow marital\ status = widowed \quad (12)$$

The probabilities of the rules were again determined based on the German Federal Statistical Office data regarding the respective age, gender and marital status (e.g., 0.108 for the rule in (12)). The results from applying our metric using this new rule set are illustrated in Table 3. Here, we focus on the age group per gender with the highest number of widowed customers. The first two columns again specify which customers were taken into account. The probability of the rules for this age group and gender based on data from the German Federal Statistical Office is given in the third column. The fourth to sixth columns show the relative frequencies of rule fulfillment and the corresponding metric values for a missing *month of acquisition*, a *month of acquisition* before 2013 and a *month of acquisition* in 2013-2016.

Age Group	**Gender**	**Probability of Corresponding Rules**	**Relative Frequency of Rule Fulfillment (Insurer Dataset)/ Value of the Metric if Value of *month of acquisition is...***		
			...missing	**...before 2013**	**...in 2013-2016**
63-80	male	0.078	0.760 / 0.000	0.494 / 0.000	0.067 / 0.792
55-68	female	0.108	0.794 / 0.000	0.458 / 0.000	0.105 / 0.978

Table 3. Results of the Metric for Consistency considering the Month of Acquisition

This more detailed analysis shows that metric values are equal to 0.000 in the case of a missing *month of acquisition* or a *month of acquisition* before 2013, caused by very large relative frequencies of the data value *widowed* compared to the low probabilities of the corresponding rules. In contrast, for a *month of acquisition* in 2013-2016, relative frequencies and probabilities are much closer (0.067 and 0.078 resp. 0.105 and 0.108), resulting in higher metric values (0.792 resp. 0.978). We concluded that mainly records with a missing *month of acquisition* or a *month of acquisition* before 2013 were problematic and caused the consistency problems.

We discussed our findings with a board member of the insurer. He confirmed that an organizational restructuring had taken place in 2013. It included a revamp of the data capturing process, giving a reason why the customer data from 2013 onwards showed significantly higher consistency. However, it was not known that a data quality problem concerning the marital status *widowed* had existed beforehand. This problem was neither recognized nor solved during the restructuring process and thus still persisted in the customer data. Subsequently conducted internal evaluations of the insurer revealed that the *marital status* of customers had not been captured rigorously in the past and thus its values for customers with a value of *month of acquisition* before 2013 were not trustworthy. This clarified the too large relative frequency of *widowed* in the case of a *month of acquisition* before 2013 (and a missing *month of acquisition*, indicating an even more erroneous record). Further, the values of our metric for consistency allowed to quantify the too large relative frequency of *widowed* and to decide whether the deviation was significant. Thus, due to the clear interpretation, the metric values could then be used to decide which data values of *marital status* to consider as trustworthy in the future. Later on, the board member stated that initiatives to check the marital status of customers acquired before 2013 were started in order to rectify erroneous records (and, e.g., prevent unwarranted life insurance payouts). To do so, employees of the insurer began to analyze old paper-based documents containing customer data. Moreover, the insurer aimed to improve its data quality by contacting customers whose marital status was (highly) probably erroneous as identified by our metric. These initiatives facilitate an improved customer management, for instance regarding the design of future customer campaigns. In particular, a high data quality of the marital status of customers supports to conduct successful campaigns focusing on a specific target group of customers.

In addition, we analyzed the efforts for the instantiation and application of the proposed metric (in the sense of required time) as well as the corresponding benefits in this real-world setting.

With respect to efforts, time was required to (i) find and prepare the census data of the German Federal Statistical Office, (ii) calculate the probabilities $p(r)$ based on the census data and conclude the rule set, (iii) assess the consistency by means of the metric and (iv) interpret and discuss the results. To conduct these four steps in our application setting, the following amount of time was necessary: With respect to (i), the data from the German Federal Statistical Office could be easily found online via a quick research. Due to their clear structure, preprocessing this data was not difficult after an initial familiarization. All in all, step (i) could be completed in one person-day. In another person-day, the rule set including the probabilities $p(r)$ was obtained and discussed. Indeed, the rule set and the probability for each rule could be determined in an automated manner. Based on this rule set, the assessment of the consistency of the dataset (iii) could be performed in less than one second using a Java implementation, which was realized in three further person-days. Of course, this effort is necessary only once and the implementation can be reused in further assessments, even in different application contexts. Finally, the results were interpreted and discussed (iv) both internally and in cooperation with the insurer in the course of two more person-days. Thus, the four steps (i) to (iv) to instantiate and apply the metric including the discussion of the findings resulted in overall efforts of about seven person-days.

These steps could be seen as part of a typical data quality assessment and improvement process (Wang, 1998). Here, in a preceding step, the data quality problems at hand have to be recognized and analyzed before the metric for semantic consistency is applied. For the insurer, this resulted in focusing on the consistency of the customers' marital status depending on their age. Similarly, in a succeeding step, initiatives to fix identified inconsistencies can be performed. Both complementing steps are related to the particular application context and, for instance, depend on the extent of identified semantic inconsistencies. In the case of the insurer, initiatives were conducted to improve the quality of the customer data to support future campaigns. In this regard, it is important to note that the efforts of the steps (i) to (iv) are reduced if an instantiated metric is reused in future consistency assessments. For example, data of new customers can be assessed using the same rules, probabilities (i.e., $p(r)$) and (tool) implementation. Only after some time (e.g., several years), an update of the underlying census data may become necessary to reflect demographical changes and to thus ensure valid results. However, even in this case, the four steps (i) to (iv) remain the same and the existing implementation can be used, resulting in smaller efforts compared to an initial conduction.

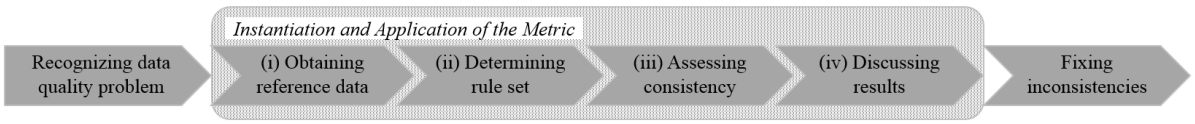

Figure 3. Core Steps to instantiate and apply the proposed Metric

Compared to the efforts for performing the steps (i) to (iv), which can be determined in a straightforward manner, the benefits of both (re)using the metric results (i.e., the resulting probabilities) and (re)using the improved data values are not easy to assess. From a methodological

and decision-oriented perspective, both benefits can be estimated by comparing the effects resulting from decisions with respectively without considering the metric results and the improved data quality (for a detailed discussion cf. Heinrich et al., 2018; Heinrich and Hristova, 2016). Not having or considering the metric results means that customers who actually have a marital status different from the focused target group are selected for the campaigns. Thus, products are offered to those customers wrongly. This may result in claims, which can be counted, assessed and attributed to a campaign as they arrive, allowing the quantification of their amount and severity. Preventing these claims by taking into account the metric results manifests a first benefit. However, such claims put forward to the insurer will just occur in a small number of cases and constitute only the "tip of the iceberg", as many customers would be annoyed by the campaign conducted based on low data quality, but not complain at all. The prevention of this decreasing customer satisfaction as a second (soft) benefit is difficult to measure. Moreover, using data values with improved data quality based on applying the consistency metric can lead to further improved decisions. More precisely, customers with corrected marital status can then be addressed in campaigns for which they would otherwise have been disregarded. Product sales being caused by these additionally considered customers constitute a third benefit resulting from fixing identified inconsistencies (thus representing a succeeding effect of applying the metric). In addition, both metric results and improved data quality cannot only be used in a single campaign, but also in future campaigns and customer interactions resulting in further benefits dependent on the particular application context (for a general decision-oriented framework comprising efforts and benefits of data quality assessment, we refer to Heinrich et al. (2018)). Overall, in the case of the insurer, the efficiency can be supported; however, without any doubt efficiency has to be examined individually for each application context.

4.4 Comparison of the Results with existing Metrics for Consistency

In order to further evaluate our approach, we also instantiated and applied existing metrics for consistency (Alpar and Winkelsträter, 2014; Cordts, 2008; Heinrich et al., 2007; Heinrich and Klier, 2015a; Hinrichs, 2002; Hipp et al., 2001; Hipp et al., 2007; Kübart et al., 2005; Pipino et al., 2002) for the case of the German insurer and compared the results. Thereby, we used the same dataset and again focused on the attributes *age*, *gender* and *marital status*. To instantiate the existing metrics, we determined association rules with *marital status* in the consequent. The values for minimum support and minimum confidence were chosen in accordance with the respective works. In particular, each existing metric was instantiated using three different settings for minimum support and minimum confidence, leading to rule sets of different sizes: In Setting 1 (minimum support: 0.01, minimum confidence: 0.80), 26 association rules were determined. Setting 2 (minimum support: 0.00025, minimum confidence: 0.85) led to 111 rules and Setting 3 (minimum support: 0.0001, minimum confidence: 0.75) to 153 rules. Further, not all existing metrics provide values within the interval $[0;\ 1]$. Thus, to be able to compare the results, we transformed all metric values to this interval. This was done so that for each approach, the value

0 resp. 1 represent the minimal resp. maximal determined consistency.

For each approach and setting, we analyzed the minimum, average and maximum metric values over all records with marital status *widowed*. Regarding the existing approaches, the consistency of the attribute value *widowed* of the attribute *marital status* in the dataset is actually assessed to be rather high or even very high. Indeed, all approaches except the ones by Alpar and Winkelsträter (2014) and Hipp et al. (2007) assess the dataset as perfectly consistent or almost perfectly consistent (average metric value of at least 0.991). Even the metric values determined by the approaches of Alpar and Winkelsträter (2014) and Hipp et al. (2007) do not indicate a (critical) consistency problem as the average metric values are still at least 0.689 and thus rather high. Hence, existing approaches do not identify the severe consistency problem existing in the data and acknowledged by the insurer. In contrast, this problem is clearly indicated by the very low metric values (0.000 each as minimum, average and maximum metric value) determined by our approach using uncertain rules. The evaluation results are presented in Table 4 (higher metric values are represented by cells with darker background).

	Setting	**Minimum...**	**Average...**	**Maximum...**
		...metric value of records with marital status *widowed*		
Our proposed Metric	N/A	0.000	0.000	0.000
Alpar and Winkelsträter (2014); Hipp et al. (2007)	1	0.492	0.716	1.000
	2	0.483	0.689	1.000
	3	0.269	0.796	1.000
Hipp et al. (2001); Kübart et al. (2005)	1	1.000	1.000	1.000
	2	1.000	1.000	1.000
	3	0.556	0.996	1.000
Hinrichs (2002)	1	1.000	1.000	1.000
	2	1.000	1.000	1.000
	3	0.304	0.993	1.000
Cordts (2008); Pipino et al. (2002)	1	1.000	1.000	1.000
	2	1.000	1.000	1.000
	3	0.000	0.991	1.000
Heinrich et al. (2007); Heinrich and Klier (2015a)	1	1.000	1.000	1.000
	2	1.000	1.000	1.000
	3	0.000	0.991	1.000

Table 4. Comparison of the Results with existing Metrics for Consistency

To sum up, regarding (E1), the evaluation in a real-world setting demonstrated the practical applicability of our metric for consistency. Publicly available data could be used to determine a rule set with probabilities for each rule and instantiate the metric. Thereafter, the metric could be applied to identify consistency problems in the considered dataset. With respect to (E2), the evaluation also substantiated the effectiveness of our metric. Applying the metric multiple times (for increasingly detailed analyses) led to the identification of specific consistency problems in

a real-world customer dataset, which, in comparison, was not supported when using existing metrics for consistency.

5 Conclusion, Limitations and Future Work

In this paper, we present a probability-based metric for the data quality dimension semantic consistency using uncertain rules. Existing approaches for measuring semantic consistency only consider rules that are "true by definition", which means, the fulfillment of such a rule is always used as an indicator for high consistency. This impedes the consideration of rules that are expected to be *not* fulfilled for a higher number of data values. For example, a rule which is expected to be fulfilled only rarely, but is actually fulfilled very often in the assessed dataset, is an important indicator for inconsistent data. In addition, "true by definition" rules based on the assessed data can lead to misleading results if, for instance, a large part of the data is erroneous due to a systematic data error. Then the smaller part of accurately stored data values would be considered as inconsistent. Consequently, many consistency problems cannot be detected and assessed. We thus consider uncertain rules in the assessment of consistency by taking into account the probability with which a rule is expected to be fulfilled. This allows to determine a metric value which represents the probability that the dataset to be assessed is free of internal contradictions with regard to uncertain rules. The theoretical foundation for determining the metric values are statistical tests and the concept of the p-value. In particular, the fulfillment of a rule is modeled as a Bernoulli-distributed random variable. On this foundation, our metric is defined as the two-sided p-value of a binomial distribution. Thus, the metric values can be interpreted as the probability that the data values to be assessed do not contradict the considered rule set. This clear interpretation is relevant to support decision-making based on the metric values. We provide a formal metric definition and present different possibilities for the instantiation of the metric, in particular for determining a rule set. Further, we evaluate the practical applicability and effectiveness of our metric in a real-world setting by analyzing a customer dataset of an insurance company. Here, our metric could be applied to identify consistency problems in the data, which was not supported when using existing metrics for consistency.

There are also some limitations that may constitute the starting point for future research. To begin with, we evaluated our metric by analyzing a single customer dataset. Future research could, first of all, cover the application of the metric to additional datasets containing master data. Further, an application of the metric to different contexts such as, for example, sensor data is promising as well and has already yielded interesting results in an initial analysis we conducted. Moreover, for our application to the customer dataset, we determined a rule set based on reference data from the German Federal Statistical Office. Other ways to instantiate the metric are also feasible, but may require additional considerations (e.g., how to conduct a cost-efficient survey to determine the rule set). Future research should thus evaluate the application of other types of rules such as association rules, rules obtained by a survey and rules derived by experts. Moreover, the dataset we assessed contained about 2,400 records and is thus not very

large. It would be interesting to apply the metric to a larger dataset and compare the results. Another possible path for future research is to develop elaborate aggregation procedures which take the statistical properties of the metric into account. For instance, an aggregation could be defined based on the sum of random variables following a Bernoulli distribution and thus also be interpreted as p-value. Finally, our metric is defined for structured data. However, in general, it can be extended to semi- and unstructured data by applying text mining methods such as inverted term frequency.

6 References

Abboura, A., S. Sahri, L. Baba-Hamed, M. Ouziri and S. Benbernou (2016). "Quality-based online data reconciliation" *ACM Transactions on Internet Technology (TOIT)* 16 (1).

Agrawal, R., T. Imieliński and A. Swami (1993). "Mining association rules between sets of items in large databases". In: *ACM SIGMOD Record*, pp. 207–216.

Agrawal, R. and R. Srikant (1994). "Fast algorithms for mining association rules". In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994)*, pp. 487–499.

Alkharboush, N. and Yuefeng Li (2010). "A decision rule method for assessing the completeness and consistency of a data warehouse". In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2010)*.

Alpar, P. and S. Winkelsträter (2014). "Assessment of data quality in accounting data with association rules" *Expert Systems with Applications* 41 (5), 2259–2268.

Baker, E. and O. Olaleye (2013). "Combining experts: Decomposition and aggregation order" *Risk Analysis* 33 (6), 1116–1127.

Batini, C. and M. Scannapieco (2006). *Data-centric systems and applications. Concepts, methodologies and techniques:* Springer.

Batini, C. and M. Scannapieco (2016). *Data and information quality:* Springer.

Blake, R. H. and P. Mangiameli (2009). "Evaluating the semantic and representational consistency of interconnected structured and unstructured data". In *Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009)*.

Blake, R. H. and P. Mangiameli (2011). "The effects and interactions of data quality and problem complexity on classification" *Journal of Data and Information Quality (JDIQ)* 2 (2), 1–28.

Bohannon, P., W. Fan, F. Geerts, X. Jia and A. Kementsietsidis (2007). "Conditional functional dependencies for data cleaning". In: *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE 2013)*, pp. 746–755.

Bronselaer, A., J. Nielandt, R. de Mol and G. de Tré (2016). "Ordinal assessment of data consistency based on regular expressions". In: *Proceedings of the 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2016)*, pp. 317–328.

Chiang, F. and R. J. Miller (2008). "Discovering data quality rules" *Proceedings of the VLDB Endowment* 1 (1), 1166–1177.

Cong, G., W. Fan, F. Geerts, X. Jia and S. Ma (2007). "Improving data quality. Consistency and accuracy". In: *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007)*, pp. 315–326.

Cordts, S. (2008). "Implementierung eines Datenqualitätsdienstes zur evolutionären Datenqualitätsverbesserung in relationalen Datenbankmanagementsystemen". Dissertation. University of Hamburg.

Dunne, A., Y. Pawitan and L. Doody (1996). "Two-sided p-values from discrete asymmetric distributions based on uniformly most powerful unbiased tests" *The Statistician* 45, 397–405.

Even, A., G. Shankaranarayanan and P. D. Berger (2010). "Evaluating a model for cost-effective data quality management in a real-world CRM setting" *Decision Support Systems (DSS)* 50, 152–163.

Fan, W., F. Geerts, N. Tang and W. Yu (2013). "Inferring data currency and consistency for conflict resolution". In: *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE 2013)*, pp. 470–481.

Fisher, C. W., E. J. M. Lauria and C. C. Matheus (2009). "An accuracy metric: Percentages, randomness, and probabilities" *Journal of Data and Information Quality (JDIQ)* 1 (3), 1–21.

German Federal Statistical Office (2014). *Current population*. URL: https://www.destatis.de/EN/FactsFigures/SocietyState/Population/CurrentPopulation/CurrentPopulation.html (visited on 02/09/2017).

Golab, L., H. Karloff, F. Korn, D. Srivastava and B. Yu (2008). "On generating near-optimal tableaux for conditional functional dependencies" *Proceedings of the VLDB Endowment* 1 (1), 376–390.

Goodman, S. (2008). "A dirty dozen: twelve p-value misconceptions" *Seminars in Hematology* (45), 135–140.

Heinrich, B. and D. Hristova (2016). "A quantitative approach for modelling the influence of currency of information on decision-making under uncertainty" *Journal of Decision Systems (JDS)* 25 (1), 16–41.

Heinrich, B., D. Hristova, M. Klier, A. Schiller and M. Szubartowicz (2018). "Requirements for Data Quality Metrics" *Journal of Data and Information Quality (JDIQ)* 9 (2), 12.

Heinrich, B., M. Kaiser and M. Klier (2007). "Metrics for measuring data quality - foundations for an economic oriented management of data quality". In *Proceedings of the 2nd International Conference on Software and Data Technologies (ICSOFT 2007)*.

Heinrich, B. and M. Klier (2011). "Assessing data currency—a probabilistic approach" *Journal of Information Science* 37 (1), 86–100.

Heinrich, B. and M. Klier (2015a). "Datenqualitätsmetriken für ein ökonomisch orientiertes Qualitätsmanagement". In *Daten-und Informationsqualität: Auf dem Weg zur Information Excellence. 2. Auflage*, pp. 49–67: Vieweg + Teubner.

Heinrich, B. and M. Klier (2015b). "Metric-based data quality assessment—Developing and evaluating a probability-based currency metric" *Decision Support Systems (DSS)* 72, 82–96.

Hinrichs, H. (2002). "Datenqualitätsmetriken in Data Warehouse-Systemen". Dissertation. University of Oldenburg.

Hipp, J., U. Güntzer and U. Grimmer (2001). "Data quality mining - making a virtue of necessity". In: *Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2001)*.

Hipp, J., M. Müller, J. Hohendorff and F. Naumann (2007). "Rule-Based Measurement Of Data Quality In Nominal Data". In: *Proceedings of the 12th International Conference on Information Quality (ICIQ 2007)*, pp. 364–378.

Kim, W., B.-J. Choi, E.-K. Hong, S.-K. Kim and D. Lee (2003). "A taxonomy of dirty data" *Data Mining and Knowledge Discovery* 7 (1), 81–99.

Koordinierungsstelle für IT-Standards (2014). *Datensatz für das Meldewesen. Einheitlicher Bundes-/Länderteil (DSMeld)*.

Kotsiantis, S. and D. Kanellopoulos (2006). "Association rules mining: A recent overview" *GESTS International Transactions on Computer Science and Engineering* 32 (1), 71–82.

Kübart, J., U. Grimmer and J. Hipp (2005). "Regelbasierte Ausreißersuche zur Datenqualitätsanalyse" *Datenbank-Spektrum* 14, 22–28.

Laranjeiro, N., S. N. Soydemir and J. Bernardino (2015). "A survey on data quality. Classifying poor data". In: *21st IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2015)*, pp. 179–188.

Lee, Y. W., L. Pipino, D. M. Strong and R. Y. Wang (2004). "Process-embedded data integrity" *Journal of Database Management (JDM)* 15 (1), 87–103.

Lee, Y. W., L. L. Pipino, J. D. Funk and R. Y. Wang (2006). *Journey to Data Quality:* The MIT Press.

Liu, L. and L. N. Chi (2002). "Evolutional data quality: a theory-specific view". In *Proceedings of the 7th International Conference on Information Quality (ICIQ 2002)*, pp. 292–304.

Mecella, M., M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci and C. Batini (2002). "Managing Data Quality in Cooperative Information Systems". In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pp. 486–502: Springer.

Meyer, M. A. and J. M. Booker (2001). *Eliciting and analyzing expert judgment: a practical guide:* SIAM.

Mezzanzanica, M., M. Cesarini, F. Mercorio and R. Boselli (2012). "Towards the Use of Model Checking for Performing Data Consistency Evaluation and Cleansing". In: *Proceedings of the 17th International Conference on Information Quality (ICIQ 2012)*, pp. 163–177.

Moges, H.-T., K. Dejaeger, W. Lemahieu and B. Baesens (2011). "Data quality for credit risk management: new insights and challenges". In *Proceedings of the 16th International Conference on Information Quality (ICIQ 2011)*, pp. 632–646.

Moges, H.-T., V. van Vlasselaer, W. Lemahieu and B. Baesens (2016). "Determining the use of data quality metadata (DQM) for decision making purposes and its impact on decision outcomes—An exploratory study" *Decision Support Systems (DSS)* 83, 32–46.

Moore, S. (2018). *How to Create a Business Case for Data Quality Improvement*. URL: http://www.gartner.com/smarterwithgartner/how-to-create-a-business-case-for-data-quality-improvement/ (visited on 03/25/2019).

Ngai, E. W. T., A. Gunasekaran, S. F. Wamba, S. Akter and R. Dubey (2017). "Big data analytics in electronic markets" *Electronic Markets (EM)* 27 (3), 243–245.

Oliveira, P., F. Rodrigues and P. Henriques (2005). "A Formal Definition of Data Quality Problems". In: *Proceedings of the 10th International Conference on Information Quality (ICIQ 2005)*.

Orr, K. (1998). "Data quality and systems theory" *Communications of the ACM* 41, 66–71.

Pipino, L. L., Y. W. Lee and R. Y. Wang (2002). "Data quality assessment" *Communications of the ACM* 45 (4), 211–218.

Prat, N., I. Comyn-Wattiau and J. Akoka (2015). "A taxonomy of evaluation methods for information systems artifacts" *Journal of Management Information Systems (JMIS)* 32 (3), 229–267.

Rahm, E. and H. H. Do (2000). "Data cleaning. Problems and current approaches" *IEEE Bulletin on Data Engineering* 23 (4), 3–13.

Redman, T. C. (1996). *Data Quality for the Information Age:* Artech House.

Scannapieco, M., P. Missier and C. Batini (2005). "Data Quality at a Glance" *Datenbank-Spektrum* 14, 6–14.

Shankaranarayanan, G., B. Iyer and D. Stoddard (2012). "Quality of Social Media Data and Implications of Social Media for Data Quality". In: *Proceedings of the 17th International Conference on Information Quality (ICIQ 2012)*, pp. 311–325.

Singh, R. and K. Singh (2010). "A descriptive classification of causes of data quality problems in data warehousing" *International Journal of Computer Science Issues (IJCSI)* 7 (3), 41–50.

Srikant, R. and R. Agrawal (1996). "Mining quantitative association rules in large relational tables". In: *ACM SIGMOD Record*, pp. 1–12.

Wand, Y. and R. Y. Wang (1996). "Anchoring data quality dimensions in ontological foundations" *Communications of the ACM* 39 (11), 86–95.

Wang, H., J. Li and H. Gao (2016). "Data Inconsistency Evaluation for Cyberphysical System" *International Journal of Distributed Sensor Networks (IJDSN)* 12 (8).

Wang, R. Y. (1998). "A product perspective on total data quality management" *Communications of the ACM* 41 (2), 58–65.

Wang, R. Y. and D. M. Strong (1996). "Beyond accuracy: what data quality means to data consumers" *Journal of Management Information Systems (JMIS)* 12 (4), 5–33.

Witchalls, C. (2014). *Gut & gigabytes: Capitalising on the art & science in decision making.* PwC.

Zak, Y. and A. Even (2017). "Development and evaluation of a continuous-time Markov chain model for detecting and handling data currency declines" *Decision Support Systems (DSS)* 103, 82–93.

2.2 Paper 2: Event-driven Duplicate Detection – A Probability-based Approach

Current Status	Full Citation
accepted and published in the 2018 Proceedings of the European Conference on Information Systems (06/2018); winner of the Claudio Ciborra Award for the most innovative paper of ECIS 2018	Heinrich, B., M. Klier, A. Obermeier and A. Schiller (2018). "Event-Driven Duplicate Detection: A probability-based Approach". In: *Proceedings of the 26th European Conference on Information Systems (ECIS)*, June 23-28, Portsmouth, UK.

Summary

This paper deals with RQ2 by presenting a novel probability-based approach for duplicate detection. It seeks to determine the probability for a pair of records to be a duplicate caused by a real-world event (e.g., relocating customers). By first grounding the approach on such real-world events and subsequently formalizing corresponding mathematical expressions, the underlying causes for duplicates are considered in the assessment. A formal definition of the approach and multiple possibilities for its instantiation are provided. Similar to Paper 1, the practical applicability and effectiveness of the approach are evaluated in a real-world setting by analyzing customer data of an insurance company. Here, the approach is used to analyze potential duplicates caused by relocations. Moreover, the presented approach is shown to outperform the well-known state-of-the-art approach Febrl with respect to classifying pairs of records into duplicates and non-duplicates in this setting.

To quantify uncertainty and engage decision-making under uncertainty, the work relies on concepts and methods from probability theory. In particular, real-world events are explicitly modeled as outcomes in a probability space. Based on this, conditional duplicate probabilities are determined for each real-world event, the conditions being expressed by a feature vector. Both probability and causes for duplicates are beneficial to know for decision support, which is also confirmed in the evaluation. For instance, revealing the causes for duplicates can be a starting point for data quality improvement measures. Moreover, due to the interpretation of the results as probabilities, the integration into a decision calculus such as an expected value calculus is possible in a well-founded manner.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style. Moreover, terms only common in British English have been converted to corresponding American English terms.

The paper as published by AIS is available at: https://aisel.aisnet.org/ecis2018_rp/198

Abstract:

The importance of probability-based approaches for duplicate detection has been recognized in both research and practice. However, existing approaches do not aim to consider the underlying real-world events resulting in duplicates (e.g., that a relocation may lead to the storage of two records for the same customer, once before and after the relocation). Duplicates resulting from real-world events exhibit specific characteristics. For instance, duplicates resulting from relocations tend to have significantly different attribute values for all address-related attributes. Hence, existing approaches focusing on high similarity with respect to attribute values are hardly able to identify possible duplicates resulting from such real-world events. To address this issue, we propose an approach for event-driven duplicate detection based on probability theory. Our approach assigns the probability of being a duplicate resulting from real-world events to each analyzed pair of records while avoiding limiting assumptions (of existing approaches). We demonstrate the practical applicability and effectiveness of our approach in a real-world setting by analyzing customer master data of a German insurer. The evaluation shows that the results provided by the approach are reliable and useful for decision support and can outperform well-known state-of-the-art approaches for duplicate detection.

Keywords: duplicate detection, record linkage, entity resolution, data quality

1 Introduction

Organizations continue to rely more and more on large amounts of data to gain competitive advantage and to support decision-making (Ngai et al., 2017). However, poor data quality impedes organizations from generating high value based on their data (Heinrich et al., 2018b; Heinrich et al., 2018a; Moges et al., 2016). For instance, in a survey by Experian Information Solutions (2016), 83% of participants indicated that data quality problems have hurt their business objectives. According to Gartner, poor data quality is estimated to cost organizations on average $9.7 million per year (Moore, 2018). One of the most prevalent and critical reasons for poor data quality are *duplicates* (Fan, 2015; Helmis and Hollmann, 2009), pairs of records that represent the same real-world entity (Draisbach and Naumann, 2011). Duplicates are known to cause a large variety of problems, for instance misjudgements of customers (Bleiholder and Schmid, 2015), incorrect strategic and operational decisions (Helmis and Hollmann, 2009) and additional operative expenses (Draisbach, 2012). Thus, detecting duplicates has long been recognized to be of crucial importance in many areas, such as master data management, data warehousing, customer relationship management (CRM), fraud detection and healthcare (Elmagarmid et al., 2007; Fan, 2015; Hua and Pei, 2012). Major goal of approaches for duplicate detection is to identify such duplicates in order to allow, for instance, a subsequent merge into a single (“golden”) record.

Duplicates in datasets occur due to different reasons. For instance, failures during (repeated) data capturing which provoke misreported values (e.g., typos) result in duplicates which differ to some extent but tend to have similar attribute values. Besides, a more complex cause for duplicates are events in the real-world which change how a real-world entity should correctly be stored in a dataset hosted by an organization. For example, a marriage as an event can change the last name of a person. In case this event is unknown to the hosting organization, the last name of this person is not changed in the dataset. If this person is now stored a second time in the dataset, for example because s/he buys a product of the organization or concludes a contract, this leads to two stored records with a different value of the attribute *last name*, but representing the same person. In the following, we refer to events such as marriage as "real-world events", because they change the "state" of the considered real-world entity. Existing approaches for duplicate detection typically focus on (syntactic) similarities (cf. Section 3) and thus do not aim to cope with real-world events and their methodical consideration.

However, taking real-world events into account is crucial for duplicate detection in many application areas such as CRM. This importance is underlined by several studies. For instance, Schönfeld (2007) analyzed a customer database of a company with more than 20 million customers. Here, every year about 2 million customers changed their place of residence (event "relocation") and 60,000 got divorced, resulting in a large number of duplicates. Ignoring such data quality defects resulted in an annual loss of more than EUR 2 million for the company just based on inadequate customer contacts (Franz and von Mutius, 2008). In a B2B-context, analyses of Kraus (2004) on a dataset of business customers document that the contact person changed with a rate of 20% to 35% per year, depending on their position. This means that duplicates were caused by events such as "promotion" and "relocation" within a company. To summarize, in these studies "changes" are interpretable as real-world events resulting in a large number of duplicates. More generally, duplicates in datasets are often caused by real-world events.

Considering real-world events adds uncertainty to the task of identifying duplicates, when the organization hosting the considered dataset does not know whether such events occurred (cf. Section 2). We argue that the principles and the knowledge base of probability theory are adequate and valuable, providing well-founded methods to describe and analyze situations under uncertainty. More precisely, we base our approach on probability theory and aim to give an indication rather than a certain statement in regard to whether a specific pair of records is a duplicate resulting from a real-world event. Thus, the presented event-driven approach for detecting duplicates provides a way to detect duplicates not yet targeted by existing approaches (cf. Section 3). Moreover, the proposed approach addresses an important and relevant data quality problem.

The remainder of the paper is structured as follows: In Section 2, we introduce our problem context and a running example, which is then used to discuss related work and to clarify the research gap in Section 3. In Section 4, we step-by-step develop our event-driven probability-

based approach for duplicate detection. Section 5 contains the evaluation of the practical applicability and effectiveness of the approach using real-world customer master data from a German insurer. Finally, we conclude, reflect on limitations and provide an outlook on future research.

2 Problem Context and Running Example

We illustrate our problem context using a customer dataset, which serves as running example throughout the paper. A real-world event typically resulting in a large number of duplicates in customer datasets is relocation. This event changes the place of residence of a customer, which is usually represented by address-related attributes such as *street*, *ZIP code* and *city*, while other attributes such as *first name*, *last name* and *date of birth* remain unchanged. If a customer is once more stored in a customer dataset after relocation, two records representing the same customer emerge. Table 1 shows four customer records and their respective values for the attributes *first name*, *last name*, *street*, *ZIP code*, *city* and *date of birth*. The first two records with IDs 1 and 2 exhibit a typical pattern for the real-world event relocation: All non-address-related attribute values (*first name, last name* and *date of birth*) agree whereas all address-related attribute values (*street, ZIP code* and *city*) differ significantly. Using a non-event-driven approach for duplicate detection may require high similarity with respect to attribute values to identify possible duplicates. This may lead to a false negative error in this case of a duplicate. This is because not considering the event relocation means that the cause behind the significant differences between the address-related attribute values is neglected. Thus, the records with IDs 1 and 2 may be incorrectly classified as non-duplicate due to significant differences with respect to some attributes (i.e., *street, ZIP code* and *city*). On the other hand, pairs of similar records with different values for some specific attributes may not only result from real-world events but also by pure chance. In the context of the customer dataset, one could think about two different Mary Smiths living in different cities and sharing the same date of birth. Therefore, it cannot be said with certainty whether pairs of records exhibiting the typical relocation pattern are indeed duplicates or not without a so-called real-world check. Moreover, the records with IDs 3 and 4 in Table 1 illustrate that even more complex cases exist: Here, only the values for *city* and *street* differ while typos or other errors complicate the analysis. To conclude, the example illustrates that just based on the mere records, it cannot be said with certainty whether a pair of records is a duplicate or not. We refer to this fact as the first layer of uncertainty. In addition, it is not clear whether real-world entities were stored in a dataset multiple times as consequence of a real-world event, causing duplicates. We refer to this fact as additional second layer of uncertainty. These aspects emphasize the need for an event-driven probability-based approach for duplicate detection.

ID	First Name	Last Name	Street	ZIP Code	City	Date of Birth
1	Mary	Smith	Main Street 1	98101	Seattle	18.07.1967
2	Mary	Smith	South Road 3	10005	New York	18.07.1967
3	Franklin	Jefferson	Jennifer Road 17	90120	Beverly Hills	20.02.1952
4	Franklin	Jefferson	Jenifer Road 17	90120	Los Angeles	20.02.1952

Table 1. Illustration of four Records in a Customer Dataset (Running Example)

3 Related Work

Literature provides many well-known approaches for duplicate detection (Christen, 2012; Elmagarmid et al., 2007; Winkler, 2006). The two major strategies for duplicate detection are *probability-based* vs. *deterministic* duplicate detection (Tromp et al., 2011). As our focus is on duplicates resulting from possible real-world events, a probability-based approach addressing both layers of uncertainty is needed. In addition, commonly used deterministic approaches often rely on very complex handcrafted rulesets (Hettiarachchi et al., 2014) and cannot outperform probability-based approaches (Tromp et al., 2011). Thus, in the following, we focus on probability-based approaches.

Based on prior work of Newcombe et al. (1959), the classical probability-based framework for duplicate detection was presented by Fellegi and Sunter (1969). This work serves as foundation for various probability-based approaches for duplicate detection (e.g., Belin and Rubin, 1995; DuVall et al., 2010; Schürle, 2005; Steorts, 2015; Steorts et al., 2016; Thibaudeau, 1992; Winkler, 1988; Winkler, 1993). Approaches based on the work of Fellegi and Sunter (1969) share the main concept of grasping syntactical agreements and similarities as distinctive characteristic of duplicates. They address the first layer of uncertainty by modeling the probability of a given pair of records to be a duplicate conditioned on agreements and similarities of the respective records' attribute values. To quantify the syntactical agreements and similarities, a comparison vector is introduced. More precisely, the components of the comparison vector are values between zero and one according to the outcome of comparisons performed on the attribute values of a pair of records. These comparisons might lead to binary outcomes (e.g., "value of attribute *a* agrees") or continuous outcomes (e.g., "Jaro-similarity of the values of attribute *a*"). Based on its comparison vector, each analyzed pair of records is classified into one of three mutually exclusive subsets: the set of duplicates, the set of non-duplicates and the set of pairs requiring manual review.

Thereby, existing approaches based on the Fellegi-Sunter framework are defined based on two key limiting assumptions: (L1) the classification relies on syntactical agreements and similarities without considering the underlying causes of potential duplicates as second layer of uncertainty and (L2) the classification incorporates decision rules based on (L2a) independence or (L2b) monotonicity assumptions. These assumptions, which are discussed in more detail in the following, may lead to misclassifications as the causes of duplicates have crucial influence on

the syntactical similarities (Lehti and Fankhauser, 2006) and the assumptions made are often violated in practical applications (cf., e.g., Belin and Rubin, 1995; Christen, 2008a; Thibaudeau, 1992).

Ad (L1): Not considering the underlying causes for duplicates can lead to false negatives as the semantics behind possible disagreements or low similarities for some attribute values are not grasped. In the running example in Table 1, the records with the IDs 1 and 2 represent the same person, which relocated. The disagreements for the address-related attribute values are represented in terms of low or zero values for the corresponding entries of the comparison vector. Approaches based on these syntactical comparisons are prone to wrongly classify such pairs as non-duplicates. This may lead to the potential misclassification of a large number of duplicates caused by relocations, or, more generally, real-world events. To the best of our knowledge, none of the existing approaches addresses the second layer of uncertainty resulting from real-world events causing duplicates.

Ad (L2a): The decision rule presented by Fellegi and Sunter (1969) relies on independence assumptions regarding the agreements across attributes (i.e., agreement/disagreement in one attribute does not influence agreement/disagreement in other attributes). Many subsequent approaches continue to use these independence assumptions even though they were shown to be violated in most practical applications (cf. Belin and Rubin, 1995; Thibaudeau, 1992). An example for an obvious violation of the independence assumption is provided in Table 1: If two records agree in their value with respect to the attribute *ZIP code*, an agreement for the attribute *city* becomes much more likely and vice versa (Tromp et al., 2011). Another example of dependencies in datasets is the fact that people form households (i.e., different real-world persons share the same address). As household members often have identical last names, this leads to natural dependencies between address-related attributes and the attribute *last name* (Thibaudeau, 1992). Therefore, approaches relying on independence assumptions tend to lead to inadequate results as these assumptions are violated in many practical applications.

Against this background, several works have tried to alleviate the independence assumptions. In particular, important contributions with respect to a relaxation of the independence assumptions have been presented by Thibaudeau (1992), Winkler (1993) as well as Larsen and Rubin (2001). Thibaudeau (1992) introduces a model tailored to account for certain dependencies among address-related attributes of non-duplicates based on an empirical correlation analysis. Winkler (1993) suggests to include a specified small set of interactions, for example all three-way interactions or a selection of interactions based on knowledge of some true duplicate statuses. This selection of modeled dependencies is taken up by Larsen and Rubin (2001). As these authors state, the selection of dependencies to be modeled relies on personal knowledge and experience. In addition, all of the proposed methods are only suitable for binary comparisons and do not cover continuous similarity-based comparisons (Lehti and Fankhauser, 2006). Moreover, the explicit modeling of all possible interactions between all attributes is computationally

expensive, as the number of parameters to be fitted rises strongly. The high number of parameters also causes potential overfitting. In summary, the proposed works are able to relax the independence assumptions hampering probability-based approaches. However, they suffer from limitations regarding practical applicability and complexity.

Ad (L2b): Other approaches (Ravikumar and Cohen, 2004; Lehti and Fankhauser, 2006) avoid the independence assumptions by replacing them with a monotonicity assumption. This monotonicity assumption states that higher similarities of attribute values lead to a higher probability of being a duplicate and vice versa. However, this assumption is also violated in many practical applications. For example, after a relocation, a high similarity of the new values to the respective old values for address-related attributes may or may not occur, depending on chance. In fact, even an increase in probability with lower similarity is possible. For instance, as illustrated by the records with the IDs 3 and 4 in Table 1, "Los Angeles" is often used instead of the actual city name "Beverly Hills". Such commonly used syntactical dissimilar attribute values are an example for the violation of the monotonicity assumption, as many pairs of records exhibit low similarities despite being duplicates (Christen, 2008). Therefore, approaches relying on the monotonicity assumption may also lead to inadequate results, as monotonicity is not guaranteed in many practical applications.

To conclude, existing approaches for probability-based duplicate detection are either based on limiting independence or monotonicity assumptions or are severely restricted in their applicability by relaxing the independence assumptions. Moreover, none of these approaches considers the second layer of uncertainty arising from the underlying causes for duplicates, which means, they are hardly able to identify possible duplicates resulting from real-world events. To address this research gap, we propose an event-driven probability-based approach for duplicate detection in the following.

4 A Novel Approach for Duplicate Detection

In this section, we present a novel approach for duplicate detection. First, we outline the general setting and the basic idea. Then, we discuss our approach, which comprises two steps. Finally, we outline possible ways to instantiate the approach.

4.1 General Setting and Basic Idea

We consider a dataset with records representing entities by means of attributes (e.g., a relation in a database). The set of records is denoted by $T = \{t_1, \dots, t_n\}$ and the set of attributes by $\{a_1, \dots, a_m\}$. In our running example of a customer dataset (cf. Table 1), for instance, the attribute value of *first name* for t_1 is *Mary*. The set $T \times T$ contains all pairs (t_i, t_j) of records. Pairs of records in $C \coloneqq \{(t_i, t_j) \in T \times T \mid i \neq j\}$ may possibly be a duplicate (e.g., if both records t_1 and t_2 represent the same real-world entity). The aim of approaches for duplicate detection is to analyze such pairs of records. Thereby, two layers of uncertainty have to be considered (cf.

Section 2):

1. Based on the mere records (i.e., only considering the attribute values of the records) it cannot be said with certainty whether a pair of records is a duplicate.
2. It is not clear whether real-world entities were stored in a dataset multiple times as consequence of a real-world event such as marriage or relocation, causing duplicates.

As described in the previous section, existing probability-based approaches addressing the first layer of uncertainty are defined based on limiting assumptions such as independence or monotonicity. Instead, our approach addresses this layer of uncertainty by determining a probability for each pair of records to be a duplicate while avoiding these assumptions. Moreover, the clear interpretation of the results of our approach as probabilities has further advantages. For instance, it allows the integration into a decision calculus (e.g., based on decision theory) to support decision-making in a well-founded manner.

Another disadvantage of existing approaches (cf. Section 3) is that they are hardly able to identify duplicates caused by real-world events, because the second layer of uncertainty is not considered. For instance, in a customer dataset, this can lead to a large number of undetected duplicates caused by relocations or marriages. In our approach the second layer of uncertainty is addressed by explicitly analyzing pairs of records in regard to being a duplicate caused by real-world events. More precisely, to represent the second layer of uncertainty, we model the interrelation of a pair of records as an outcome in the probability space $(\Omega, 2^{\Omega}, P)$. Thereby, Ω includes all outcomes representing a duplicate and the complementary outcome that the pair of records is no duplicate. We particularly refer to outcomes $E_1, ..., E_r$ representing a duplicate caused by real-world events. More precisely, E_k represents the outcome "duplicate caused by real-world event k" ($k = 1, \dots, r$) with r as the number of relevant real-world events. For instance, the real-world event "marriage" may be expressed by outcome E_1 and the real-world event "relocation" by outcome E_2. The probability measure P assigns a probability to each event. For our approach, we focus on the values $P(E_k)$ for each real-world event k, representing the probability that a considered pair of records is a duplicate caused by real-world event k. In this way, the probability for a pair of records to be a duplicate caused by, for instance, a marriage or a relocation is specified.

The basic idea of our approach for duplicate detection is to accurately model the probability space $(\Omega, 2^{\Omega}, P)$ by grounding the approach on the real-world events $E_1, ..., E_r$ (Step 1) and formalizing the probabilities $P(E_k)$ for all analyzed pairs of records (Step 2).

4.2 Event-driven Approach for Duplicate Detection

Our approach consists of two steps. First, we ground our approach on real-world events causing duplicates. Second, we formalize the conditional probability that a pair of records is a duplicate caused by a specific real-world event.

4.2.1 Step 1: Grounding the Approach on Real-world Events

The probability space $(\Omega, 2^{\Omega}, P)$ defined in the previous section provides the basis for our approach. To address shortcomings of existing approaches, we explicitly model real-world events causing duplicates within this probability space. In the first step, these real-world events need to be determined. For a specific dataset, the real-world events causing duplicates can be obtained in multiple ways. For example, they can be derived from analysis or may be inferred by the user or a domain expert (cf. Section 4.3.1). Each real-world event is included in the set of outcomes Ω. This way, the set $\{E_1, \dots, E_r\} \subset \Omega$ is formed with E_k representing the outcome "duplicate caused by real-world event k". For instance, outcome E_1 may represent duplicates caused by relocations, outcome E_2 duplicates caused by marriages, and outcome E_3 duplicates caused by the combination of these two real-world events.

Our approach aims to determine a probability for a pair of records to be a duplicate caused by a real-world event. This probability is estimated by the probability measure P. $P(E_k)$ represents the probability that a pair of records is a duplicate caused by real-world event k. Thus, for a probability estimation based on the outcomes $E_1, \dots, E_r$, the corresponding values $P(E_1), \dots, P(E_r)$ have to be determined for each pair of records. These probabilities depend on the pair of records (t_i, t_j). In particular, characteristic patterns of interrelations – possibly pointing to a specific real-world event – can be considered for the probability estimation. To give an example, matching first name and last name but different address and ZIP code may indicate a relocation. More generally, such characteristic patterns may for instance include matching or missing attribute values and are given by (a subset of) the attribute values of the two records t_i and t_j. Thus, when determining the probabilities $P(E_k)$, it is necessary to condition on (t_i, t_j). Moreover, further data which helps to determine $P(E_k)$ more accurately may be available. This may be data derived from the dataset to be analyzed (different from the attribute values of t_i and t_j) or external data. For illustration purposes, consider the case of a customer dataset. Here, ceteris paribus, two records with the same very rare last name are more likely to be a duplicate than two records with the same very common last name. Thus, for instance, useful additional data derived from the dataset to be analyzed may be relative frequencies of last names, indicating how common the last name of the pair of records is. External additional data can, for instance, be empirical data from a Federal Statistical Office providing the number of relocations per year for the respective geographical region. Conditioning on such additional data Z allows to account for any further data available and more accurately determine $P(E_k)$. If additional data is not available for a particular pair of records, missing values may be replaced by estimations (e.g., by integrating over the probability space).

To sum up, the approach is grounded on real-world events by modeling the probability space $(\Omega, 2^{\Omega}, P)$ as follows: Relevant real-world events are included as outcomes $E_1, \dots, E_r$ in Ω. Based on this, the conditional probabilities $P\big(E_k|(t_i, t_j), Z\big)$ need to be determined for each real-world event k contained in Ω. Then, $P\big(E_k|(t_i, t_j), Z\big)$ represents the probability that the

pair of records $(t_i, t_j) \in C$ resulted from k, conditioned on the pair of records (t_i, t_j) and additional data Z. Thus, $P(E_k|(t_i, t_j), Z)$ expresses the probability that (t_i, t_j) is a duplicate resulting from the real-world event k.

4.2.2 Step 2: Formalizing the Conditional Probabilities

As introduced above, the term $P(E_k|(t_i, t_j), Z)$ represents the probability that (t_i, t_j) is a duplicate resulting from the real-world event k. Conditioning on the pair of records (t_i, t_j) and on additional data Z needs to be formalized to enable an application. To do so, both the information on the interrelation of the pair of records and the additional data Z can be expressed by numerical values. To give an example, the agreement or non-agreement of attribute values of (t_i, t_j) can be indicated by the values 1 and 0. For concise representation, these numerical values are combined in a vector which is called *feature vector* in the following. It can be seen as a generalization of the comparison vectors used in other approaches for duplicate detection (e.g., using syntactical similarity measures). To allow for maximum flexibility with respect to the interrelations and additional data used, no kind of independence, monotonicity or other specific interrelation between the components of our feature vector is assumed (cf. research gap at the end of Section 3). The feature vector may be defined differently for each real-world event k to allow taking the specific aspects of real-world events into account. More precisely, the feature vector is formed by mapping (t_i, t_j) and Z onto a f_k-dimensional outcome-specific vector $\zeta_k := C \rightarrow \mathbb{R}^{f_k}, f_k \in \mathbb{N}$, so that it holds $P(E_k|(t_i, t_j), Z) = P\left(E_k|\zeta_k\left((t_i, t_j)\right)\right)$ for outcome E_k.

Duplicates caused by a real-world event often exhibit a particular characteristic. To identify pairs of records showing this characteristic, similarity measures for all attributes can be taken into account as component of ζ_k. For instance, in the context of our running example, the real-world events relocation and marriage lead to different specific characteristics of the resulting pairs of records: For duplicates caused by relocations, the similarity between the address-related attribute values is usually low, whereas for duplicates caused by marriages, the similarity between the last names is usually low. Additional data helpful for determining more accurate probability estimations can be incorporated into these components as well or into additional components of the feature vector. For instance, the rate of relocations depending on the age or marital status or the frequency of last names can be considered this way.

4.3 Possible Ways to instantiate the Approach

In the following, we describe how our approach can be instantiated. Both the identification of relevant real-world events as well as the determination of conditional probabilities are discussed.

4.3.1 Identification of relevant Real-world Events

Duplicate detection is an important task in many domains (Cohen and Richman, 2002; Fan,

2015; Hua and Pei, 2012; Lehti and Fankhauser, 2006). In each domain, different real-world events may lead to duplicates. Thus, for a dataset to be analyzed, the relevant real-world events need to be determined. We propose three different ways to obtain them: (a) Review of publicly available data and publications (e.g., from public or scientific sources), (b) Analysis of company-owned data, and (c) Surveying experts.

Ad (a): A promising option to identify relevant real-world events is to analyze publicly available data and publications. For instance, the German Federal Statistical Office offers detailed data about the population of Germany and thus for many typical attributes of master data. To give an example, extensive data on the migration in Germany is available at fine granular level. Moreover, publications can be reviewed to obtain the causes of duplicates in datasets. For example, Bilenko and Mooney (2002) discuss how differently used city names relate to duplicates in a restaurant database. Finally, publicly available datasets containing identified duplicates can be analyzed to determine relevant real-world events.

Ad (b): The dataset to be analyzed or other company-owned data may be examined. To obtain relevant real-world events, a sufficient number of duplicates can be (e.g., manually) identified. Afterwards, the underlying causes for these duplicates can be determined and categorized into different real-world events. Also, for instance, data about orders and transactions may be captured in multiple departments. Then, data captured by one department may be used to support detecting duplicates in another.

Ad (c): Another possible way is surveying experts. This may be reasonable if neither external nor internal data is available for analysis, if the analysis is too time-consuming and costly or if the major causes for duplicates in the dataset to be analyzed are already known by domain experts. For example, instead of analyzing a given dataset, a company's key account managers may be surveyed to determine the main causes for duplicates in the company's databases. For instance, key account managers may know about the common practice of some customers who intentionally create duplicate accounts in order to surreptitiously receive monetary bonuses for new customers, causing duplicates.

4.3.2 Determination of Conditional Probabilities

Our approach is based on the conditional probabilities $P(E_k|(t_i, t_j), Z)$ resp. $P\left(E_k|\zeta_k\left((t_i, t_j)\right)\right)$ for each real-world event k (cf. Step 2). We briefly describe two common ways to determine these probabilities: (i) Estimation based on training data and (ii) Estimation based on surveying experts.

Ad (i): The first possibility refers to the analysis of training data containing pairs of records for which it is known whether they are a duplicate caused by a specific real-world event or not. One way to obtain such data is to manually label a sample of potential duplicates (t_i, t_j) in the dataset to be analyzed. For example, if a customer dataset is to be assessed, a random sample of pairs of customer records (t_i, t_j) can be drawn and labelled by hand. To ensure reliable

results, the sample should be representative and sufficiently large, which can be underpinned using statistical tests. For domain experts, such a manual labelling is usually straightforward to carry out and can be performed with a high degree of reliability (i.e., expert estimations will not substantially change over time or between experts). Another possible source for training data is company-owned (historical) data. The historical data may, for example, stem from previous data quality projects. This represents an opportunity to reuse results of analyses (i.e., duplicates recognized by customer feedback) conducted in the past, not requiring additional effort. Finally, conducting a study is a further possibility to generate training data. For the example of a customer dataset, a random sample of pairs of records (t_i, t_j) can be drawn and the respective customers can be surveyed. This is equivalent to a real-world check for these pairs of records. Moreover the results of the survey can also be used to assess the duplicate status of the customers not part of the survey.

The feature vectors of the pairs of records in the training data can then be used to obtain an estimation for the conditional probability $P\left(E_k|\zeta_k\left(\left(t_i, t_j\right)\right)\right)$ for a pair of records (t_i, t_j). We propose two methods for this estimation: an interval-based approach and kernel density estimation. Both of these methods can be performed with a high degree of automation and require little computational effort.

For the interval-based approach, H sets of intervals $I_{h,l} \subset \mathbb{R}, 1 \leq l \leq f_k$ and $1 \leq h \leq H$ (with $H \in \mathbb{N}$) are defined. Each set contains f_k intervals (one interval for each dimension of the feature vector ζ_k). Then, for each set of intervals a multidimensional interval $I_h \subset \mathbb{R}^{f_k}$ with $I_h = I_{h,1} \times I_{h,2} \times \ldots \times I_{h,f_k}$ is constructed. Finally, the relative frequency $\varrho_{k,h}$ of duplicates caused by real-world event k in I_h is calculated based on the training data. The relative frequency $\varrho_{k,h}$ can be determined efficiently (e.g., via a simple database query) and is used to estimate $P\left(E_k|\zeta_k\left(\left(t_i, t_j\right)\right)\right)$ for (t_i, t_j) with $\zeta_k\left(\left(t_i, t_j\right)\right) \in I_h$:

$$P\left(E_k|\zeta_k\left(\left(t_i, t_j\right)\right)\right) \approx \varrho_{k,h} \tag{1}$$

However, in some scenarios, it might be difficult to determine an appropriate set of intervals to apply the interval-based approach. Therefore, we further propose a nonparametric density estimation method called multivariate conditional kernel density estimation (Elgammal et al., 2002; Scott, 2015). Generally, any density function $P(x)$ of a random variable x can be estimated using a kernel density estimator $\hat{f}(x)$. Based on a sample x_i (with $i = 1, \ldots, n$) drawn from x, the distribution of x is estimated by summing up and normalizing multivariate kernel functions K placed over the values of x_i:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K(x - x_i) \approx P(x) \tag{2}$$

Typically, Gaussians are used as kernel functions. As the kernel density estimator asymptotically converges to the density function, it can be used even if the underlying density is not

known. In our setting, kernel density estimation can be used to estimate the conditional density $P\left(E_k|\zeta_k\left((t_i,t_j)\right)\right)$ for a pair of records (t_i,t_j) based on training data. Here, the definition of conditional probabilities is applied and numerator and denominator are estimated separately by multivariate kernel density estimation:

$$P\left(E_k|\zeta_k\left((t_i,t_j)\right)\right)=\frac{P\left(\zeta_k\left((t_i,t_j)\right)\wedge E_k\right)}{P\left(\zeta_k\left((t_i,t_j)\right)\right)}\approx\frac{\hat{f}\left(\zeta_k\left((t_i,t_j)\right)\wedge E_k\right)}{\hat{f}\left(\zeta_k\left((t_i,t_j)\right)\right)} \quad (3)$$

Ad (ii): As another option, estimations of the conditional probabilities $P\left(E_k|\zeta_k\left((t_i,t_j)\right)\right)$ based on experts' assessments can be used. For example, to detect duplicates caused by marriages in a customer dataset, experts could be surveyed and asked to estimate the probability of a pair of records being a duplicate caused by a marriage for different attribute values of the customers.

5 Evaluation

In this section we evaluate (E1) the practical applicability and (E2) the effectiveness of our approach for duplicate detection. First, we discuss the reasons for selecting the case of an insurer and describe the analyzed customer dataset. Then, we show how the approach could be instantiated for this case. Finally, we present the results of its application and compare them to those of a state-of-the-art approach.

5.1 Case Selection and Dataset

To evaluate (E1) and (E2), the approach was applied to a customer dataset of a major German provider of life insurances. As the insurance contracts typically last a long time, the customers are likely to relocate during the contract term. If a customer contacts the insurer after relocation and the customer is not associated with the existing record, a duplicate record is stored. Such duplicates are a major source of data quality problems in the insurer's customer master data. However, this data is of particular importance for the insurer (e.g., for CRM). Hence, the insurer aimed to identify respective duplicates.

To apply and evaluate our approach with regard to detecting duplicates caused by a real-world event, the insurer provided us with a subset of its customer record data containing four master data attributes. Each record in this subset has a value for both the attribute *first name* and the attribute *last name*. In addition, for each customer street and house number are stored in the attribute *street*. Finally, the attribute *date of birth* is stored in a standard date format. Note that analogously to our running example, one would expect duplicates caused by relocations to have matching values for the attributes *first name*, *last name*, and *date of birth* but largely differing values for the attribute *street*. These attributes are typical for customer master data and were

used to apply and evaluate our approach. More precisely, 4,552 pairs of records – exclusively potential duplicates caused by the real-world event relocation – were analyzed.

Before applying the approach, an instantiation is necessary. This instantiation can then be reused for further applications of the approach in the respective domain. In our case, as we had access to a data expert of the insurer and thus also to further confidential data (e.g., the customers' bank accounts), we aimed to generate quality assured training data for our instantiation (cf. Section 4.3.2). Thus, 20% of the 4,552 pairs of records were drawn randomly. A predominantly manual search for duplicates caused by relocations was performed on the drawn pairs of records with the help of the data expert of the insurer. This careful search for duplicates with the aid of additional confidential data ensured an accurate identification of duplicates caused by relocations. The drawn pairs of records were labelled accordingly as duplicates vs. non-duplicates and used for the instantiation of our approach. Using only 20% training data gives credit to the fact that generating training data may be costly and time-consuming. Indeed, the application of our approach would not be practical if the true duplicate status for all or most pairs of records needed to be known. For evaluation purposes only, the remaining 80% of the data were labelled as well. In total, 414 pairs of records in the given dataset (i.e., 9.1% of the analyzed 4,552 pairs of records) were duplicates caused (exclusively) by the real-world event relocation. In the following, we present the results using 20% training data; however, all evaluations have also been conducted using different percentages of training data between 5% and 50% without substantially differing results.

5.2 Instantiation of our Approach for Duplicate Detection

For illustration purposes, we focused on the real-world event relocation. Thus, the set of considered real-world events consisted of this single event. The respective feature vector had to be fitted to the typical characteristics of duplicates caused by relocations. To capture these characteristics, we let the feature vector comprise four string-based similarities based on the attribute values of each pair of records. Being a frequent and established choice for attributes representing names (Cohen et al., 2003), Jaro-Winkler similarity (Winkler, 1990) was selected for the attributes *first name*, *last name* and *street*. The Jaro-Winkler similarity of two strings accounts for the number of matching characters as well as the minimum number of character transpositions required to transform one string into the other, putting more weight on the first characters. To weight all digits equally, Levenshtein similarity (Levenshtein, 1966) was used for the attribute *date of birth*. The Levenshtein similarity accounts for the minimum number of edits (i.e., deletions, insertions and substitutions) required to transform one string into the other.

In Section 4.3.2 two methods for probability estimations using a labelled training dataset were proposed: interval method and kernel density estimation. To instantiate the approach using the interval method, we identified disjoint relevant multidimensional intervals so that all other intervals could be excluded from further analysis due to not containing any duplicates. In total, the pairs of records fell into one of 22 relevant intervals. The relative frequency of duplicates

in each interval was calculated based on the training data. The interval with the highest relative frequency of duplicates contained 59 pairs of records, of which 57 were duplicates caused by relocations, resulting in a relative frequency of 96.6%. In the following, the instantiation of our approach based on the interval method is referred to as "Intervals".

For the instantiation of our approach based on the kernel density estimation method, in the following referred to as "KDE", we used a common implementation presented by Seabold and Perktold (2010) and the same feature vectors and training data as in the interval method. Further, we aimed to include knowledge about the frequencies of first and last names to analyze the benefits of using additional metadata. With this selection we aim to illustrate the potential of considering additional metadata, but it is certainly also promising to consider further additional data such as the customers' age. Two records sharing a rare value for the attributes *first name* or *last name* are, ceteris paribus, more likely to be a duplicate than if the values are common. For instance, ceteris paribus, two records with the (common) name "Mary Smith" are less likely to be a duplicate than two records with the (rare) name "Franklin Jefferson". To consider this fact, we extended the feature vector with two supplementary components for additional metadata reflecting the rarity of the values for the attribute *first name* and *last name*, respectively. More precisely, we determined the number of records from the whole dataset whose values for *first name* or *last name*, respectively, corresponded to the respective attribute values of the analyzed pairs of records. Then, following inverse document frequency logic (Sparck Jones, 1972), the logarithm of the total number of records divided by the determined number of records was calculated and used as supplementary component of the feature vector. In the following, our instantiation based on kernel density estimation and extended feature vectors will be referred to as "KDE with metadata".

5.3 Application and Results

(E1) Practical applicability

The approach was implemented in Python and applied to the remaining 80% of the dataset. After initial instantiation, our approach could be applied in an automated manner without further manual configuration. This low effort underlines its practical applicability (E1). For each pair of records, the three instantiations described above yielded estimations for the conditional probabilities of being a duplicate caused by relocation. Figure 1 shows a histogram of the estimated probabilities.

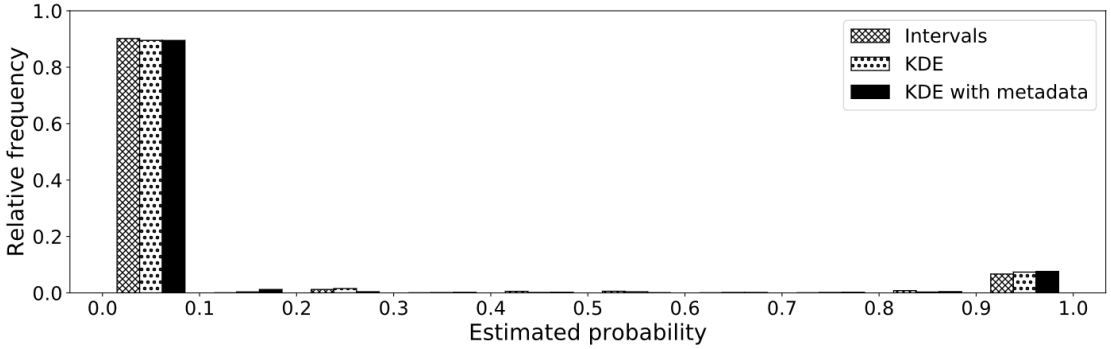

Figure 1. Histogram of the estimated Duplicate Probabilities

The relative frequency of pairs of records is given in ten bins of equal size according to the estimated probability. The approach assigned either a very low or a very high duplicate probability to the vast majority of pairs of records, regardless of the chosen instantiation (i.e., Intervals, KDE and KDE with metadata). Such a distribution of estimated duplicate probabilities is favorable, as it builds the basis for a clear and comprehensible classification. Additionally, Figure 1 illustrates that the distributions closely resemble the actual ratios of duplicates caused by relocations (9.1%) and non-duplicates (90.9%) in the dataset. To conclude, the approach could be applied and provided useful results (cf. also (E2) below) using 20% training data. After initial instantiation it could be applied repeatedly, in an automated manner and without determination of additional parameters or distributions. This supports both efficiency and practical applicability (E1).

Our approach aims to determine duplicate probabilities for pairs of records which can then be used to classify into duplicates and non-duplicates. Therefore, to evaluate the effectiveness (E2), we first analyze whether the proposed approach is able to provide duplicate probability estimations of high quality (E2.1). Then, the effectiveness of our approach with respect to the classification is assessed (E2.2). The results are analyzed for the three instantiations of our approach based on the different probability estimation methods: "Intervals", "KDE" and "KDE with metadata".

(E2.1) Effectiveness with respect to the estimated duplicate probabilities

The duplicate probabilities determined by our approach can be integrated into decision calculus. For instance, decisions regarding whether to assess a pair of records as a duplicate or whether to perform a data quality improvement measure can be made. To enable well-founded decisions, it must be ensured that the estimated probabilities correspond to the actually observed relative frequencies, which can be assessed in terms of reliability (Hoerl and Fallin, 1974; Murphy and Winkler, 1977; Murphy and Winkler, 1987; Sanders, 1963). In our context reliability expresses that the mean of the estimated duplicate probabilities in an interval must be approximately equal to the relative frequency of duplicates in that interval. Reliability is commonly evaluated using the reliability curve (Bröcker and Smith, 2007). To calculate the points of this curve, the data is arranged to bins according to the estimated duplicate probability. Then, the mean of the esti-

mated duplicate probability ("mean estimated probability") as well as the actual relative frequency of duplicates caused by the real-world event relocation ("fraction of positives") is calculated and plotted for each bin. For a perfectly reliable estimation, all points of the reliability curve would lie on the diagonal. Reliability can also be quantitatively assessed in terms of the reliability score, which is defined as the mean squared deviation from the diagonal weighted by the number of test cases in each bin (Murphy, 1973). Therefore, the smaller the value of the reliability score, the smaller the discrepancy between the estimated probabilities and the actually observed relative frequencies. The left section of Figure 2 shows the reliability curves for the three instantiations of our approach. To obtain a sufficient number of test cases in each bin, the number of bins was set to four. The results show that our approach assigns reliable probabilities to the pairs of records regardless of the instantiation as all three curves follow the diagonal rather closely. The duplicate probabilities estimated by our approach exhibited the best fit for the instantiation based on KDE with metadata (reliability score of 0.0025%). This underlines the advantage of our approach of being able to incorporate additional metadata.

Based on the duplicate probabilities estimated by our approach, duplicates can be distinguished from non-duplicates. Thus, to evaluate this aspect, we determined the discrimination of the estimated duplicate probabilities. The discrimination was assessed in terms of the area under curve (AUC) under the receiver operating characteristic (ROC) curve (Hanley and McNeil, 1982). The ROC curve is calculated by plotting the true positive rate of a classification based on the estimated duplicate probabilities against the false positive rate, when the classification threshold is varied. The ROC curves are given in the right section of Figure 2. For each instantiation, the ROC curve is very close to the curve of a perfect discrimination. With an area under the ROC curve of 97.39%, the probabilities based on KDE with additional metadata show the best discrimination in our application. Overall, these results support that the probabilities provided by our approach are able to discriminate between duplicates and non-duplicates. Further, they motivate the classification of pairs of records into duplicates and non-duplicates based on the approach, which is focused in the following.

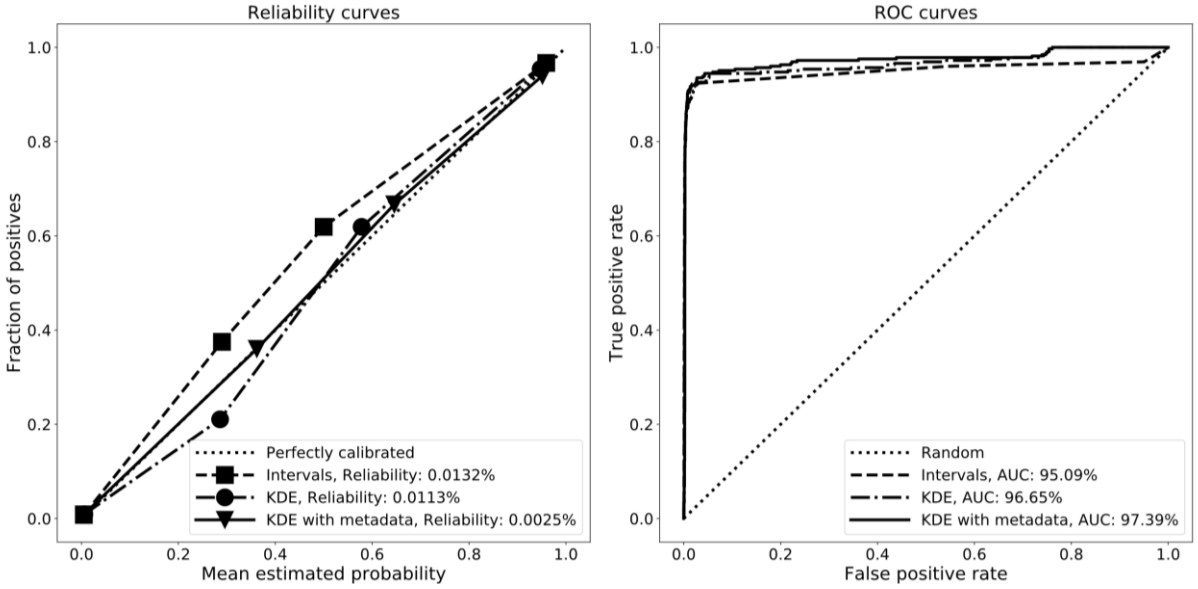

Figure 2. Reliability and Receiver Operating Characteristic Curves for the three Instantiations

(E2.2) Effectiveness with respect to classification into duplicates and non-duplicates

To evaluate the effectiveness of our approach with respect to the classification of pairs of records into duplicates and non-duplicates, the quality of the results is assessed and compared to the well-known state-of-the-art approach Febrl (Christen, 2008b) based on the Fellegi-Sunter framework (cf. Section 3). To classify into duplicates and non-duplicates, the pairs of records exhibiting an estimated duplicate probability above 50% were classified as duplicates and vice versa. This was done to represent the classification of each pair of records into its most probable class; however, as Figure 1 shows, other threshold values such as 30%, 40%, 60% or 70% could also be chosen and lead to very similar results.

A 5-fold inverse cross-validation was performed to account for variations caused by the random selection of 20% training data. Inverting the cross-validation (i.e., switching test data and training data compared to conventional cross-validation) ensures that in each fold only 20% of the dataset are used for training. To assess the quality of the classification into duplicates and non-duplicates, the performance measures accuracy, precision, recall and F-measure (F1) are provided in Table 2. F-measure combines precision and recall and is defined as their harmonic mean. As in the dataset the vast majority of pairs of records are non-duplicates (which is common for datasets in practice), the exact numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are also disclosed in Table 2. Please note that the sum of TP, FP, TN and FN is 18,208 in each row and that there were 1,656 positives (duplicates caused by relocations) and 16,552 negatives due to the inverse cross-validation performed. On the given dataset, our approach provides very promising results. Indeed, regardless of the method chosen for instantiation, the classification is effective. For instance, if KDE with metadata is used, the classification based on our approach is able to identify 88.59% of the duplicates contained in the dataset (recall). Without using metadata our approach is still able to

identify 86.90% (KDE) resp. 87.02% (Intervals) of duplicates. Our approach also exhibits maximum precision (i.e., the highest proportion of pairs of records classified as duplicates which actually are duplicates) when instantiated with KDE with metadata, with a value of 93.26%. Overall, using this instantiation leads to a very high accuracy as 98.38% of the pairs of records are correctly classified, and to the maximal value of 0.9086 for the F-measure, stressing the advantages of integrating additional metadata in the estimation.

To compare our results with the results of Febrl, a configuration of the syntactic similarity measures for each attribute used by Febrl was necessary. Thereby, the same similarity measures as in the instantiation of our approach were chosen. The thresholds for classification were automatically set by the optimal threshold model provided by Febrl. The bin width required by Febrl was chosen carefully and optimized to obtain best results for this method. Febrl classified almost all pairs of records as non-duplicates. Most pairs indeed are non-duplicates. This unbalance of the dataset means that Febrl was able to achieve a rather high accuracy of 93.67% (cf. Table 2) despite its difficulty to identify actual duplicates. As Febrl was very restrictive with judging pairs of records to be a duplicate, the few pairs of records identified as duplicates by Febrl were almost all correctly classified, resulting in a high precision of 94.37%. However, Febrl mainly just identified the rather obvious duplicates, leading to this high precision but a critically low recall. More precisely, Febrl was not able to detect the majority of duplicates, identifying only 32.37% of them as indicated by the recall. Reasons for this fact have already been discussed in Section 3: Indeed, real-world events such as relocations can lead to a large number of false negatives in approaches based on the Fellegi-Sunter framework. The low number of identified duplicates also resulted in an unsatisfactory F-measure of 0.4820 and a ROC AUC value of only 79.58% compared to values of over 95% for our approach (cf. Figure 2). Overall, based on the given dataset our approach seems much better suited to identify duplicates caused by relocations than Febrl, regardless of the instantiation method. In particular, it was able to handle the unbalance in the dataset well and identified almost all duplicates.

	Accuracy	Precision	Recall	F1	TP	FP	TN	FN
Intervals	98.17%	92.43%	87.02%	0.8964	1,441	118	16,434	215
KDE	98.21%	92.96%	86.90%	0.8983	1,439	109	16,443	217
KDE with metadata	**98.38%**	93.26%	**88.59%**	**0.9086**	**1,467**	106	16,446	**189**
Febrl (Optimal Threshold)	93.67%	**94.37%**	32.37%	0.4820	536	**32**	**16,520**	1,120

Table 2. Performance Measures for Classification into Duplicates and Non-duplicates

To conclude, our approach showed promising results regarding (E2.2) the performance of the classifier based on the estimated duplicate probabilities. As good results with respect to (E2.1) and (E2.2) could be achieved using only 20% training data, the practical applicability (E1) of the approach is supported.

6 Conclusion, Limitations and Future Work

Duplicate detection is an important issue in both research and practice. In this paper, we present an event-driven probability-based approach for this task. It aims at determining the probability for a pair of records to be a duplicate caused by a real-world event (e.g., relocating customers). Existing approaches are hardly able to identify such duplicates, which we address by explicitly modeling real-world events in a probability space. Moreover, the practical applicability and the effectiveness of the approach are evaluated based on real-world customer master data from a German insurer. The approach neither relies on limiting assumptions (e.g., independence or monotonicity) nor suffers from restrictions in its applicability like existing probability-based approaches. Additionally, in contrast to existing approaches, our approach is able to determine probabilities regarding different possibly underlying causes for a duplicate. Both probability and cause may be especially helpful for decision-making. More precisely, due to the interpretation of the results of our approach as probabilities, the integration into a decision calculus (e.g., expected value calculus) can be done easily and in a well-founded manner. The evaluation shows that the provided probabilities for being a duplicate are reliable and useful for decision support. Furthermore, when using the probabilities for a classification into duplicates and non-duplicates, the presented approach showed promising results and outperformed the well-known state-of-the-art approach Febrl.

Nevertheless, our work also has limitations which may constitute the starting point for future research. In this paper we focused on detecting duplicates caused by real-world events. Future research could explore whether failures during data capturing (e.g., mistakes caused by mishearing) can also be successfully modeled as such "events". Furthermore, the approach was applied to a real-world customer dataset of an insurer. Future research could evaluate it on further datasets containing master data. Moreover, the approach should be applied to datasets from other contexts, focusing on different real-world events. Further evaluations (e.g., on synthetical datasets) could also provide interesting insights regarding different possibilities to instantiate the approach and which additional data to use for effectiveness.

7 References

Belin, T. R. and D. B. Rubin (1995). "A method for calibrating false-match rates in record linkage" *Journal of the American Statistical Association (JASA)* 90 (430), 694–707.

Bilenko, M. and R. J. Mooney (2002). "Learning to combine trained distance metrics for duplicate detection in databases". In: *Proceedings of the 11th International Conference on Information and Knowledge Management*, pp. 1–19.

Bleiholder, J. and J. Schmid (2015). "Datenintegration und Deduplizierung". In *Daten-und Informationsqualität: Auf dem Weg zur Information Excellence. 2. Auflage*, 121–140. Heidelberg: Vieweg + Teubner.

Bröcker, J. and L. A. Smith (2007). "Increasing the reliability of reliability diagrams" *Weather and Forecasting* 22 (3), 651–661.

Christen, P. (2008a). "Automatic record linkage using seeded nearest neighbour and support vector machine classification". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 151–159.

Christen, P. (2008b). "Febrl. A freely available record linkage system with a graphical user interface". In: *Proceedings of the 2nd Australasian Workshop on Health Data and Knowledge Management*, pp. 17–25.

Christen, P. (2012). *Data matching. Concepts and techniques for record linkage, entity resolution, and duplicate detection:* Springer-Verlag.

Cohen, W. W. and J. Richman (2002). "Learning to match and cluster large high-dimensional data sets for data integration". In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 475–480.

Draisbach, U. (2012). *Partitionierung zur effizienten Duplikaterkennung in relationalen Daten:* Springer Vieweg.

Draisbach, U. and F. Naumann (2011). "A generalization of blocking and windowing algorithms for duplicate detection". In: *IEEE International Conference on Data and Knowledge Engineering (ICDKE 2011)*, pp. 18–24.

DuVall, S. L., R. A. Kerber and A. Thomas (2010). "Extending the Fellegi-Sunter probabilistic record linkage method for approximate field comparators" *Journal of Biomedical Informatics* 43 (1), 24–30.

Elgammal, A., R. Duraiswami, D. Harwood and L. S. Davis (2002). "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance" *Proceedings of the IEEE* 90 (7), 1151–1163.

Elmagarmid, A. K., P. G. Ipeirotis and V. S. Verykios (2007). "Duplicate record detection. A survey" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 19 (1), 1–16.

Experian Information Solutions (2016). *Building a business case for data quality.* Experian Information Solutions.

Fan, W. (2015). "Data Quality. From Theory to Practice" *ACM SIGMOD Record* 44 (3), 7–18.

Fellegi, I. P. and A. B. Sunter (1969). "A theory for record linkage" *Journal of the American Statistical Association (JASA)* 64 (328), 1183–1210.

Franz, T. and C. von Mutius (2008). *Kundendatenqualität - Ein Schlüssel zum Erfolg im. Kundendialog:* Swiss CRM Forum 2008.

Hanley, J. A. and B. J. McNeil (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve" *Radiology* 143 (1), 29–36.

Heinrich, B., D. Hristova, M. Klier, A. Schiller and M. Szubartowicz (2018a). "Requirements for Data Quality Metrics" *Journal of Data and Information Quality (JDIQ)* 9 (2), 12.

Heinrich, B., M. Klier, A. Schiller and G. Wagner (2018b). "Assessing data quality – A probability-based metric for semantic consistency" *Decision Support Systems (DSS)* 110, 95–106.

Helmis, S. and R. Hollmann (2009). *Webbasierte Datenintegration. Ansätze zur Messung und Sicherung der Informationsqualität in heterogenen Datenbeständen unter Verwendung eines vollständig webbasierten Werkzeuges:* Springer Vieweg.

Hettiarachchi, G. P., N. N. Hettiarachchi, D. S. Hettiarachchi and A. Ebisuya (2014). "Next generation data classification and linkage. Role of probabilistic models and artificial intelligence". In: *Global Humanitarian Technology Conference (GHTC)*, pp. 569–576.

Hoerl, A. E. and H. K. Fallin (1974). "Reliability of subjective evaluations in a high incentive situation" *Journal of the Royal Statistical Society: Series A (General)* 137 (2), 227–230.

Hua, M. and J. Pei (2012). "Aggregate queries on probabilistic record linkages". In: *Proceedings of the 15th International Conference on Extending Database Technology*, pp. 360–371.

Kraus, C. (2004). *Address- und Kundendatenbanken für das Direktmarketing. Aufbau, Pflege, Nutzung:* Businessvillage.

Larsen, M. D. and D. B. Rubin (2001). "Iterative automated record linkage using mixture models" *Journal of the American Statistical Association (JASA)* 96 (453), 32–41.

Lehti, P. and P. Fankhauser (2006). "Unsupervised duplicate detection using sample non-duplicates" *Journal on Data Semantics VII*, 136–164.

Levenshtein, V. I. (1966). "Binary codes capable of correcting deletions, insertions, and reversals" *Soviet Physics Doklady* 10, 707–710.

Moges, H.-T., V. van Vlasselaer, W. Lemahieu and B. Baesens (2016). "Determining the use of data quality metadata (DQM) for decision making purposes and its impact on decision outcomes - An exploratory study" *Decision Support Systems (DSS)* 83, 32–46.

Moore, S. (2018). *How to Create a Business Case for Data Quality Improvement*. URL: http://www.gartner.com/smarterwithgartner/how-to-create-a-business-case-for-data-quality-improvement/ (visited on 03/25/2019).

Murphy, A. H. (1973). "A new vector partition of the probability score" *Journal of Applied Meteorology* 12 (4), 595–600.

Murphy, A. H. and R. L. Winkler (1977). "Reliability of subjective probability forecasts of precipitation and temperature" *Applied Statistics*, 41–47.

Murphy, A. H. and R. L. Winkler (1987). "A general framework for forecast verification" *Monthly Weather Review* 115 (7), 1330–1338.

Newcombe, H. B., J. M. Kennedy, S. J. Axford and A. P. James (1959). "Automatic linkage of vital records" *Science* 130 (3381), 954–959.

Ngai, E. W. T., A. Gunasekaran, S. F. Wamba, S. Akter and R. Dubey (2017). "Big data analytics in electronic markets" *Electronic Markets (EM)* 27 (3), 243–245.

Ravikumar, P. and W. W. Cohen (2004). "A hierarchical graphical model for record linkage". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pp. 454–461.

Sanders, F. (1963). "On subjective probability forecasting" *Journal of Applied Meteorology* 2 (2), 191–201.

Schönfeld, A. (2007). *AdressDrehScheibe – Regelbasierter Datenaustausch mit Open Source:* Open Source Meets Business 2007.

Schürle, J. (2005). "A method for consideration of conditional dependencies in the Fellegi and Sunter model of record linkage" *Statistical Papers* 46 (3), 433–449.

Scott, D. W. (2015). *Multivariate density estimation. Theory, practice, and visualization:* John Wiley & Sons.

Seabold, S. and J. Perktold (2010). "Statsmodels. Econometric and statistical modeling with python". In: *Proceedings of the 9th Python in Science Conference*, pp. 57–61.

Sparck Jones, K. (1972). "A statistical interpretation of term specificity and its application in retrieval" *Journal of Documentation* 28 (1), 11–21.

Steorts, R. C. (2015). "Entity resolution with empirically motivated priors" *Bayesian Analysis* 10 (4), 849–875.

Steorts, R. C., R. Hall and S. E. Fienberg (2016). "A Bayesian approach to graphical record linkage and deduplication" *Journal of the American Statistical Association (JASA)* 111 (516), 1660–1672.

Thibaudeau, Y. (1992). *The discrimination power of dependency structures in record linkage.* US Bureau of the Census.

Tromp, M., A. C. Ravelli, G. J. Bonsel, A. Hasman and J. B. Reitsma (2011). "Results from simulated data sets. Probabilistic record linkage outperforms deterministic record linkage" *Journal of Clinical Epidemiology* 64 (5), 565–572.

Winkler, W. E. (1988). "Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage". In: *Proceedings of the Section on Survey Research Methods*: American Statistical Association.

Winkler, W. E. (1990). *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.* US Bureau of the Census.

Winkler, W. E. (1993). "Improved decision rules in the fellegi-sunter model of record linkage". In: *Proceedings of Survey Research Methods Section*: American Statistical Association, pp. 274–279.

Winkler, W. E. (2006). *Overview of record linkage and current research directions.* US Bureau of the Census.

2.3 Paper 3: Requirements for Data Quality Metrics

Current Status	Full Citation
accepted and published (01/2018) in Volume 9, Issue 2 of *Journal of Data and Information Quality*	Heinrich, B., D. Hristova, M. Klier, A. Schiller, and M. Szubartowicz (2018). "Requirements for Data Quality Metrics". *Journal of Data and Information Quality (JDIQ)* 9 (2), 12.

Summary

This paper addresses RQ3 by proposing five requirements for data quality metrics. The proposed requirements are condensed from a variety of requirements stated in literature. Further, they are justified based on a decision-oriented framework. In particular, the requirements are shown to be indispensable for a metric that aims to support an economically oriented management of data quality and decision-making under uncertainty. Moreover, their clear definition makes it possible to easily and transparently verify them, which is crucial for application in practice. The applicability and efficacy of the requirements is demonstrated by applying them to evaluate five well-known data quality metrics for different data quality dimensions, showing that the requirements are neither impossible nor trivial to fulfill. Moreover, practical implications of the requirements are discussed.

The work is based on various concepts and methods from decision-making under uncertainty. In particular, concepts from this field such as decision matrices and expected payoffs are central to the decision-oriented framework and the justifications of the requirements. On this basis, the proposed set of requirements is shown to form an essential concept for the evaluation of existing data quality metrics as well as for the design of new metrics. With respect to application in practice, it is found that certain requirements are of particular relevance in various situations, for instance, to support decision-making when multiple related data quality assessments are performed over time. Overall, the paper thus supports decision-making under uncertainty and an economically oriented management of data quality.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style.

The paper as published by ACM is available at: https://doi.org/10.1145/3148238

Abstract:

Data quality and especially the assessment of data quality have been intensively discussed in research and practice alike. To support an economically oriented management of data quality and decision-making under uncertainty, it is essential to assess the data quality level by means of well-founded metrics. However, if not adequately defined, these metrics can lead to wrong decisions and economic losses. Therefore, based on a decision-oriented framework, we present a set of five requirements for data quality metrics. These requirements are relevant for a metric that aims to support an economically oriented management of data quality and decision-making under uncertainty. We further demonstrate the applicability and efficacy of these requirements by evaluating five data quality metrics for different data quality dimensions. Moreover, we discuss practical implications when applying the presented requirements.

Keywords: data quality, data quality assessment, data quality metrics, requirements for metrics

1 Introduction

Due to the rapid technological development, companies increasingly rely on data to support decision-making and to gain competitive advantage. To make informed and effective decisions, it is crucial to assess and assure the quality of the underlying data. 83% of the respondents of a survey conducted by Experian Information Solutions (2016) state that poor data quality has actually hurt their business objectives, and 66% report that poor data quality has had a negative impact on their organization in the last twelve months. Another report reveals that 84% of the CEOs are concerned about the quality of the data they use for decision-making (KPMG, 2016; Rogers et al., 2017). In addition, Gartner indicates that the average financial impact of poor data quality amounts to $9.7 million per year and organization (Moore, 2018). Overall, it is estimated that poor data quality costs the US economy $3.1 trillion per year (IBM Big Data and Analytics Hub, 2016). In the light of the current proliferation of big data with large amounts of heterogeneous, quickly-changing data from distributed sources being analyzed to support decision-making, assessing and assuring data quality becomes even more relevant (Buhl et al., 2013; Cai and Zhu, 2015; Flood et al., 2016; IBM Global Business Services, 2012). Indeed, the three characteristics volume, velocity and variety, often called the three Vs of big data, make the assurance of data quality increasingly challenging (e.g., due to the integration of various data sources or when considering linked data; cf. also Cappiello et al., 2016; Debattista et al., 2016). Thus, the consequences of wrong decisions are becoming even more costly (Rogers et al., 2017; SAS Institute, 2013). This has resulted in the addition of a fourth V (=veracity) reflecting the importance of data quality in the context of big data (Flood et al., 2016; IBM Big Data and Analytics Hub, 2016; Lukoianova and Rubin, 2014).

Data quality can be defined as “the measure of the agreement between the data views presented by an information system and that same data in the real world” (Orr, 1998, p. 67; cf. also Heinrich et al., 2009; Parssian et al., 2004). Data quality is a multi-dimensional construct (Eppler,

2003; Lee et al., 2002; Redman, 1996; Taleb et al., 2016) comprising different data quality dimensions such as accuracy, completeness, consistency and currency (Batini and Scannapieco, 2016; Wang et al., 1995). Each data quality dimension provides a particular perspective on the quality of data views. As a result, researchers have developed corresponding metrics for the quantitative assessment of these dimensions for data views (e.g., Ballou et al., 1998; Blake and Mangiameli, 2011; Even and Shankaranarayanan, 2007; Fisher et al., 2009; Heinrich et al., 2007; Heinrich et al., 2009; Heinrich et al., 2012; Heinrich and Hristova, 2016; Heinrich and Klier, 2015; Hinrichs, 2002; Wechsler and Even, 2012). Metrics assessing such data quality dimensions for data views and data values stored in IS are in the focus of this paper. In contrast, for instance metrics addressing the quality of data schemes are not directly considered.

Data quality metrics provide measurements for data views with greater (lower) metric values representing a greater (lower) level of data quality and each data quality level being represented by a unique metric value. They are needed for two main reasons. First, the metric values are used to support data-based decision-making under uncertainty. Here, well-founded data quality metrics are required to indicate to what extent decision makers should rely on the underlying data values. Second, the metric values are used to support an economically oriented management of data quality (cf., e.g., Heinrich et al., 2009; Wang, 1998). In this context, data quality improvement measures should be applied if and only if the benefits (due to higher data quality) outweigh the associated costs. To be able to analyze which data quality improvement measures are efficient from an economic perspective, well-founded data quality metrics are needed to assess (the changes in) the data quality level.

While both research and practice have realized the high relevance of well-founded data quality metrics, many data quality metrics still lack an appropriate methodical foundation as they are either developed on an ad hoc basis to solve specific problems (Pipino et al., 2002) or are highly subjective (Cappiello and Comuzzi, 2009). Hinrichs (2002), for example, defines a metric to assess the correctness of a stored data value ω as $DQ(\omega, \omega_m) \coloneqq \frac{1}{d(\omega,\omega_m)+1}$ where ω_m represents the the corresponding real-world value and d a domain-specific distance measure. For instance, as proposed by Hinrichs (2002), let $d(\omega, \omega_m)$ be the Hamming distance between the stored and the correct value (i.e., the number of positions at which the corresponding symbols of two data strings are different). Applying this metric to $(\omega, \omega_m) =$ ('Jefersonn','Jefferson') and $(\omega, \omega_m) =$ ('Jones','Adams') to determine the correctness of customers' surnames in a product campaign yields the following results: DQ('Jefersonn','Jefferson') $= \frac{1}{5+1} \approx 16.67\%$ and DQ('Jones','Adams') $= \frac{1}{4+1} = 20\%$. If the decision criterion in the product campaign is a metric value of at least 20%, a sales letter is sent to 'Jones', which will most probably not reach its destination, whereas no sales letter is sent to 'Jefersonn', which would much more likely reach its destination. To avoid such problems, both researchers and practitioners set out to propose requirements for data quality metrics (e.g., Even and Shankaranarayanan, 2007; Heinrich et al., 2007; Hüner, 2011; Loshin, 2010; Mosley et al., 2009; Pipino et al., 2002). Most of them,

however, did not aim at justifying the requirements based on a decision-oriented framework. As a result, the literature on this topic is fragmented and it is not clear which requirements are indeed relevant to support decision-making. Moreover, as some of the requirements leave room for interpretation, their verification is difficult and subjective. This results in a research gap which we aim to address by answering the following research question:

> Which clearly defined requirements should a data quality metric satisfy to support both decision-making under uncertainty and an economically oriented management of data quality?

To address this research question, we propose a set of five requirements, namely the existence of minimum and maximum metric values (R1), the interval scaling of the metric values (R2), the quality of the configuration parameters and the determination of the metric values (R3), the sound aggregation of the metric values (R4), and the economic efficiency of the metric (R5).

We analyze existing literature and justify this set of requirements based on a decision-oriented framework. As a result, our requirements support both decision-making under uncertainty and an economically oriented management of data quality. Data quality metrics which do not meet them can lead to wrong decisions and/or economic losses (e.g., because the efficiency of the metric's application is not ensured). Moreover, the presented requirements facilitate a well-founded assessment of data quality, which is crucial for supporting data governance initiatives (Allen and Cervo, 2015; Khatri and Brown, 2010; Otto, 2011; Weber et al., 2009) and an efficient data quality management (cf. also Cappiello and Comuzzi, 2009; Fan, 2015).

The need for such requirements is further supported by the discussions in other fields of research such as software engineering. For example, Briand et al. (1996) provide a universal set of properties for the sound definition of software measures. The proposed properties can be used by researchers to "validate their new measures" (p. 2) and can be interpreted as necessary requirements for software metrics. In addition, in the context of ISO/IEC standards the SQuaRE series aims to "assist those developing and acquiring software products with the specification and evaluation of quality requirements" (p. V in ISO/IEC 25020, 2007; cf. also Azuma, 2001). In particular, ISO/IEC 25020 provides criteria for selecting software quality measures with the same motivation as above.

The remainder of the paper is structured as follows. In the next section, we provide an overview of the related work and identify the research gap. Section 3 comprises the decision-oriented framework for our work. In Section 4, we propose a set of five requirements for data quality metrics which are defined and justified based on this framework. In Section 5, we demonstrate the applicability and efficacy of these requirements using five data quality metrics from literature. Section 6 contains a discussion of practical implications. The last section provides conclusions, limitations and directions for future research.

2 Related Work

In this section, we analyze existing works, which propose requirements for data quality metrics. Following the guidelines of standard approaches to prepare the related work (e.g., Webster and Watson, 2002; Levy and Ellis, 2006), we searched the databases ScienceDirect, ACM Digital Library, EBSCO Host, IEEE Xplore, and the AIS Library as well as the Proceedings of the International Conference on Information Quality (ICIQ) for the following search term and without posing a restriction on the time period: *("data quality" and metric* and requirement*) or ("data quality" and metric* and standard*) or ("information quality" and metric* and requirement*) or ("information quality" and metric* and standard*)*. This search led to 136 papers which were manually screened based on title, abstract, and keywords. The remaining 43 papers were analyzed in detail and could be divided into three disjoint categories A, B and C. Category A comprises requirements for data quality metrics and data quality metric values from a methodical perspective. Category B contains requirements concerning the *general data quality assessment process* in an *organization* (e.g., measurement frequency). Category C consists of requirements and (practical) recommendations for the *concrete organizational integration* of data quality metrics (e.g., within business processes). Regarding our research question, we focused on Category A comprising five relevant papers on which we performed an additional forward and backward search, resulting in a total of eight relevant papers discussed in the following.

Pipino et al. (2002) propose the functional forms *simple ratio, min or max operation*, and *weighted average* to develop data quality metrics. *Simple ratio* measures the ratio of the number of desired outcomes (e.g., number of accurate data units) to the total number of outcomes (e.g., total number of data units). *Min or max operation* can be used to define data quality metrics requiring the aggregation of multiple assessments, for instance on the level of data values, tuples, or relations. Here, the minimum (or maximum) value among the normalized values of the single assessments is calculated. *Weighted average* is an alternative to the min or max operation and represents the weighted average of the single assessments. The major goal of Pipino et al. (2002) is to present feasible and useful functional forms which can be seen as a first important step towards requirements for data quality metrics. They ensure the range [0; 1] for the metric values and address the aggregation of multiple assessments.

Even and Shankaranarayanan (2007) aim at an economically oriented management of data quality. They propose four consistency principles for data quality metrics. *Interpretation consistency* states that the metric values on different data view levels (data values, tuples, relations, and the whole database) must have a consistent semantic interpretation. *Representation consistency* requires that the metric values are interpretable for business users (typically on the range [0; 1] with respect to the utility resulting from the assessed data). *Aggregation consistency* states that the assessment of data quality on a higher data view level has to result from the aggregation of the assessments on the respective lower level. The aggregated result should take

values, which are not higher than the highest or lower than the lowest metric value on the respective lower level. *Impartial-contextual consistency* means that data quality metric values should reflect whether the assessment is context-dependent or context-free.

Heinrich et al. (2007; 2009; 2012) analyze how data quality can be assessed by means of metrics in a goal-oriented and economic manner. To evaluate data quality metrics, they define six requirements. *Normalization* requires that the metric values fall into a bounded range (e.g., [0; 1]). *Interval scale* states that the difference between any two metric values can be determined and is meaningful. *Interpretability* means that the metric values have to be interpretable, while *aggregation* states that it must be possible to aggregate metric values on different data view levels. *Adaptivity* requires that it is possible to adapt the metric to the context of a particular application. *Feasibility* claims that the parameters of a metric have to be determinable and that this determination must not be too cost-intensive. Moreover, this requirement states that it should be possible to calculate the metric values in an automated way.

Mosley et al. (2009) and Loshin (2010) discuss requirements for data quality metrics from a practitioners' point of view. Both contributions comprise the requirements *measurability* and *business relevance* claiming that data quality metrics have to take values in a discrete range and that these values need to be connected to the company's performance. Loshin (2010) adds that it is important to clearly define the metric's goal and to provide a value range and an interpretation of the parts of this range (*clarity of definition*). In addition, Mosley et al. (2009) require *acceptability*, which implies that a metric is assigned a threshold at which the data quality level meets business expectations. If the metric value is below this threshold, it has to be clear who is accountable and in charge to take improvement actions. The corresponding requirements *accountability/stewardship* and *controllability*, however, refer to the integration of a data quality metric within organizations (cf. Category C) and are thus not within the focus of this paper. The same holds for the requirements *representation* and *reportability* as found in both works and also *drill-down capability* by Loshin (2010). Representation claims that the metric values should be associated with a visual representation, *reportability* points out that they should provide enough information to be included in aggregated management reports, and *drill-down capability* states that it should be possible to identify a data quality metric's impact factors within the organization. Finally, *trackability* which requires a metric to be repeatedly applicable at several points of time in an organization (cf. Category B) is also beyond the focus of this paper.

Hüner (2011) proposes a method for the specification of business-oriented data quality metrics to support both the identification of business critical data defects and the repeated assessment of data quality. Based on a survey among experts, he specifies 21 requirements for data quality assessment methods (cf. Appendix B). However, only some of them constitute methodical requirements for data quality metrics and metric values (cf. Category A) and are thus considered further. These are *cost/benefit*, *definition of scale*, *validity range*, *comparability*, and *comprehensibility*. The other requirements refer to Category B (e.g., *repeatability*, *definition of measurement frequency*, *definition of measurement point*, *definition of measurement procedure*) or

Category C (e.g., *responsibility*, *escalation process*, *use in SLAs*) and are not within the focus of this paper.

To sum up, prior works provide valuable contributions by stating a number of possible requirements for data quality metrics and their respective values. While some of them overlap, existing literature is still very fragmented. In addition, many requirements are not clearly defined, which makes their application and verification very difficult. To address these issues, we organize the existing requirements in six groups with each group being characterized by a clear, unique characteristic (cf. Table 1). Note that some of the requirements which leave room for interpretation (cf. brackets in Table 1) are classified in more than one group. Further, some of these existing requirements (e.g., *simple ratio*, *weighted average*) could also be understood as a way to define a data quality metric. In the following, however, they are considered as requirements for data quality metrics. For example, *simple ratio* in Group 1 means that a data quality metric should attain values in [0; 1].

Group	Keyword	Requirements
1	range	*normalization, validity range, clarity of definition (range), simple ratio (bounded in [0; 1]), representation consistency (range), measurability*
2	scale	*interval scale, definition of scale (scale)*
3	interpretation	*interpretability, clarity of definition (interpretation), simple ratio (interpretation), interpretation consistency (interpretation), comparability, comprehensibility, definition of scale (interpretation), representation consistency (interpretation)*
4	context	*weighted average (context), impartial-contextual consistency, adaptivity*
5	aggregation	*aggregation consistency, aggregation, min or max operation, weighted average (aggregation), interpretation consistency (aggregation)*
6	cost	*cost/benefit, feasibility, acceptability, business relevance*

Table 1. Groups of Requirements

Group 1 comprises requirements stating that data quality metrics have to take values within a given range. *Simple ratio* and *representation consistency* aim at metric values in the range [0; 1]. *Measurability* results in a bounded range defined by the lowest and the highest discrete value. Hence, these requirements as well as *clarity of definition* (with respect to the range), *normalization* and *validity range* are assigned to this group. Group 2 contains requirements regarding the scale of measurement of the metric values. Since *definition of scale* may not only concern the interpretation of the metric values but also their scale, this requirement is included as well. Group 3 covers requirements claiming an interpretation of the metric values. Here, *clarity of definition* is interpreted as *interpretability*. In addition, metric values satisfying the *simple ratio* requirement can be interpreted as a percentage, and *interpretation consistency* requires a consistent semantic interpretation of the metric values regardless of the hierarchical

level. While *comparability*, *comprehensibility* and *definition of scale* require some kind of interpretation of the metric values (e.g., as a percentage), *representation consistency* directly implies a clear interpretation with respect to the utility of the data under consideration. The requirements in Group 4 state that data quality metrics should be able to consider adequately the particular context of application, for example by means of weights that decrease or increase the influence of contextual characteristics. Group 5 concerns the (consistent) aggregation of the metric values on different data view levels. *Min or max operation* and *weighted average* specify how this aggregation has to be performed and *interpretation consistency* requires the same interpretation of the metric values on all data view levels. Finally, Group 6 focusses on the application of a data quality metric from a cost-benefit perspective. *Feasibility* is part of this group, because it requires that the costs for determining a metric's parameters are taken into account and that it should be possible to calculate the metric values in a widely automated way – a fact that results in lower application costs. *Business relevance* implies that a metric goes along with some benefit for the company, whereas *acceptability* is part of this group because business expectations are defined considering a cost-benefit perspective.

Table 1 provides an overview of the existing requirements for data quality metrics, which are partly fragmented and vaguely defined. Prior work does in fact lack a methodical framework and does not aim at stating and justifying which requirements for data quality metrics support decision-making under uncertainty and an economically oriented management of data quality. To address this research gap, in the next section we present a decision-oriented framework, enabling us to propose a set of requirements for data quality metrics in Section 4. In addition to that, the decision-oriented framework helps to clearly and unambiguously define the presented requirements as well as to justify them. In this way, it is possible to reason that a data quality metric should satisfy the presented requirements to support both decision-making under uncertainty and an economically oriented management of data quality. Finally, this set of clearly defined requirements combines, concretizes, and enhances the identified groups of existing requirements (cf. Table 1) and thus helps to alleviate the fragmentation within the literature on requirements for data quality metrics.

3 Decision-oriented Framework

The decision-oriented framework for our work is based on the following fields: i) decision-making under uncertainty by considering the influence of assessed data quality metric values and ii) economically oriented management of data quality by considering the costs and benefits of applying data quality metrics.[1]

[1] Note that i) may also be seen as an important means for ii). However, due to the high relevance of i) in the context of data quality metrics, we have decided to distinguish both cases.

The literature on decision-making under uncertainty (and in particular under risk) uses the well-known concept of decision matrices to represent the situation decision makers are facing (Laux, 2007; Nitzsch, 2006; Peterson, 2009). Decision makers can choose among a number of alternatives while the corresponding payoff depends on the state of nature. Each possible state of nature occurs with a certain probability. Hence, in case of a risk-neutral decision maker (if this is not the case, the payoffs need to be determined considering risk adjustments), the one alternative is chosen which results in the highest expected payoff when considering the probability distribution over all possible states of nature. Table 2 illustrates a decision matrix for a simple situation with two alternatives a_i $(i = 1, 2)$, two possible states of nature s_j $(j = 1, 2)$, and the respective payoff p_{ij} for each pair (a_i, s_j). The probabilities of occurrence of the possible states of nature are represented by $w(s_j)$. To select the alternative with the highest expected payoff, the decision maker has to compare the expected payoffs for choosing alternative a_1 (i.e., $p_{11}w(s_1) + p_{12}w(s_2)$) and alternative a_2 (i.e., $p_{21}w(s_1) + p_{22}w(s_2)$). The two-by-two matrix serves for illustration purposes only. Generally, we represent the possible states of nature s_j $(j = 1, \dots, n)$ by the vector $S = (s_1, s_2, \dots, s_n)$, the respective probabilities of occurrence by $w(s_j)$, the alternatives a_i $(i = 1, \dots, m)$ by the vector $A = (a_1, a_2, \dots, a_m)$[2], and the payoffs for alternative a_i by the vector $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$. The *expected* payoff for choosing alternative a_i is denoted by $E(a_i, P_i, S) = \sum_{j=1}^{n} p_{ij}w(s_j)$; the maximum expected payoff is given by $max_{a_i} E(a_i, P_i, S)$. An overview of the notation is provided in Appendix A.

	Probability $w(s_1)$	Probability $w(s_2)$
	State s_1	State s_2
Alternative a_1	Payoff p_{11}	Payoff p_{12}
Alternative a_2	Payoff p_{21}	Payoff p_{22}

Table 2. Decision Matrix

Requirements for data quality metrics must guarantee that i) the metric values can support decision-making under uncertainty. To address i) it is necessary to examine the influence of data quality and thus of the data quality metric values on the components of the decision matrix (i.e., the *probabilities of occurrence*, the *payoffs*, and the *alternatives*). In this respect, literature provides useful insights. Heinrich et al. (2012), for example, propose a metric for the data quality dimension currency (cf. also Heinrich and Klier, 2015). The metric values represent probabilities that the data values under consideration still correspond to their real-world value at the instant of assessing data quality. They apply the metric to determine the *probabilities of occurrence* (represented by the metric values) in a decision situation. The influence of data quality on the *payoffs* is considered, for example, by Ballou et al. (1998), Cappiello and Comuzzi (2009) and Even and Shankaranarayanan (2007). All of them argue that less than perfect data

[2] In case of a continuous decision space, this will be a vector of infinitely many alternatives. If not all alternatives are known, the concept of bounded rationality is applied (Jones 1999; Simon 1956, 1969).

quality (represented by the data quality metric values) may affect and reduce the payoffs. Other works such as Fisher et al. (2003), Heinrich et al. (2007), and Jiang et al. (2007) examine the influence of data quality on the choice of the *alternative*.

More precisely, there are several possible ways to express, quantify and integrate the influence of data quality on decision-making. For instance, Even and Shankaranarayanan (2007) consider the effects of data quality on the payoffs for each record of a dataset. They select a subset of attributes which is relevant in the considered application scenario and set the payoffs for a record to zero if the value of at least one relevant attribute is missing. Moreover, having determined the influence of each data quality dimension, there may be several ways to weight and aggregate these influences (e.g., by calculating the weighted sum across all data quality dimensions; cf. Cappiello and Comuzzi, 2009). Therefore, we do not present an explicit formula or method to quantify the influence of data quality on the decision matrix but, instead, specify this impact more generally as follows: Let DQ represent the data quality metric value and $E(a_i, DQ, P_i, S)$ the *expected* payoff for choosing alternative a_i when considering DQ as well as the payoff vector P_i and the vector of states of nature S. Let further $max_{a_i} E(a_i, DQ, P_i, S)$ be the maximum *expected* payoff when considering data quality. It is obvious and in line with prior works (cf. above) that considering data quality may result in choosing a different optimal alternative as compared to not considering data quality (i.e., $a_1 = argmax_{a_i} E(a_i, DQ, P_i, S)$ and $a_2 = argmax_{a_i} E(a_i, P_i, S)$ with $a_1 \neq a_2$). Hence, it is useful to consider data quality by means of well-founded metrics in decision-making under uncertainty.

When developing requirements for data quality metrics, it is further necessary to take into account the field of ii) economically oriented management of data quality to avoid inefficient or impractical metrics. Existing literature has already addressed the question of whether to apply data quality improvement measures from a cost-benefit perspective (Campanella, 1999; Feigenbaum, 2004; Heinrich et al., 2007; Heinrich et al., 2012). Indeed, applying data quality improvement measures may increase the data quality level and thus bring benefits. At the same time, the associated costs have to be taken into account and the improvement measures should only be applied if the benefits outweigh these costs. In decision-making, the benefits result from being enabled to choose a better alternative (i.e., with an additional expected payoff) due to the improved data quality. The costs include the ones for conducting the improvement measures as well as the ones for assessing data quality by means of data quality metrics. The latter have rarely been considered in the literature, even so they play an important role (Heinrich et al., 2007) and must not be neglected. Indeed, if applying a data quality metric is too resource-intensive, it may not be reasonable to do so from a cost-benefit perspective. Thus, requirements for data quality metrics have to explicitly consider this aspect.

Based on the literature on i) and ii) and the above discussion, Figure 1 presents the decision-oriented framework which is used to justify our requirements (for a similar illustration cf. Heinrich et al., 2007, 2009). Data quality metrics are applied to data views to assess the data quality level (cf. I-III). The assessed data quality level (represented by the metric values) influences i)

decision-making under uncertainty and in particular the chosen alternative, and the expected payoff of the decision maker (cf. IV-VI). Thus, the decision maker may apply improvement measures to increase the data quality level represented by the metric values (cf. IX). However, applying data quality improvement measures creates costs (cf. VII). This also holds for the application of the metric including the determination of its parameters (cf. II). Hence, the optimal data quality level (cf. VIII) has to be determined based on an economical perspective.

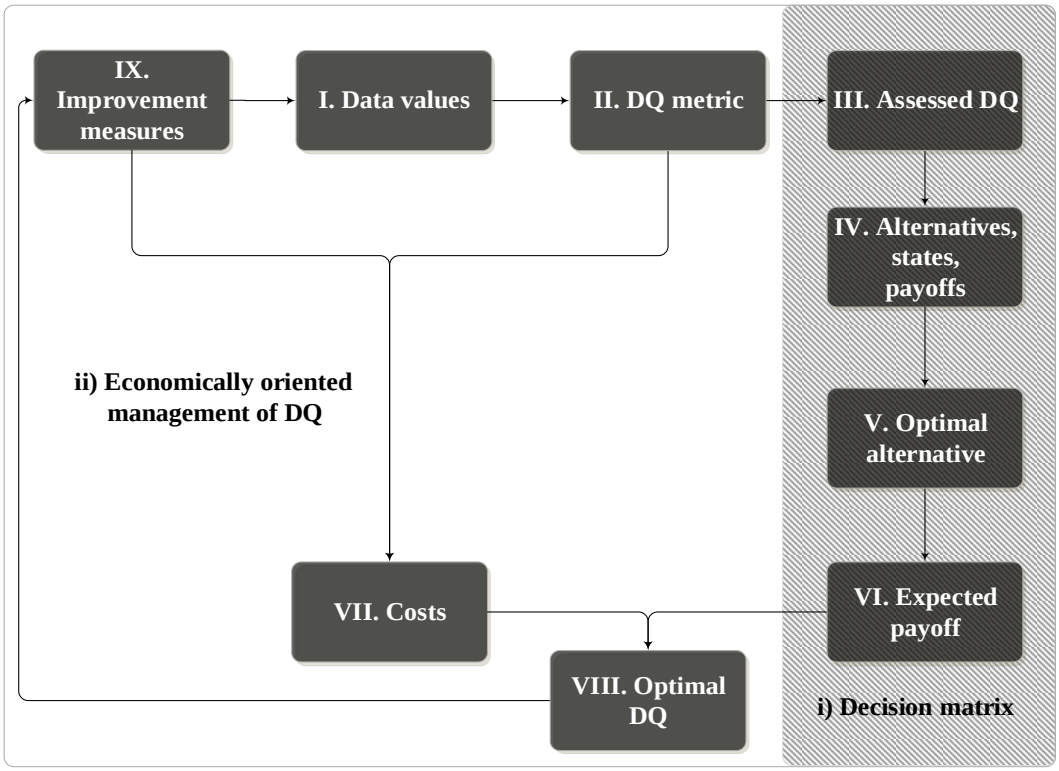

Figure 1. Decision-oriented Framework

4 Requirements for Data Quality Metrics

In this section, we present a set of five clearly defined requirements for data quality metrics. They combine, concretize, and enhance existing approaches covering the six groups of requirements identified in Section 2. Moreover, based on the decision-oriented framework we justify that our requirements support both i) decision-making under uncertainty and ii) an economically oriented management of data quality.

4.1 Requirement 1 (R1): Existence of Minimum and Maximum Metric Values

Group 1 states that data quality metrics have to take values within a given range. Most of the requirements in this group (e.g., *validity range* and *clarity of definition*) are vaguely defined and thus difficult to verify. Hence, both the relevance of these requirements and the possible

consequences of them not being fulfilled remain unclear (e.g., *measurability* just claims that the range should be discrete). To address these issues, we propose and justify the following requirement:

Requirement 1 (R1) (Existence of minimum and maximum metric values). The metric values have to be bounded from below and from above and must be able to attain both a minimum (representing perfectly poor data quality) and a maximum (representing perfectly good data quality). In particular, for each real-world value ω_m, minimum and maximum value have to be attainable in regard to ω_m.

Justification. In a first step, we discuss the following statement (a) which will be used recurrently in the remainder of this justification:

(a) There has to be exactly one metric value representing perfectly good data quality and exactly one metric value representing perfectly poor data quality.

Re (a): Based on the definition of data quality by Orr (1998) used in this paper, perfectly good data quality implies a perfect agreement between stored data views and the real-world. This is a unique situation and therefore there is exactly one level of perfectly good data quality. In the case of the data quality dimension accuracy, existing metrics use a distance function to measure the difference between the real-world data values and the stored data values. Due to the finite number of possibilities for the stored data values (e.g., a 32 bit integer in Java can represent one of 232=4,294,967,296 possible numbers; this holds for other data types used for the assessed data value as well), there is always one or more data value(s) for which the distance to the real-world data value is maximal. For this/these data value/s, the data quality level "perfectly poor data" is reached and cannot become even worse; "even more inaccurate data" cannot be represented. Hence, there is exactly one level of perfectly poor data quality. Summing up and with respect to the discussion of Figure 1, as each data quality level is represented by a metric value and different metric values represent different data quality levels, there has to be exactly one metric value representing perfectly good data quality as well as exactly one metric value representing perfectly poor data quality.

Based on statement (a), we justify (R1). If a data quality metric does not fulfill (R1), this implies that the metric values

(b) are not bounded from below and/or from above and/or

(c) do not attain their minimum and/or maximum.

We denote by ω a stored data value (e.g., a stored customer address) of perfectly good data quality that perfectly represents the corresponding real-world value ω_m. Further, we denote the metric value for ω by $DQ(\omega, \omega_m)$.

Re (b): If there is no upper bound for the metric values, another stored data value ω' can exist which – compared to ω – results in a higher metric value (i.e., $DQ(\omega', \omega_m) > DQ(\omega, \omega_m)$ for the real-world value ω_m corresponding to ω and ω'). As higher metric values represent better

data quality, this implies that ω' is of better data quality than ω. However, ω was defined to be of perfectly good data quality and only one metric value can represent perfectly good data quality (cf. statement (a)). Hence, the metric values indeed need to be bounded from above. The existence of a lower bound can be justified analogously by using a data value of perfectly poor data quality (e.g., the value 'NULL' stored for an unknown customer address which, however, does exist in the real-world).

Re (c): The metric values need to be bounded from below and from above (cf. re (b)). Hence, a supremum M (lowest upper bound) exists. If the metric values do not attain a maximum, it follows that $DQ(\omega, \omega_m) < M$ for a data value ω of perfectly good data quality. As M is the lowest upper bound, there exists another data value ω'' corresponding to the real-world value ω_m with $DQ(\omega, \omega_m) < DQ(\omega'', \omega_m) < M$ (otherwise, $DQ(\omega, \omega_m)$ would be an upper bound and the maximum of the metric values). However, ω was defined to be of perfectly good data quality. Hence, the metric values indeed have to attain a maximum. The existence of a minimum can be justified analogously by using a data value of perfectly poor data quality.

So far, we discussed the existence of a maximum (representing perfectly good data quality) and a minimum (representing perfectly poor data quality) for the metric values with regard to an arbitrary, but fixed real-world value ω_m. However, as there is always exactly one metric value representing perfectly good (resp. poor) data quality (cf. (a)), these maxima and minima coincide across all real-world values. Therefore, the metric values have to be bounded from below and from above and must attain both a minimum and a maximum (cf. I-III in Figure 1), equal for all real-world values.

When a data quality metric is represented by a mathematical function, (R1) means that this function has to be bounded from below and from above and must attain a minimum and maximum. However, some existing metrics (cf., e.g., Alpar and Winkelsträter, 2014; Hinrichs, 2002; Hipp et al., 2001; Hipp et al., 2007) do not attain a minimum or maximum and may thus lead to a wrong evaluation of decision alternatives (cf. III-VI in Figure 1). In these cases it is, for example, not possible to decide whether the assessed data quality level can or should be increased to allow for better decision-making (cf. VI-IX in Figure 1). As a result, for instance, unnecessary improvement measures for data values of already perfectly good data quality may be performed since the metric values cannot represent the fact that perfectly good data quality has already been reached. Moreover, when assessing data quality multiple times with a metric which does not satisfy (R1), neither the comparability nor the validation (e.g., against a benchmark, such as a required completeness level of 90% of the considered database) of the metric values in different assessments are guaranteed. Moreover, when a specific data quality improvement measure is performed, no benchmark in the sense of a minimum and maximum exists to compare the rankings in the course of time (e.g., consider a user survey regarding the existing data quality level without any information in regard to the scale of values to be entered by the users). This contradicts an economically oriented management of data quality.

4.2 Requirement 2 (R2): Interval-Scaled Metric Values

The requirements in Group 2 focus on the scale of measurement of the metric values. These requirements have not been justified, and some of them do not specify a precise scale (e.g., *definition of scale* is not defined, but only illustrated by a very wide range of examples). To address this gap, we state and justify the following requirement:

Requirement 2 (R2) (Interval-scaled metric values). The values of a data quality metric have to be interval-scaled[3]. Based on the classification of scales of measurement (Stevens, 1946), this means that differences and intervals can be determined and are meaningful.

Justification. We argue that a metric which does not provide interval-scaled values (cf. I-III in Figure 1) cannot support both the evaluation of decision alternatives and an economically oriented management of data quality in a well-founded way (cf. Section 3). For this, we take into account the decision matrix in Table 2 with the payoff vectors $P_1 = (p_{11}, p_{12})$ and $P_2 = (p_{21}, p_{22})$ for the alternatives a_1 and a_2 and let the expected payoffs for these alternatives be calculated based on the metric values DQ_1 and DQ_2, respectively. We consider a situation in which the expected payoffs for choosing alternative a_1 and alternative a_2 are the same (i.e., $E(a_1, DQ_1, P_1, S) = E(a_2, DQ_2, P_2, S)$) while $p_{11} > p_{21}$, $p_{12} = p_{22}$, and $DQ_1 < DQ_2$ holds. Hence, the decision maker faces a situation in which in state s_1 choosing alternative a_1 goes along with a higher payoff than choosing a_2 ($p_{11} > p_{21}$), but due to the lower metric value DQ_1 compared to DQ_2, the expected payoff for both alternatives which takes into account the effects of DQ_1 and DQ_2 is the same (cf. III-VI in Figure 1). In this situation, the decision maker is indifferent between the two alternatives[4]. Thus, the lower payoff for a_2 – compared to a_1 – is accepted if its estimation is based on data of higher data quality. This means that the decision maker equally evaluates both a change in payoffs from p_{11} to p_{21} and a change in data quality metric values from DQ_1 to DQ_2. As both the payoffs and expected payoffs are interval-scaled, the differences between payoffs (resp. expected payoffs) are meaningful and their change can be quantified and evaluated by calculating these differences. To support decision-making under uncertainty, this quantified, interval-scaled change in payoffs has to be comparable to a change in data quality. Hence, it has to be possible to calculate the change between the metric values DQ_1 and DQ_2. When the values provided by a metric are not interval-scaled, there is a missing interpretability of the changes between the metric values compared to the respective existing

[3] They may also be ratio-scaled, which is a stronger property and includes interval scaling (Stevens 1946).

[4] If such a situation does not exist, the decision is trivial: If $E(a_1, DQ_1, P_1, S) > E(a_2, DQ_2, P_2, S)$ holds for $p_{11} > p_{21}, p_{12} = p_{22}$ and all possible values for DQ_1 and DQ_2 (i.e., it is not necessarily $DQ_1 < DQ_2$), the decision maker will always choose a_1 regardless of the metric values. In this case, data quality does not matter, which means that assessing data quality is not necessary at all. The same argumentation applies analogously for $E(a_1, DQ_1, P_1, S) < E(a_2, DQ_2, P_2, S)$ where alternative a_2 will always be chosen.

and meaningful differences in the payoffs which impedes the evaluation of decision alternatives. Hence, at most ordinal-scaled data quality metric values cannot support both the evaluation of decision alternatives and an economically oriented management of data quality.

(R2) has a significant practical impact. Indeed, many existing data quality metrics (cf., e.g., Ballou et al., 1998; Hinrichs, 2002), which do not provide interval-scaled values, may lead to wrong decisions when evaluating different decision alternatives (cf. III-VI in Figure 1). Moreover, when evaluating, interpreting and comparing the effects of different data quality improvement measures for an economically oriented management of data quality, interval-scaled metric values are highly relevant. For example, let an ordinal-scaled metric take the values "very good", "good", "medium", "poor" and "very poor". Then there is no possibility of specifying the meaning of the difference between "very good" and "medium" and a decision maker cannot assess whether it would have the same business value as a difference in payoffs of \$500 or \$600. In contrast, this difference in payoffs may be equivalent to a difference of 0.2 in metric values for an interval-scaled metric. In particular, it is not enough to state which measure results in the greatest improvement of the data quality level based on ordinal-scaled metric values. In the example of an ordinal-scaled metric above, it cannot be determined whether an improvement from "very poor" to "medium" is of the same magnitude as an improvement from "medium" to "very good". Similarly, it is unclear whether an improvement from "very poor" to "medium" is twice as much as an improvement from "very poor" to "poor". In contrast, for an interval-scaled metric, an improvement of 0.2 is twice as much as an improvement of 0.1. To ensure the selection of efficient data quality improvement measures, their benefits (i.e., the additional expected payoff) resulting from a clearly specified increase in the data quality level need to be determined precisely and compared to their costs (cf. VI-IX in Figure 1).

The requirements in Group 3 state that the metric values must have an interpretation. However, existing requirements (e.g., *comprehensibility*, *comparability*, *interpretability*, *definition of scale*, *interpretation consistency*, and *clarity of definition*) have neither been justified nor do they specify what exactly is meant by interpretation, making the verification of data quality metrics in this regard very difficult. In the following, we argue that we do not need to define a separate requirement for Group 3, because a clear interpretation is already ensured by the combination of (R1) and (R2). Indeed, a metric which meets both (R1) and (R2) is *interpretable* in terms of the measurement unit *one* (Bureau International des Poids et Mesures, 2006). To justify this, let m be the minimum (representing perfectly poor data quality) and M be the maximum (representing perfectly good data quality) of the metric values (cf. (R1)). Since equal differences result in equidistant numbers on an interval scale (cf. (R2)), each value DQ of the metric can be interpreted as the $\frac{(DQ-m)}{(M-m)}$ fraction of the maximum difference $(M-m)$. Thus, a data quality metric that meets both (R1) and (R2) is inherently interpretable in terms of the measurement unit *one* (i.e., as percentage).

A clear interpretation of the metric values is helpful to understand the actual meaning of the data quality level and is thus important in practical applications, such as the communication to

business users. This is the case if the metric values are ratio-scaled. Ratio-scaled metric values support statements such as "a metric value of 0.6 is twice as high as a metric value of 0.3". Ratio-scale can be achieved by a simple transformation of each interval-scaled data quality metric whose minimum m of the metric values is transformed to 0 so that each metric value can be interpreted as a fraction with respect to the maximum data quality value.

4.3 Requirement 3 (R3): Quality of the Configuration Parameters and the Determination of the Metric Values

Group 4 contains requirements stating that it must be possible to adjust a data quality metric to adequately reflect the particular context of application. This, however, addresses only one relevant aspect. There are well-known scientific quality criteria (i.e., objectivity, reliability, and validity) that must be satisfied by data quality metrics but have not been considered in the literature yet. In addition, not only the metric values, but also the configuration parameters of a data quality metric should satisfy these quality criteria to avoid inadequate results (cf. II-III in Figure 1).[5] To address these drawbacks, we propose and justify the following requirement:

Requirement 3 (R3) (Quality of the configuration parameters and the determination of the metric values). It must be possible to determine the configuration parameters of a data quality metric according to the quality criteria objectivity, reliability, and validity (cf. Allen and Yen, 2002; Cozby and Bates, 2012; Zikmund et al., 2012). The same holds for the determination of the metric values.

There exists a large body of literature dealing with the quality criteria objectivity, reliability, and validity of measurements in general (cf., e.g., Allen and Yen, 2002; Cozby and Bates, 2012; Litwin, 1995; Marsden and Wright, 2010; Zikmund et al., 2012). In the following, we first briefly discuss these criteria for the context of data quality metrics. Afterwards, we justify their relevance based on our decision-oriented framework.

Objectivity of both the configuration parameters and the data quality metric values denotes the degree to which the respective parameters and values as well as the procedures for determining them (e.g., SQL queries) are independent of external influences (e.g., interviewers). This criterion is especially important for data quality metrics requiring expert estimations to determine the configuration parameters or the metric values (cf., e.g., Ballou et al., 1998; Hinrichs, 2002; Cai and Ziad, 2003; Even and Shankaranarayanan, 2007; Hüner et al., 2011; Heinrich and Hristova, 2014). Here, *objectivity* is violated if the estimations are provided by too few experts or if external influences such as the particular behavior of the interviewers are not minimized. In general, *objectivity* becomes an issue if metrics lack a precise specification of (sound) procedures for the determination of the respective parameters and values. In this case, metrics may

[5] Note that in line with our focus on a methodical perspective on requirements for data quality metrics, we concentrate on methodical criteria. Organizational aspects such as the frequency of applying the metric (defined and idiosyncratic per company) are not discussed.

result in different results if applied multiple times. To avoid highly subjective results and ensure *objectivity*, the data quality metric and its configuration parameters have to be unambiguously (e.g., formally) defined and determined with objective procedures (e.g., statistical methods; cf., e.g., Heinrich et al., 2012).

Reliability of measurement refers to the accuracy with which a parameter is determined. *Reliability* conceptualizes the replicability of the results of the methods used for the determination of the configuration parameters or the metric values. In particular, methods will not be reliable, if expert estimations which change over time or among different groups of experts are involved. *Reliability* can be analyzed based on the correlation of the results obtained from the different measurements. Thus, data quality metrics which rely on expert estimations (cf., e.g., Ballou et al., 1998; Even and Shankaranarayanan, 2007; Hüner et al., 2011; Heinrich and Hristova, 2014) have to define a reliable procedure to determine the configuration parameters and the metric values. Generally, to ensure *reliability* of the configuration parameters and the metric values, correct database queries or statistical methods may be used. In this case, the result of the respective procedure remains the same when being applied multiple times to the same data.

Validity is defined as the degree to which a metric "measures what it purports to measure" (Allen and Yen, 2002) or as "the extent to which [a metric] is measuring the theoretical construct of interest" (Marsden and Wright, 2010). Hence, the *validity* of a method for determining the configuration parameters or the metric values refers to the degree of accuracy with which a proposed method actually measures what it should measure.[6] Typically, the *validity* of the determination of a configuration parameter or a metric value is violated if the determination contradicts the aim. There are several examples which illustrate the practical relevance of *validity* in the context of data quality metrics. The metric for timeliness by Batini and Scannapieco (2006, p. 29), for example, involves the configuration parameter $Currency$ which is intended to represent "how promptly data are updated". Its mathematical specification $Currency = Age + (DeliveryTime - InputTime)$, however, seems to contradict this aim. Similarly, Hüner et al. (2011, p. 150) state that a metric value of zero indicates that "each data object validated contains at least one critical defect". However, the mathematical definition of the metric reveals that, to be zero, each data object must actually contain *all* possible critical defects. *Validity* can be achieved by consistent definitions, database queries, or statistical estimations constructed to determine the corresponding parameter or value according to its definition. Additionally, restricting the application domain of a metric (cf., e.g., Ballou et al., 1998; Heinrich et al., 2007) also contributes to *validity*.

Justification. To justify the relevance of (R3) based on the decision-oriented framework, we consider a data quality metric for which *objectivity*, *reliability* and/or *validity* are violated but their values are used to support decision-making under uncertainty (cf. Figure 1). For example,

[6] If validity and reliability are fulfilled for a data quality metric, variations in metric values reflect variations in the level of data quality (i.e., sensitivity is guaranteed; cf., e.g., (Allen and Yen 2002)).

objectivity and/or *reliability* may be violated due to different expert estimations for the configuration parameters of the metric and *validity* may be violated due to an inaccurate definition of the metric or its configuration parameters (cf. above). We analyze a decision situation as illustrated by the decision matrix in Table 2. In case *objectivity* and/or *reliability* are violated, two applications of the data quality metric result in two different data quality levels DQ_1 and DQ_2 with $DQ_1 \neq DQ_2$ (e.g., depending on different expert estimations). In case *validity* is violated, the data quality level DQ_1 estimated by the metric does not accurately represent the actual data quality level DQ_2 in the real-world. In either case, consider that DQ_1 and DQ_2 result in choosing different alternatives. To be more precise, this means $a_1 = argmax_{a_i} E(a_i, DQ_1, P_i, S)$ and $a_2 = argmax_{a_i} E(a_i, DQ_2, P_i, S)$ with $a_1 \neq a_2$[7] (cf. III-VI in Figure 1). If *objectivity* and/or *reliability* are violated, it is not clear to the decision maker whether DQ_1, DQ_2 or none of them correctly reflects the actual data quality level and thus whether a_1 or a_2 is the accurate decision. Similarly, if *validity* is violated, then the decision maker will choose a_1 instead of the accurate choice a_2. Thus, in case *objectivity*, *reliability* and/or *validity* are violated, decision makers will make wrong decisions.

The above justification reveals that data quality metrics which do not fulfill (R3) can lead to wrong decisions when evaluating alternatives (cf. III-VI in Figure 1). In addition, such metrics result in serious problems when evaluating data quality improvement measures (cf. VII-IX in Figure 1). Indeed, assessing data quality before and after a data quality improvement measure with a metric not fulfilling (R3) results in inaccurate metric values. This makes it impossible to determine the increase in the data quality level in a well-founded way (e.g., a data quality improvement measure evaluated as effective before its application may not even lead to an increase in the actual data quality level). To support an economically oriented management of data quality, it is thus important to ensure (R3).

4.4 Requirement 4 (R4): Sound Aggregation of the Metric Values

Group 5 addresses the consistent aggregation of the metric values on different data view levels. Again, the requirements in this group are not motivated based on a sound framework. In addition, applying the *min or max* and the *weighted average operations* – as proposed by existing works – does not necessarily assure a consistent aggregation. We address these issues by the following requirement:

Requirement 4 (R4) (Sound aggregation of the metric values). A data quality metric has to be applicable to single data values as well as to sets of data values (e.g., tuples, relations, and a whole database). Furthermore, it has to be assured that the aggregation of the resulting metric

[7] In case $a_1 = a_2$, data quality does not matter, which means that assessing data quality is not necessary at all (cf. the justification of (R2)).

values is consistent throughout all levels. To be more precise, for data D_{l+1} at a data view level $l+1$ with a disjoint decomposition $D_{l+1} = D_l^1 \cup D_l^2 \cup \ldots \cup D_l^H$ at data view level l (i.e., $D_l^i \cap D_l^j = \emptyset$ for all $i, j \in \{1, \ldots, H\}, i \neq j$), there has to exist an aggregation function f: $DQ(D_{l+1}) = f(DQ(D_l^1), \ldots, DQ(D_l^H))$ with $f(DQ(D_l^1), \ldots, DQ(D_l^H)) = f(DQ(\widetilde{D}_l^1), \ldots, DQ(\widetilde{D}_l^K))$ for all disjoint decompositions D_l^h, $\widetilde{D}_l^k$ of D_{l+1}.

Justification. To justify the relevance of (R4), we argue that a data quality metric needs to

(a) be applicable to different data view levels and

(b) provide a consistent aggregation of metric values

in order to support decision-making under uncertainty and an economically efficient management of data quality.

Re (a): Consider a situation (cf. Figure 1) in which data used for decision-making is not restricted to the level of single data values, but also covers sets of data values (e.g., tuples, relations, and the whole database). This implies that for decision-making under uncertainty and an economically oriented management of data quality, it must be possible to determine data quality at several data view levels.

Re (b): Consider a data quality metric which is defined at both a lower data view level l (e.g., relations) and a higher data view level $l+1$ (e.g., database). In the following, we justify that the metric values must have a consistent aggregation from l to $l+1$. To be more precise, we argue that if an aggregation function f for determining the metric value at level $l+1$ based on the metric values at level l does not assure a consistent aggregation, the metric values cannot support decision-making under uncertainty and an economically oriented management of data quality in a well-founded way (cf. Section 3). In this case, the aggregation of the metric values at l to the metric value at $l+1$ does not adequately reflect the characteristics of the underlying datasets at l (e.g., size, importance). For our argumentation, we consider a disjoint decomposition of a dataset D_{l+1} at $l+1$ into the subsets D_l^h $(h = 1, \ldots, H)$ at l (e.g., a database D_{l+1} which is decomposed into non-overlapping relations D_l^h): $D_{l+1} = D_l^1 \cup D_l^2 \cup \ldots \cup D_l^H$ and $D_l^i \cap D_l^j = \emptyset \; \forall i \neq j$. The metric values for the subsets D_l^h are denoted by $DQ(D_l^h)$. On this basis, the metric value for D_{l+1} can be determined by means of the aggregation function f: $DQ(D_{l+1}) = f(DQ(D_l^1), \ldots, DQ(D_l^H))$. If the aggregation function f does not assure a consistent aggregation of the metric values from l to $l+1$, there exists another decomposition $D_{l+1} = \widetilde{D}_l^1 \cup \widetilde{D}_l^2 \cup \ldots \cup \widetilde{D}_l^K$ of D_{l+1} at l with $DQ'(D_{l+1}) = f(DQ(\widetilde{D}_l^1), \ldots, DQ(\widetilde{D}_l^K))$, $\widetilde{D}_l^i \cap \widetilde{D}_l^j = \emptyset \; \forall i \neq j$ and $DQ'(D_{l+1}) \neq DQ(D_{l+1})$. Following this, the resulting metric value for D_{l+1} depends on the decomposition of the dataset and can hence be manipulated accordingly (i.e., there are two or more possible metric values for the same dataset). Thus, we face the same situation as in the justification of (R3) where it is also not known which metric value actually represents the "real" data quality level of the dataset D_{l+1}. It analogously follows that this situation results in wrong

decisions (cf. III-VI in Figure 1). To sum up, a data quality metric requires a consistent aggregation of the metric values throughout the different data view levels to support decision-making under uncertainty and an economically oriented management of data quality (cf. Section 3).

When data quality metrics are seen as mathematical functions, (R4) means that these functions for the different data view levels have to be compatible with aggregation. Decision situations usually rely on the data quality of (large) sets of data values. However, many data quality metrics in the literature do not provide (consistent) aggregation rules for different data view levels (cf., e.g., Alpar and Winkelsträter, 2014; Hipp et al., 2001; Hipp et al., 2007; Li et al., 2012). As the above justification reveals, this may lead to wrong decisions when evaluating different decision alternatives (cf. III-VI in Figure 1). In addition, a consistent interpretation of the metric values on all aggregation levels is important to support an economically oriented management of data quality. Otherwise, (repeated) measurements of data quality will provide inconsistent and/or wrong results (e.g., when assessing sets of data values that change their volume over time), making it impossible to precisely determine the benefits of improvement measures and to decide whether they should be applied from a cost-benefit perspective (cf. VI-IX in Figure 1).

4.5 Requirement 5 (R5): Economic Efficiency of the Metric

Finally, Group 6 comprises requirements addressing the cost-benefit perspective when applying data quality metrics[8]. Existing requirements in this group are not motivated based on a framework. Moreover, for some of them, their definition, specification, and interpretation remain unclear (e.g., *business relevance* and how to determine the threshold for *acceptability*), making them difficult to verify. We address these issues by proposing and justifying the following requirement:

Requirement 5 (R5) (Economic efficiency of the metric). The configuration and application of a data quality metric have to be efficient from an economic perspective. In particular, the additional expected payoff from the intended application of a metric has to outweigh the expected costs for determining both the configuration parameters and the metric values.

Justification. To justify (R5), we analyze a decision situation as shown in the decision matrix in Table 2. Let alternative a_1 be chosen by a decision maker who does not consider data quality in decision-making (and thus does not apply the metric). Furthermore, let another alternative a_2 be chosen if data quality is considered. To be more precise, it holds $a_1 = argmax_{a_i} E(a_i, P_i, S)$ and $a_2 = argmax_{a_i} E(a_i, DQ, P_i, S)$ with $a_1 \neq a_2$.[9] In this situation, from

[8] We consider a decision scenario (and the related expected payoffs and costs) in which a data quality metric supports an economically oriented management of data quality from a methodical perspective. We do not focus on organizational aspects such as the conduction of a decision-making process in organizations.

[9] In case $a_1 = a_2$, data quality does not matter, which means that assessing data quality is not necessary at all (cf. the justification of (R2)).

a decision-making perspective, considering data quality represents *additional information* influencing the evaluation of the decision alternatives and their choice. This means that the existing data quality level is an additional information affecting the (ex post) realized payoffs. Thus, the benefit of this additional information is assessed by the difference between the expected payoffs (cf. III-VI in Figure 1) when choosing a_1 resp. a_2 both under consideration of the additional information, which means, $E(a_2, DQ, P_2, S) - E(a_1, DQ, P_1, S)$ (for details cf. Heinrich and Hristova, 2016). Thereby, the application of the data quality metric is economically efficient and therefore justifiable with respect to the decision-oriented framework (cf. Figure 1) if and only if the difference between the expected payoffs outweighs the expected costs for applying the data quality metric. Otherwise, the metric contradicts an economically oriented management of data quality.

Regarding (R5), especially metrics requiring configuration parameters not directly available to the user have to be analyzed in detail. For example, the metric for correctness by Hinrichs (2002) involves determining real-world values as input, which is usually very resource-intensive and raises the question why a subsequent data quality assessment is even necessary (for a detailed discussion cf. Section 5.4). In case of metrics not fulfilling (R5), the determination of the configuration parameters or the procedure for determining the metric values is expected to be too costly compared to the estimated additional expected payoffs (cf. I-IX in Figure 1). In some cases, it may be possible to use automated approximations and estimations (especially for configuration parameters) to reduce the effort. Metrics not fulfilling (R5) can still be valuable from a theoretical perspective, but they are not of practical relevance. (R5) is of particular importance in data governance and data quality management. Indeed, metrics not fulfilling (R5) are usually not suitable for use in a data governance initiative for data quality assessment, as the valuation and success of actions (such as applying a data quality metric) taken in such initiatives is ultimately to be determined by economic efficiency (Sarsfield, 2009).

5 Application of the Requirements

We demonstrate the applicability and efficacy of our requirements by evaluating five metrics from literature (Alpar and Winkelsträter, 2014; Ballou et al., 1998; Blake and Mangiameli, 2011; Hinrichs, 2002; Yang et al., 2013). We chose these metrics covering timeliness, completeness, reliability, correctness and consistency to provide a broad perspective on different dimensions of data quality and to show that the presented requirements can indeed be applied to various dimensions for data views and data values stored in an information system. To make the evaluation of the metrics more transparent and comprehensible, we refer to the following context of application (cf. Even et al., 2010; Heinrich and Klier, 2015): Based on the stored data of existing customers (e.g., corporate customers), a company has to decide which customers to contact with a new product offer in a CRM mailing campaign. The two decision alternatives for the company with respect to each customer in the database are a_1: to select the customer for the campaign or a_2: not to do so. The possible states of nature (occurring depending

on a certain probability of acceptance) are s_1: the customer accepts or s_2: the customer rejects the offer. The benefits of applying a data quality metric in this context are generally non-negligible. Indeed, considering the quality of the customer data (as discussed by (Even et al., 2010) and (Heinrich and Klier, 2015)) will lead to better decisions (e.g., if an offer is sent to an outdated or incomplete address, this will only cause mailing costs).

5.1 Metric for Timeliness by Ballou et al. (1998)

The data quality metric for timeliness proposed by Ballou et al. (1998) is defined as follows:

$$Timeliness := \max\left[1 - \frac{age\ of\ the\ data\ value}{shelf\ life}, 0\right]^s \quad (1)$$

The parameter *age of the data value* represents the time difference between the occurrence of the real-world event (i.e., when the data value was created in the real-world) and the assessment of timeliness of the data value. The parameter *shelf life* is defined as the maximum length of time the values of the considered attribute remain up-to-date. Thus, a higher value of the parameter *shelf life*, ceteris paribus, implies a higher value of the metric for timeliness, and vice versa. The exponent $s > 0$, which has to be determined based on expert estimations, influences the sensitivity of the metric to the ratio $\frac{age\ of\ the\ data\ value}{shelf\ life}$. In Table 3, we present the evaluation of the metric based on the requirements.

R1: Existence of minimum and maximum metric values	**(Fulfilled)**
For all values of the parameter $s > 0$, the metric values are within the bounded interval [0; 1]. The minimum of zero (which represents perfectly poor data quality) is attained if the parameter *age of the data value* is greater than or equal to the parameter *shelf life*. The maximum of one (which represents perfectly good data quality) is attained if the parameter *age of the data value* equals zero (e.g., a stored customer address is certainly up-to-date). It follows that (R1) is fulfilled.	
R2: Interval-scaled metric values	**(Not fulfilled)**
For $s = 1$ the metric values can be interpreted as the percentage of the data value's remaining shelf life (e.g., a stored customer address is up-to-date with 50%). As a consequence, for $s = 1$ we observe a ratio scale which implies that the values are interval-scaled as well. Apart from this particular case (i.e., for $s \neq 1$), however, the metric values are not interval-scaled. This is due to the fact that for any two interval scales it is always possible to transform one of them to the other by applying a positive linear transformation of the form $x \mapsto ax + b$ (with $a > 0$) (Allen and Yen, 2002). Obviously, such a transformation does not exist for $s \neq 1$, as the mapping $x \mapsto x^s$ is not linear for $s \neq 1$. That is why the metric values are generally not interval-scaled and (R2) is not fulfilled. To conclude: The parameter s allows to control the sensitivity of the metric values with respect to the ratio of *age of the data value* and *shelf life*, which may be advantageous in specific contexts. To obtain interval-scaled metric values, however, the value $s = 1$ has to be chosen.	

R3: Quality of the configuration parameters and the determination of the metric values	**(Not fulfilled)**
In the context of corporate customer data, the values of the attribute "address" do not have a known and fixed maximum shelf life. Indeed, company addresses are not characterized by a maximum length of time during which they remain up-to-date (e.g., some companies have been located at the same address for hundreds of years). In this case, it is not possible to determine a fixed value for the configuration parameter *shelf life* of the metric. As a result, (R3) is not fulfilled.	
R4: Sound aggregation of the metric values	**(Fulfilled)**
The authors propose to use the weighted arithmetic mean to aggregate the metric values from single data values to a set of data values. (R4) is fulfilled, as this aggregation rule ensures a consistent aggregation of the metric values on all levels. This allows to use the results from an application of the metric for a broad variety of decisions. For example, in the context of customer data, the metric values can be used for the selection of individual customers for the mailing campaign (i.e., a decision on the level of tuples). However, the metric values could – after aggregation – also be used for the decision whether to perform a data quality improvement measure for a larger portfolio of customers.	
R5: Economic efficiency of the metric	**(Not fulfilled)**
Ballou et al. (1998) define the parameter *age of the data value* based on the point of time when the data value was created in the real-world. Therefore, to determine the parameter *age of the data value* for the given context of a customer's address, it has to be known when the customer moved to this address. This point of time, however, is usually neither stored nor easily accessible for companies (e.g., due to privacy protection laws) making the expected costs of configuration parameter determination very high. Indeed, for the above context of a customer database it would not be efficient to determine the configuration parameter *age of the data value* to be able to calculate the metric values for the company's customers. Actually, it would even be easier and less resource-intensive – independent of the benefits of the campaign – to directly evaluate whether the data values are still up-to-date (e.g., by contacting the customers). Therefore, (R5) is not fulfilled in our considered application context.	

Table 3. Evaluation of the Metric by Ballou et al. (1998)

Overall, while the metric for timeliness proposed by Ballou et al. (1998) fulfills (R1) and (R4), it does not fulfill (R2), (R3), and (R5).

5.2 Metric for Completeness by Blake and Mangiameli (2011)

The metric for completeness by Blake and Mangiameli (2011) is defined as follows. On the level of data values, a data value is incomplete (i.e., the metric value is zero) if and only if it is 'NULL', otherwise it is complete (i.e., the metric value is one). Here, all data values which represent missing or unknown values in a specific application scenario (e.g., blank spaces or '9/9/9999' as a date value) are represented by the data value 'NULL'. A tuple in a relation is

defined as complete if and only if all data values are complete (i.e., none of its data values is 'NULL'). For a relation R, let T_R be the number of tuples in R which have at least one 'NULL'-value and let N_R be the total number of tuples in R. Then, the completeness of R is defined as follows:

$$Completeness := 1 - \frac{T_R}{N_R} = \frac{N_R - T_R}{N_R} \quad (2)$$

The evaluation of the metric with respect to the requirements is presented in Table 4:

R1: Existence of minimum and maximum metric values	**(Fulfilled)**
The metric values are within the bounded interval [0; 1]. This holds for all aggregation levels. The minimum of zero (which represents perfectly poor data quality) on the level of data values, tuples, and relations is attained, if a data value equals 'NULL' (e.g., the street of a single customer address is not stored), if a tuple contains at least one data value which equals 'NULL', and if each tuple of a relation contains at least one data value which equals 'NULL', respectively. The maximum of one (which represents perfectly good data quality) on the level of data values, tuples, and relations is attained if a data value does not equal 'NULL', if a tuple does not contain any data value which equals 'NULL', and if a relation does not contain any tuple with data values which equal 'NULL', respectively. It directly follows that (R1) is fulfilled.	
R2: Interval-scaled metric values	**(Fulfilled)**
On the levels of data values and tuples, the metric values are interval-scaled (i.e., the difference between the only two possible metric values zero and one is meaningful). On the level of relations, the metric values are defined as the percentage of tuples which do not contain any data value which equals 'NULL' (e.g., 50% of all tuples storing customer data are complete). That implies a ratio scale, and thus the values are also interval-scaled. Therefore, (R2) is fulfilled. Based on the metric values' interpretation, the impact of a data quality improvement measure can thus be assessed precisely. For instance, a change in metric values from 0.4 to 0.7 means that instead of 40%, now 70% of all tuples are complete, which may be important for an appropriate selection of customers.	
R3: Quality of the configuration parameters and the determination of the metric values	**(Fulfilled)**
All configuration parameters of the metric (i.e., whether a data value equals 'NULL'; whether a tuple contains a data value, which equals 'NULL'; and the number of tuples in a relation and how many of them contain at least one data value, which equals 'NULL') can be determined by means of simple database queries. Hence, the quality criteria objectivity, reliability, and validity are fulfilled. The metric values can be determined by means of mathematical formulae in an objective and reliable way. As the metric quantifies the data quality dimension completeness at different levels according to the corresponding definition, the determination of the metric values is valid. To sum up, (R3) is fulfilled.	

R4: Sound aggregation of the metric values	(Fulfilled)
The metric is applicable to single data values as well as to sets of data values (tuples and relations). The determination of the metric values on the different aggregation levels follows well-defined rules allowing for a consistent aggregation. Therefore, (R4) is fulfilled.	
R5: Economic efficiency of the metric	**(Fulfilled)**
The parameters of the metric can be determined by means of database queries and the metric values can be determined by means of mathematical formulae, both of them in an automated and effective way and at negligible costs. In case the benefits from applying the metric are non-negligible (cf. given context of application), the application of the metric is efficient and thus fulfills (R5). For instance, in the application context of the CRM mailing campaign, the costs for applying the metric will easily be made up for by saving costs for sending mailings in case of incomplete customer records.	

Table 4. Evaluation of the Metric by Blake and Mangiameli (2011)

Overall, the metric by Blake and Mangiameli (2011) satisfies all requirements (R1) to (R5).

5.3 Metric for Reliability by Yang et al. (2013)

The data quality metric for reliability proposed by Yang et al. (2013) is defined based on the answers to n equally important[10] questions referring to the reliability of a given dataset (e.g., a database). In particular, the answer to question i is represented by the triangular fuzzy number $Q_i = (a_{1i}, a_{2i}, a_{3i})$, where $a_{1i} = s_i c_i$, $a_{2i} = s_i$ and $a_{3i} = s_i c_i + 1 - c_i$ with $s_i \in [0;\ 1]$ being the satisfaction degree of question i and $c_i \in [0;\ 1]$ the corresponding certainty degree. The reliability of a dataset is defined by the total score:

$$Reliability := \sum_{i=1}^{n} Q_i \tag{3}$$

This reliability is then matched to one of three fuzzy sets, representing different levels of reliability. In order to evaluate this metric with regard to our requirements, we consider the approach proposed by the authors in a decision support context (such as the aforementioned CRM mailing campaign) to defuzzify the total score in (3). We apply the centroid method as the most common defuzzification approach (Driankov et al., 1996). On this basis, given a triangular fuzzy number $Q_i = (a_{1i}, a_{2i}, a_{3i})$, the defuzzification operator is

$$C: (a_{1i}, a_{2i}, a_{3i}) \mapsto \frac{a_{1i} + a_{2i} + a_{3i}}{3} \tag{4}$$

and the defuzzified reliability of a dataset is defined by:

[10] In the application of Yang et al. (2013), $n = 21$ is used. The authors also discuss the use of so-called "red criteria", which always need to be fulfilled. As their use is not decisive for the evaluation of the proposed metric, we do not further consider them here.

$$\sum_{i=1}^{n} C(Q_i) \tag{5}$$

In Table 5, we present the evaluation of the metric based on the requirements.

R1: Existence of minimum and maximum metric values	**(Fulfilled)**
The maximum reliability is achieved if all n questions are assigned both a satisfaction degree and certainty degree of one (e.g., all experts are certain that customer information is fully reliable). In this case, the defuzzified score in (5) is n. The minimum reliability is achieved if all n questions are assigned a satisfaction degree of zero and a certainty degree of one (e.g., all experts are certain that customer information is not reliable at all). In this case, the defuzzified score in (5) is 0. Thus, (R1) is fulfilled.	
R2: Interval-scaled metric values	**(Fulfilled)**
Consider two different reliability scores generated on two different datasets: $$Score_1 = \left(q_1^{(1)}, q_2^{(1)}, q_3^{(1)}\right) \quad (6)$$ $$Score_2 = \left(q_1^{(2)}, q_2^{(2)}, q_3^{(2)}\right) \quad (7)$$ where $q_k^{(j)} = \sum_{i=1}^{n} a_{ki}^{(j)}, j \in \{1,2\}, k \in \{1,2,3\}$ and $(a_{1i}^{(j)}, a_{2i}^{(j)}, a_{3i}^{(j)})$ as defined above. Then, the defuzzified values of $Score_1$ and $Score_2$ are: $$C(Score_1) = \frac{q_1^{(1)}+q_2^{(1)}+q_3^{(1)}}{3} \quad (8)$$ $$C(Score_2) = \frac{q_1^{(2)}+q_2^{(2)}+q_3^{(2)}}{3} \quad (9)$$ As a result: $$C(Score_1) - C(Score_2) = \frac{\left(q_1^{(1)}-q_1^{(2)}\right)+\left(q_2^{(1)}-q_2^{(2)}\right)+\left(q_3^{(1)}-q_3^{(2)}\right)}{3}$$ $$= \sum_{i=1}^{n} \frac{\left(a_{1i}^{(1)}-a_{1i}^{(2)}\right)+\left(a_{2i}^{(1)}-a_{2i}^{(2)}\right)+\left(a_{3i}^{(1)}-a_{3i}^{(2)}\right)}{3}$$ $$= \sum_{i=1}^{n} \frac{\left(s_i^{(1)}c_i^{(1)}-s_i^{(2)}c_i^{(2)}\right)+\left(s_i^{(1)}-s_i^{(2)}\right)+\left(s_i^{(1)}c_i^{(1)}+1-c_i^{(1)}-s_i^{(2)}c_i^{(2)}-1+c_i^{(2)}\right)}{3}$$ $$= \sum_{i=1}^{n} \frac{2\left(s_i^{(1)}c_i^{(1)}-s_i^{(2)}c_i^{(2)}\right)+\left(s_i^{(1)}-s_i^{(2)}\right)+\left(c_i^{(2)}-c_i^{(1)}\right)}{3}$$ $$= \sum_{i=1}^{n} \frac{\left(2s_i^{(1)}c_i^{(1)}+s_i^{(1)}-c_i^{(1)}\right)-\left(2s_i^{(2)}c_i^{(2)}+s_i^{(2)}-c_i^{(2)}\right)}{3}$$ Thus, the difference between two defuzzified reliability scores is always the sum of the differences between the defuzzified answers to each question, regardless of the particular values of $Score_1$ and $Score_2$. As a result, the metric values are interval-scaled.	

R3: Quality of the configuration parameters and the determination of the metric values	**(Fulfilled)**
The input parameters are the answers to the n questions by experts in the corresponding area. In our CRM mailing campaign scenario, these questions would aim at evaluating the reliability of the customer data with regard to the criteria that are relevant for the campaign (e.g., address, ability-to-pay, willingness-to-pay). Thus, to achieve input parameters of high quality, the answers to these questions need to be gathered by following the standard approaches for questionnaire development and application (Litwin, 1995; Marsden and Wright, 2010). Since the remainder of the metric application can be carried out in a formal, automated way, the metric fulfills (R3). This fact is critical to guarantee that the metric values can be used for decision-making, for instance in the CRM mailing campaign scenario.	
R4: Sound aggregation of the metric values	**(Not fulfilled)**
Yang et al. (2013) do not discuss the application of their metric on different data view levels. Therefore, no aggregation rule is provided. In particular, there is no information regarding the treatment of the n questions in a situation in which the reliability of multiple datasets is assessed (e.g., a possible adaptation of the questions or best practices for consulting experts). In the CRM scenario, this implies that it is not possible to consistently determine the reliability of different databases, for instance, by different external data providers. In that sense, (R4) is not fulfilled.	
R5: Economic efficiency of the metric	**(Fulfilled)**
The application of the metric requires the answers to each of the n questions by experts as well as the automated determination based on term (3) and the application of a defuzzification operator (4). The last two metric calculations can be done by means of mathematical formulae, both of them in an automated and effective way and at low costs. Moreover, both the survey and calculations are carried out once for the whole dataset and not for each single data value and are also independent of the size of the dataset. Given that in the context of our CRM mailing campaign, the benefits are expected to be significant (Even et al., 2010; Heinrich and Klier, 2011), the application of the metric is economically efficient.	

Table 5. Evaluation of the Metric by Yang et al. (2013)

Overall, the metric by Yang et al. (2013) satisfies requirements (R1) to (R3) and (R5), but does not address (R4).

5.4 Metric for Correctness by Hinrichs (2002)

The data quality metric for correctness proposed by Hinrichs (2002) is, on the level of data values, defined as follows:

$$Correctness := \frac{1}{d(\omega, \omega_m) + 1} \tag{10}$$

Here, ω is the data value to be assessed, ω_m is the corresponding real-world value and d is a domain-specific distance measure such as, for example, the Euclidean distance or the Hamming distance. A larger difference between ω and ω_m is represented by a larger value of the distance function, which in turn leads to a larger denominator and thus a smaller metric value. The evaluation of the metric with respect to the proposed requirements is presented in Table 6.

R1: Existence of minimum and maximum metric values	**(Not fulfilled)**
If ω perfectly represents the corresponding real-world value ω_m, the distance $d(\omega, \omega_m)$ is determined to be equal to 0 and the metric attains its maximum value of 1. In general, however, the metric values are dependent on the chosen distance function d (which may, for example, be the edit distance, the Euclidean distance or the Hamming distance). This distance function d necessarily varies from dataset to dataset and even between the assessed data values in a particular dataset, as specific distance functions can only be applied to specific data types (e.g., the Euclidean distance function may only be used for numerical data values). Thus, $d(\omega, \omega_m)$ may – dependent on the distance function – become arbitrarily large. Following this, the resulting metric values can indeed be very small while never reaching 0 (as this would require an infinite distance), leading to a violation of (R1). To conclude, the metric does not attain a fixed minimum metric value and (R1) is not fulfilled.	
R2: Interval-scaled metric values	**(Not fulfilled)**
Common distance measures such as the edit distance, the Euclidean distance or the Hamming distance yield interval-scaled distance values. However, the quotient in the calculation formula inhibits the interval scaling of the resulting metric values: For example, to improve the value of correctness from $\frac{1}{6}$ to $\frac{1}{4}$ (i.e., by $\frac{1}{12}$) , the value of the corresponding distance function has to be decreased from 5 to 3. To improve the value of correctness from $\frac{1}{4}$ further to $\frac{1}{3}$ (i.e., again by $\frac{1}{12}$), only a reduction in distance from 3 to 2 is needed. Thus, the differences of the metric values are in general not meaningful and the metric values are not interval-scaled. Hence, (R2) is not fulfilled.	
R3: Quality of the configuration parameters and the determination of the metric values	**(Fulfilled)**
The metric requires the real-world value corresponding to the data value to be assessed. Determining the real-world value may be resource-intensive in most cases (cf. evaluation of (R5)), but the determination is objective and reliable (as there is exactly one real-world value), and, as long as a well-founded way to determine the value is chosen, valid. For example, in the CRM mailing campaign context, data from external sources (e.g., registration offices or companies such as the German Postal Service, which offer address data) could be used, providing an accurate real-world value. No further configuration parameters are needed, and thus, objectivity, reliability and validity are not violated in this regard. The mathematical formula for calculating the metric values allows for an objective and reliable determination. Finally, the determination of the metric values is valid, because the metric	

quantifies the data quality dimension correctness according to its definition. Summing up, (R3) is fulfilled.	
R4: Sound aggregation of the metric values	**(Not fulfilled)**
To determine the metric value at the database level based on its values at the level of relations, Hinrichs (2002) suggests the unweighted arithmetic mean denoted by f in the following. Consider a database D_{l+1} which is decomposed into disjoint relations D_l^h: $D_{l+1} = D_l^1 \cup D_l^2 \cup \ldots \cup D_l^H$ with $D_l^i \cap D_l^j = \emptyset \; \forall i \neq j$ and let further, without loss of generality, the subset D_l^1 be divided into two disjoint subsets $D_l^{1\prime}$ and $D_l^{1\prime\prime}$ at l (i.e., $D_l^1 = D_l^{1\prime} \cup D_l^{1\prime\prime}$, $D_l^{1\prime} \cap D_l^{1\prime\prime} = \emptyset$). Then, let $DQ(D_{l+1}) = f(DQ(D_l^1), \ldots, DQ(D_l^H))$ and $DQ'(D_{l+1}) = f(DQ(D_l^{1'}), DQ(D_l^{1''}), DQ(D_l^2), \ldots, DQ(D_l^H))$. Because f is the unweighted arithmetic mean and the same subsets of D_{l+1} are weighted relatively with $1/H$ or $1/(H+1)$ depending on the particular decomposition used, the equation $DQ'(D_{l+1}) = DQ(D_{l+1})$ does in general not hold, which contradicts a consistent aggregation and thus (R4).	
R5: Economic efficiency of the metric	**(Not fulfilled)**
The metric is based on the comparison of the stored data value and the corresponding real-world value. In many cases, determining the real-world value as input for a data quality metric is (very) resource-intensive as for a large number of data values a real-world comparison is required. For example, in the CRM mailing campaign context, buying external data for a large customer base is (very) expensive and other methods (e.g., trying to contact all customers by phone) similarly require a very high effort. Moreover and in contradiction to an efficient application of the metric, in case the real-world value is known, simply updating the stored data value with the corresponding real-world value would result in perfectly good data quality and the calculation of the metric value would no longer be needed (as this metric value has to represent perfectly good data quality). For example, when the real address of a customer is known anyway, applying the metric to measure the correctness of a possibly wrong address provides no additional benefit. Thus, as the metric requires the corresponding real-world values for all stored data values as input, it is not economically efficient and (R5) is not fulfilled.	

Table 6. Evaluation of the Metric by Hinrichs (2002)

Overall, the metric by Hinrichs (2002) satisfies (R3), but does not satisfy (R1), (R2), (R4) and (R5).

5.5 Metric for Consistency by Alpar and Winkelsträter (2014)

Alpar and Winkelsträter (2014) define a metric for the consistency of a tuple t as

$$Consistency(t) \coloneqq \sum_{r \in R} \begin{cases} w^+(r), & if\ t\ fulfills\ r \\ w^-(r), & if\ t\ violates\ r \\ w^0(r), & if\ r\ does\ not\ apply, \end{cases} \tag{11}$$

where R is a set of association rules (Agrawal et al., 1993), $w^+(r)$ and $w^-(r)$ denote the scoring for a fulfilled and violated association rule, respectively, and $w^0(r)$ is the scoring for an inapplicable association rule (which is proposed to be equal to 0). Generally, fulfilled association rules contribute to a higher total score while violated rules lead to a decrease in total score, and tuples with a higher score are assessed as being more consistent. In Table 7, we present the evaluation of the metric based on the requirements.

R1: Existence of minimum and maximum metric values	**(Not fulfilled)**
The metric values depend strongly on the rule set R and the parameters $w^+(r)$ and $w^-(r)$. The larger the rule set R and the lower the respective weights $w^-(r)$, the lower the metric values for tuples violating many rules are. In contrast, the larger the rule set R and the larger the respective weights $w^+(r)$, the larger the metric values for tuples fulfilling many rules are. The rule set R and the weights $w^+(r)$ and $w^-(r)$ necessarily vary from dataset to dataset. As a result, the metric values are neither bounded from below nor from above. Thus, neither a minimum metric value nor a maximum metric value exists and (R1) is not fulfilled.	
R2: Interval-scaled metric values	**(Not fulfilled)**
The parameters $w^+(r)$, $w^-(r)$ and $w^0(r)$ can be set such that the metric value can be interpreted as the percentage of the association rules fulfilled by the tuple. In this case, the metric values are ratio-scaled and hence also interval-scaled. However, the parameters can also represent a non-linear transformation of this setting (e.g., the parameters are a quadratic function), which in turn leads to non-interval-scaled metric values. This is due to the fact that for any two interval scales it is always possible to transform one of them to the other by applying a positive linear transformation of the form $x \mapsto ax + b$ (with $a > 0$) (Allen and Yen, 2002). To conclude, (R2) is not fulfilled. As a consequence, the metric values may lead to wrong evaluations of different decision alternatives. For instance, the difference in consistency of two stored customer addresses is not meaningful and thus cannot be used to determine which customer to select for a CRM campaign.	
R3: Quality of the configuration parameters and the determination of the metric values	**(Fulfilled)**
Association rule mining algorithms (e.g., Agrawal and Srikant, 1994) can be used to determine the rule set R in a reliable and objective way. Further, in their application of the metric, the authors propose to use $w^+(r) = confidence(r)^\tau$, $w^-(r) = -confidence(r)^\tau$ and $w^0(r) = 0$, where $confidence(r)$ represents the confidence of an association rule and $\tau \in \mathbb{N}$ is a calibration parameter. The confidence of an association rule can be calculated reliably and objectively based on simple database queries (e.g., applied to the stored customer data used in the CRM mailing campaign). For τ, the authors suggest a value larger than 25, which is to be verified by experiments. By use of such experiments, τ can then also be determined reliably and objectively. Based upon this, the metric values themselves can be calculated. As the proposed parameters and also the metric itself additionally measure what they should measure and are thus valid, (R3) is fulfilled.	

R4: Sound aggregation of the metric values	(Not fulfilled)
Alpar and Winkelsträter (2014) do not provide a definition or an aggregation function to allow the assessment of consistency by means of their metric on a level other than the level of tuples. Hence, it is unclear how to apply the metric and assess consistency on an aggregated level. It follows that (R4) is not fulfilled. Thus, when metric values on an aggregated level are required for decision-making, the metric cannot provide guidance. For instance, the metric cannot be used to assess the consistency of a whole customer database in order to decide whether to perform a data quality improvement measure addressing the database level.	
R5: Economic efficiency of the metric	**(Fulfilled)**
Using $w^+(r) = confidence(r)^\tau$, $w^-(r) = -confidence(r)^\tau$ and $w^0(r)$, as described in the evaluation of (R3) and suggested for a concrete application, means that the expected costs for applying the metric are low: The rule set R can be determined by a common association rule mining algorithm while the parameters of the metric can be calculated by means of database queries. Similarly, the metric values can be calculated without much effort; all these steps can be performed in an automated and effective way at (rather) low expected costs. The value of the parameter τ needs to be verified by experiments, but this can be done efficiently by preparing a small test set and performing automated tests. In our application context of a CRM mailing campaign, in which significant benefits are to be expected (Even et al., 2010; Heinrich and Klier, 2011), the application of the metric is efficient and thus fulfills (R5).	

Table 7. Evaluation of the Metric by Alpar and Winkelsträter (2014)

Overall, while the metric by Alpar and Winkelsträter (2014) fulfills requirements (R3) and (R5), it does not fulfill (R1), (R2), and (R4).

To sum up, the evaluation of the five data quality metrics shows that our requirements are neither trivial nor impossible to fulfill.

6 Practical Implications

In this section, we discuss the relevance and priority of the requirements with a focus on their practical implications. We provide a combined analysis for (R1) and (R2) as well as separate discussions for (R3), (R4) and (R5). Table 8 summarizes the findings.

R1: Existence of minimum and maximum metric values
R2: Interval-scaled metric values

(R1) and (R2) are of particularly high relevance if, based on the metric values, a decision about different data quality improvement measures or, more generally, about decision alternatives *by means of economic criteria* (cf. economically oriented management of data quality) is made. More precisely: Let us suppose that in a particular application the aim is to just measure the currency of two data values of an attribute and to judge whether the first data value is more up-

to-date than the second one (i.e., to make a true/false statement). In this special case, a simple ranking of the metric values for currency of the two data values would be sufficient. Here, one is not interested in the extent of the difference between the metric values for currency of the two data values, nor does one need to know whether the interpretation of one or both metric values for currency suggests (highly) up-to-date or outdated data values.

However, for the large majority of practical applications such a (simple) ranking in the sense of a true/false statement is not sufficient. Rather, based on the metric values, a decision about different decision alternatives *assessed by means of economic criteria* needs to be made. If, in such a case, only a ranking is available, the validation against a specified benchmark (e.g., a required completeness level of 90% of the considered database) is not possible, impeding the use of the metric for decision-making. Furthermore, a ranking cannot support the decision whether the assessed data quality level should be increased based on economic criteria (resp. whether it is even possible to do so). Additionally, when using such a metric, the effects of a data quality improvement measure cannot be clearly compared to its costs. All of these aspects are crucial for an economically oriented management of data quality.

To sum up: A metric might be designed specifically for the context of analyzing the rankings of existing data quality levels or used exclusively in such a context. If this is not the case, but rather a decision about different decision alternatives *assessed by means of economic criteria* (e.g., a comparison of alternative data quality improvement measures) is made based on the metric values, then requirements (R1) and (R2) are highly relevant.

R3: Quality of the configuration parameters and the determination of the metric values

(R3) aims to guarantee that independently of the measuring subjects, one measures what one strives to measure and does so in a correct way. Thus, this requirement covering *validity*, *reliability* and *objectivity* is generally of high importance which can be illustrated by the example of assessing the data quality dimension currency. In practical applications, *internal validity* is of particular relevance. Internal validity first addresses that the underlying definition of currency ("object of interest") is indeed measured by the metric. Second, it also ensures that significant changes in the metric values (i.e., the dependent variable) are indeed caused by a change in the variables which influence currency and not by extraneous factors (control variables). In contrast, *external validity* is primarily only of high relevance if the metric is just applied to a sample of the dataset, but the results are used to derive statements regarding the whole dataset. *Reliability* aims to guarantee that the metric leads to equal or very similar results (i.e., a high stability of the results) in repeated assessments of the same data (e.g., in the course of time) and to thus ensure a correct measurement in this regard. *Objectivity* is particularly relevant to allow both an automated data quality assessment and obtaining metric values which are independent of external influences (e.g., different interviewers).

Overall, data quality metrics not fulfilling (R3) can provide insufficient metric values (cf. above). Regarding an economically oriented management of data quality, this is, for instance, problematic when evaluating the data quality level before and after conducting a data quality

improvement measure. A metric not fulfilling (R3) cannot deliver trustworthy results with respect to the actual change in the data quality level. Thus, a data quality improvement measure may be evaluated as effective but may not actually improve data quality at all or only by a very small margin.

To sum up, when designing and applying a metric, the following points need to be considered:

(a) It is important to analyze which data values, metadata and parameter values are required to instantiate and apply a data quality metric: If extensive historical data (either from internal or external sources; big / open data) is available, the required data values and parameters (in particular, the configuration parameters) can be determined in a valid, objective and reliable way using statistical techniques. If such a data basis is not available, for instance expert estimations are needed, which also have to be obtained in a transparent and verifiable way.

(b) Where possible, metrics should be formally defined such that – as long as the required data values and parameters are clearly defined – the calculation rule ensures (R3) (in particular, objectivity and reliability). If the calculation rule cannot be formally defined, the calculation of the metric values needs to be described in a stepwise, transparent way and as clear as possible to allow an intersubjective application. In any case, the correspondence between what is to be measured (in particular, an exact definition of the respective data quality dimension) and what is actually measured (operationalization of the defined data quality dimension) needs to be ensured.

R4: Sound aggregation of the metric values

(R4) is of high relevance if the assessment or the selection of decision alternatives is not just based on the isolated data quality assessment of a single data value. More precisely: Let us consider an application in which the aim is just to measure the completeness of the data values of an attribute, independently from each other. Let further the individual metric values be directly used for decision-making, for example such that in case no data value (or a data value semantically equivalent to 'NULL') is stored, apply action *a*; otherwise do not apply any action. In this case, an isolated decision based on the level of data values is performed, which does not require any aggregation. However, decisions in practice, for example regarding the application of data quality improvement measures, are usually not just based on a single data value or individual data values considered in an isolated way. Rather, this requirement is of particular practical relevance in many decision situations that rely on the data quality of (large) sets of data values. For example, the data quality of a larger part of a customer database (or even the whole database) may be considered to decide whether to conduct a marketing campaign.

To sum up: If a metric was not explicitly designed for statements regarding the data quality of single data values (resp. it is not only used in such situations), but rather is designed or used to express the data quality of multiple data values in a single metric value, (R4) is particularly relevant. The higher the importance of this aggregated metric value for decision support, the higher the relevance of (R4).

R5: Economic efficiency of the metric

(R5) is of particularly high priority if assessing data quality by means of a metric results in substantial costs resp. the metric values are used for a decision with potentially large costs and benefits. Especially against the background that in practice low data quality often results in high costs (Experian Information Solutions, 2016; IBM Big Data and Analytics Hub, 2016; Rogers et al., 2017), this requirement needs to be taken into account already in the design of a metric. More precisely: Let us consider an application in which the aim is just to measure the completeness of the data values of a single attribute in a relation with around 100 tuples. The assessment is conducted manually by a single person within a time span of five minutes (i.e., the costs for determining both the configuration parameters and the metric values are negligible). This person stores the result of the assessment (i.e., the percentage of complete data values according to the metric) just for documentation purposes in a file, no further analysis is conducted and no decisions at all are based on the result of the assessment (i.e., any additional payoff from the application of the metric is irrelevant). In such cases, in practice, evaluating the efficiency of the metric – in particular in comparison to alternative metrics which might possibly allow a slightly faster counting – is hardly necessary. Similarly, one might argue that evaluating the efficiency of metrics is not required in the application case of a data quality assessment mandatory due to legal regulations (e.g., in risk management). Here, one could reason analogously that the evaluation of the efficiency is not relevant for the decision whether to apply a metric. However, this argument may fall short: Even in the case of a mandatory assessment, a company may again evaluate the economic efficiency of two or more possible metrics to select the most appropriate one. Thus, in many cases, (R5) is highly relevant from a practical perspective. Moreover, (R5) is also of particular importance when assessing data quality as part of a data governance or data quality management initiative, as these are generally aimed at economic efficiency.

To sum up: Data quality metrics are usually not designed for assessing data quality in cases of low economic relevance of the assessment (when both the additional payoffs as well as the costs resulting from an application of the metric are negligible). Thus, the relevance of (R5) is obvious. This relevance increases the higher the expected costs resp. the expected benefits from both measuring data quality and the decisions based on the assessment are. Table 8 summarizes the findings with regard to decision situations for which specific requirements are of particular relevance.

	(R1) and (R2)	(R3)	(R4)	(R5)
Of particular relevance in practical situations in which ...	- ... decision alternatives are assessed by means of economic criteria. - ... multiple, related data quality assessments are performed, for instance over time. - ... a particular focus resides on the interpretability of the metric values. - … improvement measures and their impacts on data quality are compared or evaluated.	- ... both configuration parameters and metric values are not absolutely trivial to determine (in contrast to situations where, for instance the configuration parameters are given or can be obtained effortlessly). - ... multiple, related data quality assessments are performed, for instance over time.	- ... metric values on different aggregation levels are relevant for decision-making. - ... the decision relies on the data quality of (large) sets of data values. - ... multiple, aggregations are performed and the results are compared. - ... the aggregation of metric values is necessary for the determination of one metric value on an aggregated level.	- ... potentially large costs and benefits emerge. - ... an efficient metric has to be selected amongst different feasible metrics. - ... multiple, related data quality assessments are performed, for instance over time.

Table 8. Practical Situations with particular Relevance for specific Requirements

7 Conclusions, Limitations and Future Research

In this paper, we propose a set of five requirements for data quality metrics to support both decision-making under uncertainty and an economically oriented management of data quality. Our requirements contribute to existing literature in two ways. First, as opposed to existing approaches, which are fragmented and leave room for interpretation, we present a set of clearly defined requirements, thus making it possible to easily and transparently verify them. This is very important for practical applications. Second, in contrast to existing works, we justify our requirements based on a sound decision-oriented framework. If such a framework is missing, it is neither possible to substantiate the relevance of the requirements nor is it clear what happens if a requirement is not met. As a result, our requirements are essential for the evaluation of existing metrics as well as for the design of new metrics (e.g., in the context of design science research). Based on our requirements, inadequate metrics, which may lead to wrong decisions

and economic losses, can be identified and improved. The applicability and efficacy of the proposed requirements are demonstrated by means of five well-known data quality metrics. The application to the metric for completeness by Blake and Mangiameli (2011) reveals the existence of metrics which satisfy all requirements. The application to the metrics by Ballou et al. (1998), Yang et al. (2013), Hinrichs (2002) and Alpar and Winkelsträter (2014), however, shows that the requirements are not trivial to fulfill. Both results are crucial from a methodical and practical point of view.

The proposed requirements constitute a first but essential step to support both decision-making under uncertainty and an economically oriented management of data quality. Nevertheless, they also have limitations. First, they are designed for data quality metrics concerning data views and therefore do, for instance, not directly consider data quality metrics addressing the quality of data schemes. However, in future research, the ideas underlying the derivation of the requirements can be transferred analogously to other types of data quality metrics. Moreover, as already discussed for many other sets of requirements (e.g., in the context of software engineering), it is not possible to prove the completeness and sufficiency of a set of requirements. Indeed, extending a set of requirements is an iterative process, which should consider both theoretical and practical aspects. Thus, future research should extend the proposed set of requirements in a well-founded manner.

8 References

Agrawal, R., T. Imieliński and A. Swami (1993). "Mining association rules between sets of items in large databases". In: *ACM SIGMOD Record*, pp. 207–216.

Agrawal, R. and R. Srikant (1994). "Fast Algorithms for Mining Association Rules". In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994)*, pp. 487–499.

Allen, M. and D. Cervo (2015). *Multi-Domain Master Data Management. Advanced MDM and Data Governance in Practice:* Morgan Kaufmann.

Allen, M. J. and W. M. Yen (2002). *Introduction to measurement theory:* Waveland Press.

Alpar, P. and S. Winkelsträter (2014). "Assessment of data quality in accounting data with association rules" *Expert Systems with Applications* 41 (5), 2259–2268.

Azuma, M. (2001). "SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality". In: *European Software Control and Metrics Conference (ESCOM)*, pp. 337–346.

Ballou, D., R. Wang, H. Pazer and G. K. Tayi (1998). "Modeling information manufacturing systems to determine information product quality" *Management Science* 44 (4), 462–484.

Batini, C. and M. Scannapieco (2006). *Data quality: concepts, methodologies and techniques:* Springer.

Batini, C. and M. Scannapieco (2016). "Data Quality Dimensions". In *Data and Information Quality*, pp. 21–51: Springer.

Blake, R. and P. Mangiameli (2011). "The effects and interactions of data quality and problem complexity on classification" *Journal of Data and Information Quality (JDIQ)* 2 (2), 8.

Briand, L. C., S. Morasca and V. R. Basili (1996). "Property-based software engineering measurement" *IEEE Transactions on Software Engineering* 22 (1), 68–86.

Buhl, H. U., M. Röglinger, F. Moser and J. Heidemann (2013). "Big data. A fashionable topic with(out) sustainable relevance for research and practice?" *Business & Information Systems Engineering (BISE)* 5 (2), 65–69.

Bureau International des Poids et Mesures (2006). *The international system of units (SI):* National Institute of Standards and Technology.

Cai, L. and Y. Zhu (2015). "The challenges of data quality and data quality assessment in the big data era" *Data Science Journal* 14.

Cai, Y. and M. Ziad (2003). "Evaluating completeness of an information product". In: *Proceedings of the 9th Americas Conference on Information Systems (AMCIS 2003)*, pp. 2273–2281.

Campanella, J. (1999). *Principles of Quality Costs: Principles, Implementation and Use.:* ASQ Quality Press.

Cappiello, C. and M. Comuzzi (2009). "A utility-based model to define the optimal data quality level in IT service offerings". In *Proceedings of the 17th European Conference on Information Systems (ECIS 2009).*

Cappiello, C., T. Di Noia, B. A. Marcu and M. Matera (2016). "A quality model for linked data exploration". In: *International Conference on Web Engineering (ICWE)*, pp. 397–404.

Cozby, P. and S. Bates (2012). *Methods in behavioral research.* 11th Edition: McGraw-Hill Higher Education.

Debattista, J., S. Auer and C. Lange (2016). "Luzzu—A Methodology and Framework for Linked Data Quality Assessment" *Journal of Data and Information Quality (JDIQ)* 8 (1), 4.

Driankov, D., H. Hellendoorn and M. Reinfrank (1996). *An Introduction to Fuzzy Control.* Second, revised edition: Springer.

Eppler, M. J. (2003). *Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes:* Springer.

Even, A. and G. Shankaranarayanan (2007). "Utility-driven assessment of data quality" *ACM SIGMIS Database* 38 (2), 75–93.

Even, A., G. Shankaranarayanan and P. D. Berger (2010). "Evaluating a model for cost-effective data quality management in a real-world CRM setting" *Decision Support Systems (DSS)* 50 (1), 152–163.

Experian Information Solutions (2016). *Building a business case for data quality.* Experian Information Solutions.

Fan, W. (2015). "Data Quality. From Theory to Practice" *ACM SIGMOD Record* 44 (3), 7–18.

Feigenbaum, A. V. (2004). *Total quality control:* McGraw-Hill Professional New York.

Fisher, C. W., I. Chengalur-Smith and D. P. Ballou (2003). "The impact of experience and time on the use of data quality information in decision making" *Information Systems Research (ISR)* 14 (2), 170–188.

Fisher, C. W., E. J. M. Lauria and C. C. Matheus (2009). "An accuracy metric: Percentages, randomness, and probabilities" *Journal of Data and Information Quality (JDIQ)* 1 (3), 16.

Flood, M., H. V. Jagadish and L. Raschid (2016). "Big data challenges and opportunities in financial stability monitoring" *Financial Stability Review* 20, 129–142.

Heinrich, B. and D. Hristova (2014). "A Fuzzy Metric for Currency in the Context of Big Data". In *Proceedings of the 22nd European Conference on Information Systems (ECIS 2014)*.

Heinrich, B. and D. Hristova (2016). "A quantitative approach for modelling the influence of currency of information on decision-making under uncertainty" *Journal of Decision Systems (JDS)* 25 (1), 16–41.

Heinrich, B., M. Kaiser and M. Klier (2007). "How to measure data quality? A metric-based approach". In *Proceedings of the 28th International Conference on Information Systems (ICIS 2007)*.

Heinrich, B. and M. Klier (2011). "Assessing data currency-a probabilistic approach" *Journal of Information Science* 37 (1), 86–100.

Heinrich, B. and M. Klier (2015). "Metric-based data quality assessment—Developing and evaluating a probability-based currency metric" *Decision Support Systems (DSS)* 72, 82–96.

Heinrich, B., M. Klier and Q. Görz (2012). "Ein metrikbasierter Ansatz zur Messung der Aktualität von Daten in Informationssystemen" *Zeitschrift für Betriebswirtschaft (ZfB)* 82 (11), 1193–1228.

Heinrich, B., M. Klier and M. Kaiser (2009). "A procedure to develop metrics for currency and its application in CRM" *Journal of Data and Information Quality (JDIQ)* 1 (1), 1.

Hinrichs, H. (2002). "Datenqualitätsmanagement in Data-Warehouse-Systemen". Dissertation. University of Oldenburg.

Hipp, J., U. Güntzer and U. Grimmer (2001). "Data Quality Mining-Making a Virtue of Necessity". In *Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2001)*, pp. 52–57.

Hipp, J., M. Müller, J. Hohendorff and F. Naumann (2007). "Rule-Based Measurement Of Data Quality In Nominal Data". In *12th International Conference on Information Quality (ICIQ 2007)*, pp. 364–378.

Hüner, K. M. (2011). *Führungssysteme und ausgewählte Maßnahmen zur Steuerung von Konzerndatenqualität*. Dissertation: University of St. Gallen.

Hüner, K. M., A. Schierning, B. Otto and H. Österle (2011). "Product data quality in supply chains: the case of Beiersdorf" *Electronic Markets (EM)* 21 (2), 141–154.

IBM Big Data and Analytics Hub (2016). *Extracting business value from the 4 V's of big data*. URL: http://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data (visited on 07/19/2017).

IBM Global Business Services (2012). *Analytics: Big Data in der Praxis*. IBM Global Business Services.

ISO/IEC 25020 (2007). *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Measurement reference model and guide* 35.080.

Jiang, Z., S. Sarkar, P. De and D. Dey (2007). "A framework for reconciling attribute values from multiple data sources" *Management Science* 53 (12), 1946–1963.

Jones, B. D. (1999). "Bounded rationality" *Annual Review of Political Science* 2 (1), 297–321.

Khatri, V. and C. V. Brown (2010). "Designing data governance" *Communications of the ACM* 53 (1), 148–152.

KPMG (2016). *Now or never - 2016 Global CEO Outlook.* KPMG.

Laux, H. (2007). *Entscheidungstheorie.* 7th Edition: Springer Gabler.

Lee, Y. W., D. M. Strong, B. K. Kahn and R. Y. Wang (2002). "AIMQ: a methodology for information quality assessment" *Information & Management* 40 (2), 133–146.

Levy, Y. and T. J. Ellis (2006). "A systems approach to conduct an effective literature review in support of information systems research" *Imforming Science* 9 (1), 181–212.

Li, F., S. Nastic and S. Dustdar (2012). "Data Quality Observation in Pervasive Environments". In: *Proceedings of the IEEE 15th International Conference on Computational Science and Engineering (CSE 2012)*, pp. 602–609.

Litwin, M. S. (ed.) (1995). *How to Measure Survey Reliability and Validity:* Sage.

Loshin, D. (2010). *The practitioner's guide to data quality improvement:* Morgan Kaufmann.

Lukoianova, T. and V. L. Rubin (2014). "Veracity roadmap: Is big data objective, truthful and credible?". In: *Proceedings of the 24th ASIS SIG/CR Classification Research Workshop*.

Marsden, P. V. and J. D. Wright (eds.) (2010). *Handbook of survey research.* 2. ed.: Emerald.

Moore, S. (2018). *How to Create a Business Case for Data Quality Improvement*. URL: http://www.gartner.com/smarterwithgartner/how-to-create-a-business-case-for-data-quality-improvement/ (visited on 03/25/2019).

Mosley, M., M. Brackett and S. Earley (eds.) (2009). *The DAMA guide to the data management body of knowledge enterprise server version:* Technics Publications, LLC.

Nitzsch, R. von (2006). *Entscheidungslehre:* Verlag Mainz.

Orr, K. (1998). "Data quality and systems theory" *Communications of the ACM* 41 (2), 66–71.

Otto, B. (2011). "Data Governance" *Business & Information Systems Engineering (BISE)* 3 (4), 241–244.

Parssian, A., S. Sarkar and V. S. Jacob (2004). "Assessing data quality for information products: impact of selection, projection, and Cartesian product" *Management Science* 50 (7), 967–982.

Peterson, M. (2009). *An introduction to decision theory:* Cambridge University Press.

Pipino, L. L., Y. W. Lee and R. Y. Wang (2002). "Data quality assessment" *Communications of the ACM* 45 (4), 211–218.

Redman, T. C. (1996). *Data quality for the information age:* Artech House.

Rogers, B., E. Maguire and A. Nishi (2017). *The Data Differentiator. How Improving Data Quality Improves Business.* Forbes Insights.

Sarsfield, S. (2009). *The data governance imperative:* IT Governance Publishing.

SAS Institute (2013). *2013 Big data survey research brief:* SAS Institute.

Simon, H. A. (1956). "Rational choice and the structure of the environment" *Psychological Review* 63 (2), 129–138.

Simon, H. A. (1969). *The sciences of the artificial:* MIT Press.

Stevens, S. S. (1946). "On the theory of scales of measurement" *Science* 103 (2684), 677–680.

Taleb, I., H. T. El Kassabi, M. A. Serhani, R. Dssouli and C. Bouhaddioui (2016). "Big Data Quality. A Quality Dimensions Evaluation". In: *International IEEE Conference on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 759–765.

Wang, R. Y. (1998). "A product perspective on total data quality management" *Communications of the ACM* 41 (2), 58–65.

Wang, R. Y., V. C. Storey and C. P. Firth (1995). "A framework for analysis of data quality research" *IEEE Transactions on Knowledge and Data Engineering* 7 (4), 623–640.

Weber, K., B. Otto and H. Österle (2009). "One size does not fit all--a contingency approach to data governance" *Journal of Data and Information Quality (JDIQ)* 1 (1), 4.

Webster, J. and R. T. Watson (2002). "Analyzing the past to prepare for the future: Writing a literature review" *MIS Quarterly* 26 (2), 13–23.

Wechsler, A. and A. Even (2012). "Using a Markov-Chain Model for Assessing Accuracy Degradation and Developing Data Maintenance Policies". In: *Proceedings of the 18th Americas Conference on Information Systems (AMCIS 2012).*

Yang, L., D. Neagu, M. T. D. Cronin, M. Hewitt, S. J. Enoch, J. C. Madden and K. Przybylak (2013). "Towards a Fuzzy Expert System on Toxicological Data Quality Assessment" *Molecular Informatics* 32 (1), 65–78.

Zikmund, W., B. Babin, J. Carr and M. Griffin (2012). *Business research methods.* 8th Edition: Cengage Learning.

Appendix

A. Notation

Notation	Definition
s_j	State of nature, $j \in \{1, \dots, n\}$
$S = (s_1, s_2, \dots, s_n)$	Vector of all considered states of nature
$w(s_j)$	Probability of occurrence for a state of nature s_j
a_i	Decision alternative, $i \in \{1, \dots, m\}$
$A = (a_1, a_2, \dots, a_m)$	Vector of all considered decision alternatives
p_{ij}	Payoff if alternative a_i is chosen and state of nature s_j occurs
$P_i = (p_{i1}, p_{i2}, \dots, p_{in})$	Vector of the payoffs for alternative a_i and all considered states of nature
$E(a_i, P_i, S)$	Expected payoff without considering data quality for alternative a_i, given a vector S of states of nature and a vector P_i of payoffs for alternative a_i
DQ $DQ(\dots)$	Data quality metric value of the considered data value or set of data values
$E(a_i, DQ, P_i, S)$,	Expected payoff when considering data quality for alternative a_i, given a vector S of states of nature, a vector P_i of payoffs for alternative a_i, and a value of the data quality metric DQ
M	Supremum/maximum of the considered metric values
l	Data view level with $l \in \{1, \dots, L\}$
D_l	A dataset at data view level l
D_l^h	A subset of the dataset D_l, $h \in \{1, \dots, H\}$
$\widetilde{D}_l^k$	A subset of the dataset D_l, $k \in \{1, \dots, K\}$
f	Aggregation function

Table 9. Notation

B. Requirements for Data Quality Metrics proposed by Hüner et al. (2011)

Requirement	Description of the proposed Requirement
Cost/benefit	The costs for the definition and the calculation of the data quality metric values ought to be in a positive ratio (< 1) to the benefits (controlled error potential).
Definition of measurement frequency	The instants of time at which the values of a data quality metric are calculated should be defined.
Definition of measurement point	The measurement point (e.g., data repository, process, department) of a data quality metric should be defined.
Definition of measurement procedure	The instrument (e.g., survey, software) to determine the data quality metric value should be defined.
Definition of scale	A scale (e.g., percentage, school grades, time) should be defined for a data quality metric value.
Limitation of the application data	For a data quality metric, the data to be applied to (e.g., material master, European customers) should be defined.
Escalation process	For a data quality metric appropriate measures should be defined depending on certain threshold values (i.e., metric values to initiate data quality measures).
Validity range	A range should be defined for a data quality metric in which its values are valid.
SMART criteria	A data quality metric should fulfill the SMART criteria (specific, measurable, attainable, relevant and time-bounded).
Disturbance variables	The metadata of a data quality metric should contain information about possible disturbance variables (i.e., it should describe possible events or impacts which may distort the values of the data quality metric).
Responsibility	For a data quality metric clear responsibilities should be defined such as to whom and which values of the data quality metric are reported, who is responsible for the maintenance of the metric (e.g., up-to-date/meaningful definition, implementation of the measurement procedure).
Comparability	A data quality metric should be defined so that its values can be compared to those of other metrics (data quality metrics or process metrics).
Comprehensibility	For a data quality metric metadata should be available, which describes its purpose and the correct interpretation of its values.
Use in SLAs	It should be possible to use data quality metric values in Service Level Agreements.

Visualization	It should be possible to visualize the values of a data quality metric (e.g., time series, diagrams).
Repeatability	It should be possible to determine the values of a data quality metric not only once, but multiple times.
Target value	For a data quality metric a target value should be defined.
Assignment to a data quality dimension	It should be possible to assign a data quality metric to one or more data quality dimensions.
Assignment to a business problem	It should be possible to assign a data quality metric to a specific (company-specific) business problem.
Assignment to a process figure	It should be possible to assign a data quality metric to one or more process figures.
Assignment to the company strategy	It should be possible to assign a data quality metric to one or more strategic goals of the company.

Table 10. Requirements for Data Quality Metrics proposed by Hüner et al. (2011)

3 Analysis of Textual Data

This section comprises two papers addressing the second focal point of the dissertation, the analysis of textual data. In particular, the research questions RQ4 and RQ5 are discussed. Section 3.1 covers a topic modeling procedure for the discovery of knowledge from CVs (RQ4). In Section 3.2, a model for explaining and interpreting the overall star ratings of online customer reviews employing aspect-based sentiment analysis is proposed (RQ5).

3.1 Paper 4: Knowledge Discovery from CVs – A Topic Modeling Procedure

Current Status	Full Citation
accepted and published in the 2019 Proceedings of the Internationale Tagung Wirtschaftsinformatik (02/2019)	Schiller, A. (2019). "Knowledge Discovery from CVs – A Topic Modeling Procedure". In: *Proceedings of the 14th Internationale Tagung Wirtschaftsinformatik (WI)*, February 23-27, Siegen, Germany.

Summary

This paper addresses RQ4 by proposing a novel topic modeling procedure for the discovery of knowledge from CVs. The procedure is adapted from a process suggested in literature for topic modeling in general information systems and consists of five steps. In each step, the special characteristics of CVs are considered in order to discover interpretable topics describing fine-grained competences. This information can be used to, for instance, rapidly assess the contents of a CV, categorize CVs and identify candidates for job offers. The practical applicability and feasibility of the procedure is evaluated in an exemplary application to real-world CVs from IT experts, where the procedure is able to discover clearly interpretable topics representing specific competences (e.g., Java programming, web design). Furthermore, a topic-based search technique is developed and assessed to produce superior results compared with a keyword search.

The work builds on a multitude of existing concepts and methods from natural language processing and other (AI) fields. For text pre-processing, amongst others, part-of-speech tagging, named entity recognition and lemmatization are used. Latent Dirichlet allocation, a probabilistic machine learning-based natural language processing method, is employed to conduct the actual topic modeling. Further concepts such as Kullback-Leibler divergence (from probability theory) and semantic coherence are also utilized for the work. The proposed procedure allows, for instance, for proactive recruiting when it is applied in human resource management similar to how professional social networks are currently used in the recruitment process to rapidly source candidates before subsequent steps such as job interviews are conducted. Additionally, categorizing, tagging and searching CVs as facilitated by the procedure provides further decision support in human resource management processes.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style.

The paper as published by AIS is available at:
https://aisel.aisnet.org/wi2019/track05/papers/8/

Abstract:

With a huge number of CVs available online, recruiting via the web has become an integral part of human resource management for companies. Automated text mining methods can be used to analyze large databases containing CVs. We present a topic modeling procedure consisting of five steps with the aim of identifying competences in CVs in an automated manner. Both the procedure and its exemplary application to CVs from IT experts are described in detail. The specific characteristics of CVs are considered in each step for optimal results. The exemplary application suggests that clearly interpretable topics describing fine-grained competences (e.g., Java programming, web design) can be discovered. This information can be used to rapidly assess the contents of a CV, categorize CVs and identify candidates for job offers. Furthermore, a topic-based search technique is evaluated to provide helpful decision support.

Keywords: text mining, topic modeling, latent Dirichlet allocation, human resource management

1 Introduction

Acquiring the right personnel is one of the most critical success factors for companies (Breaugh, 2008; Hendry, 2012). In this area, recruiting via the web has gained significant importance over the last years (Allden and Harris, 2013). It is not only of practical interest, but has also received much scientific attention (cf., e.g., Abel et al., 2017; Gao and Eldin, 2014). Opportunities are, for example, provided by well-known professional online social networks such as *LinkedIn* and *XING*, which are becoming highly popular. For instance, as of Q2 2018, *LinkedIn* has more than 562 million members (LinkedIn, 2018). Recruiting via the web is further made possible by CVs provided not only on these networks, but also on private homepages and websites specializing in making available a wide range of CVs (e.g., *Indeed*, *CareerBuilder*, *Monster*). Overall, a huge number of CVs can be acquired online. Based on these CVs, companies have the prospect to identify promising job candidates and to conduct proactive recruiting.

However, to capitalize on this potential, a very large amount of semi- and unstructured data needs to be analyzed. While approaches for a manual analysis of document collections exist (Coffey and Atkinson, 1996), this task becomes too time-consuming for large collections of complex documents (such as CVs) (Debortoli et al., 2016). This issue is addressed by automated text mining methods, which have already been used successfully for human resource management (HRM) (Gupta and Lehal, 2009; Strohmeier and Piazza, 2013). In particular, topic modeling approaches such as latent Dirichlet allocation (LDA) are promising in this application context. They are able to discover the hidden thematic structure present in a document collection (Blei, 2012). Thus, they should be able to extract key information from CVs. More precisely, high-quality, fine-grained topics in a specialized topic model should represent skills, abilities, knowledge and work expertise (in the following subsumed by "competences" as in

(Gorbacheva et al., 2016)). This information can then be used to, for instance, rapidly assess the contents of a CV, categorize CVs and identify candidates for job offers. Topic models offer unique advantages compared to a keyword search on existing platforms. However, research has not yet discussed the application of topic modeling to CVs, leaving open crucial issues (cf. Section 2.2). This paper thus focuses on the following research question:

How can topic modeling be used to discover knowledge from CVs?

The remainder of the paper is structured as follows. In the next section, we discuss the problem context as well as related work and the research gap. In Section 3, we propose a procedure for knowledge discovery from CVs. Section 4 contains an application of the procedure and an evaluation of the results. Finally, we provide conclusions, limitations and directions for future research.

2 Background

In this section, we first briefly introduce topic modeling, LDA and evaluation methods for topic models. Then, we give an overview of related literature and discuss the research gap.

2.1 Problem Context

Topic modeling approaches aim to discover the latent thematic structure in a document collection and to identify thematically similar documents (Blei, 2012). A topic model consists of a number of topics, each represented by terms strongly associated to the topic. Recently, probabilistic approaches have been highly popular. Here, topics can be seen as probability distributions over terms and documents as probability distributions over topics. In this paper, we focus on LDA (Blei et al., 2003), a probabilistic approach not relying on any kind of training data. It is the most widely-applied topic modeling approach (Belford et al., 2018). Distributions are calculated using sampling or optimization procedures which take into account term-document-frequencies (Hoffman et al., 2010). LDA is based on the bag-of-words model (i.e., the order of words in documents is ignored). This makes it particularly suitable for CVs, which often are formulated in note form instead of continuous text.

While an evaluation of topic models by humans is considered the gold standard (Chang et al., 2009), the required time effort has sparked the need for automated evaluations, especially for testing a large number of pre-processing and parameter configurations. Many criteria and methods have been proposed, discussing, for instance, the similarity (Koltcov et al., 2014), stability (Belford et al., 2018) or semantic coherence (Aletras and Stevenson, 2013; Lau et al., 2014; Newman et al., 2010) of topics. While a negative correlation to human interpretability has been reported for some methods (Chang et al., 2009), semantic coherence has been shown to provide assessments of high quality (Debortoli et al., 2016; Lau et al., 2014; Newman et al., 2010; Röder et al., 2015). It is often calculated based on normalized pointwise mutual information (NPMI) (Lau et al., 2014; Röder et al., 2015). The idea is that a topic is of higher quality when the terms

strongly associated to the topic often occur in close proximity in a text corpus. Following the discussion above, measuring semantic coherence by NPMI is used in this paper for assessing various topic model configurations before the final topic model is humanly interpreted.

2.2 Related Work and Research Gap

Topic modeling and in particular LDA is widely applicable to a large range of contexts and has been successfully employed to, for instance, consumer good reviews (Debortoli et al., 2016), research articles (Fang et al., 2018), hotel critiques (Guo et al., 2017) and even in BPM (Dumont et al., 2016). A tutorial for applying LDA in IS in general has been proposed as well (Debortoli et al., 2016). However, the high degree of abstraction and flexibility also come at a price: An adaption to the application context is necessary to provide proper results. Despite much existing work to use text mining in HRM (Gupta and Lehal, 2009; Strohmeier and Piazza, 2013), a literature search revealed that little of it has addressed topic modeling.

A notable exception is a work suggesting topic modeling for job offers (Gao and Eldin, 2014). The objective was to identify competences of importance in the construction industry. This research is similar to ours in the sense that a topic modeling approach was used for this task. However, the aim differs, because instead of CVs, job offers have been analyzed. Furthermore, no procedure for knowledge discovery is described.

In further research, *LinkedIn* profiles of BPM professionals are examined via topic modeling to investigate the role of gender in BPM (Gorbacheva et al., 2016). While here, a topic modeling approach is applied to documents which are similar to CVs, the aim of the research is completely different to ours. Usability for recruiting is not discussed and no procedure for knowledge discovery is presented.

To sum up, none of the existing works has proposed a procedure for knowledge discovery from CVs using topic modeling. Thus, following existing literature in this regard leads to multiple crucial issues, which constitutes the research gap addressed by this paper: First, not considering the type and characteristics of documents to be analyzed produces non-optimal results. For instance, this is due to generic text pre-processing (e.g., in the case of CVs, no removal of author's contact details). Second, existing approaches cannot even be readily applied, as essential steps are not described. For example, the acquisition of CVs and their general pre-processing is not discussed in existing topic modeling literature. Finally, the goals of applying topic modeling to CVs are not considered in existing works. This means that, critically, it remains unclear how to use the topic modeling results for actual benefit in HRM (e.g., for recruiting).

3 Knowledge Discovery from CVs

Subsequently, our procedure for knowledge discovery from CVs is presented. Figure 1 illustrates the five steps of the procedure, adapted from the process for topic modeling in general IS as proposed in (Debortoli et al., 2016). After the initial acquisition of CVs (i), general pre-

processing (ii) and text pre-processing (iii) are required. Only then, the pre-processed CVs can be analyzed by applying LDA (iv). Finally, the results of the application can be interpreted and used (v). The five steps are described in detail in the following.

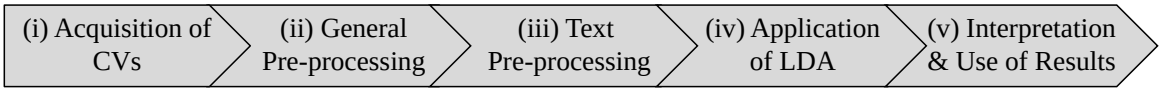

Figure 1. Steps for Knowledge Discovery from CVs

3.1 Acquisition of CVs

With unstructured data proliferating on the web, many options for acquiring CVs are available. The most prominent ones are utilizing (a) professional social networks such as *LinkedIn*, (b) specialized portals such as *Indeed* and (c) a web crawler.

(a): Professional social networks offer members the opportunity to present themselves to companies and recruiters via disclosing information on their profile. This includes, in particular, past work experience, education, skills, abilities, publications and interests as well as contact information. The information can be extracted from the profiles to generate CV-like documents. Internal and external tools for extraction are readily available for the most common professional social networks (e.g., *LinkedIn, XING*). For example, *LinkedIn* itself offers a native functionality to export member profiles as CVs in PDF format. Moreover, these portals allow members to directly upload their CVs, which can then be accessed and stored.

(b): Many job portals (e.g., *Indeed*, *CareerBuilder*, *Monster*) encourage their users to post their CV. These portals offer a (keyword) search engine which can be used to obtain CVs. For example, in Q2 2018, an exemplary search for CVs with "Data Scientist" as job title in New York City produced over 1,100 hits on *Indeed*. CVs resulting from a search can be accessed and, subsequently, stored in PDF format.

(c): Another opportunity for the acquisition of publicly available CVs is the use of a web crawler. A web crawler is an automated program able to navigate the web and store relevant information. Specifically, such a web crawler can be fed with desired search terms and programmed to find and store PDFs including these search terms. This allows the acquisition of CVs from the general searchable web. In particular, also CVs available on private homepages can be found and stored.

3.2 General Pre-processing

Once a sufficient quantity of PDFs containing CVs has been obtained, a general pre-processing of this collection of data is required to ready the collection for text pre-processing and further analyses. Depending on the way the PDFs were obtained, the common challenges (C1) or (C2) may need to be resolved:

(C1) The language of the documents does not match, causing problems for many text pre-pro-

cessing routines. To address this issue, an approach for automatic language identification (Jauhiainen et al., 2018; Shuyo, 2010) can be used. A high quality of automatic language identification can be achieved as CVs contain a substantial number of words.

(C2) The collection of data does not exclusively consist of CVs (e.g., when the PDFs were obtained using a web crawler). Obviously, this issue can lead to unsatisfactory results in the later stages of analysis, for instance, when a job offer instead of a CV is erroneously assessed to be the optimal match for a search query. A human is able to almost instantly decide whether a PDF is a CV or a non-CV with a very high degree of confidence. However, a manual distinction of CVs and non-CVs may still not be promising due to the substantial time effort required for assessing a large collection of data by hand. Thus, automated methods such as classification algorithms can be used.

In any case, PDFs should be converted to a more manageable format (such as TXT) for further analyses, and be fitted in a database for storage. As suggested in (Debortoli et al., 2016), an exploratory data analysis may be performed to detect possible data quality issues and to obtain a general understanding of the data to be analyzed.

3.3 Text Pre-processing

Meaningful text pre-processing before the application of a topic modeling approach is of high importance (Debortoli et al., 2016). This is particularly the case for CVs, which contain many terms or even whole components irrelevant to the envisioned goal of knowledge discovery. There are simple and well-known pre-processing routines which are accepted to be valuable for (almost) all kinds of texts. Common examples are the removal of formatting tags and special characters, tokenization (i.e., splitting up documents into words), lowercasing and the removal of words occurring only in few documents. These routines can be looked up in renowned sources (Boyd-Graber et al., 2014; Weiss et al., 2010); in the following, we discuss routines which possess special characteristics with regard to CVs more explicitly.

N-gram-Creation. N-grams (expressions consisting of two or more words) instead of single words can be considered for further text analysis. For instance, many competence descriptions (e.g., ability in software such as Visual Studio) in CVs contain multiple words. Thus, a thorough creation of n-grams may be of importance. However, care needs to be taken because many skill descriptions in CVs are often used together but are not a real expression. For example, disclosing language skills in English and Spanish is common, which might lead to an incorrect 2-gram "English Spanish".

Stop Word Removal. Words that occur frequently, but are uninformative and decrease the quality and interpretability of topics need to be removed. To achieve this goal in the context of CVs, multiple types of words have to be eliminated. The first type includes general language-specific stop words, which usually are words that have only a grammatical or syntactical function such as prepositions. The second type are CV-specific stop words, which are words commonly occurring in all kinds of CVs (e.g., "resume", "name"). The third type are stop words

specific for the CV database at hand. To identify these stop words, word frequency lists can be used (Boyd-Graber et al., 2014). Finally, numbers may or may not also be seen as "stop words". When competences are to be modeled as topics, numbers tend to obscure the results; thus, they should also be filtered.

Part-of-speech Filtering. Research has provided varying results with respect to which parts of speech should be filtered using LDA (Debortoli et al., 2016; Martin and Johnson, 2015). Against this background and taking into account that CVs contain a word distribution different from other types of documents (e.g., higher prevalence of nouns), part-of-speech filtering needs to be analyzed and adapted to obtain optimal results. In any case, nouns are not to be filtered as they transport essential information regarding competences.

Stemming & Lemmatization. Both stemming and lemmatization aim to decrease the number of considered terms by consolidating similar words. Stemming strives to truncate words to their stem, while lemmatization seeks to reduce words to their dictionary form. Stemming is seen as problematic for the application of LDA (Boyd-Graber et al., 2014; Schofield and Mimno, 2016; Spies, 2017), for instance due to the danger that words with substantially different meaning are consolidated. Lemmatization, on the other hand, has mostly shown positive effects (Martin and Johnson, 2015; Spies, 2017). However, CVs are structured differently than other types of documents, for example with respect to parts of speech; hence, the use of lemmatization should also be analyzed and adapted with respect to the database at hand to obtain optimal results.

Named Entity Recognition. Approaches for named entity recognition classify named entities in text into pre-defined categories (e.g., person names). The appearance of names in CVs is particular. CVs contain the name of the CV's author, possibly other person names (e.g., co-authors of publications), location names (referring to, e.g., company sites) and organization names. Location names may be useful in order to pinpoint expertise in certain areas such as the D-A-CH region. Organization names are of high relevance for many descriptions of competences (e.g., Microsoft Office). Person names, however, do not contribute to interpretable topics and should be filtered.

Overall, it has to be stated that – as it is usually the case in text mining – finding a very good pre-processing configuration for topic modeling of CVs is a non-trivial task. One has to experiment with different configurations to obtain optimal results with respect to the database at hand. Based on the discussion above, in the context of CVs, it seems particularly sensible to fix most steps but to experiment with n-gram creation, part-of-speech filtering and lemmatization.

3.4 Application of LDA

LDA requires as input two hyperparameters α and β as well as the total number of topics N. The shape of the CV-topic-distributions is determined by α (Blei et al., 2003). When α is large, CVs are described by many topics and thus competences, whereas a small α leads to few topics per CV. Obviously, α should neither be too large (resulting in an unwieldy description of CVs

which does not carve out the main competences) nor too small (resulting in only the most prominent competence being identified). The shape of the word-topic-distributions is controlled by β (Blei et al., 2003). A large β implies that topics are widespread (i.e., competences are described broadly). A small β, in turn, leads to narrow topics and competences. In practice, α and β are often set to standard values (e.g., $1/N$) which have been shown to work well for a large range of application contexts (Debortoli et al., 2016). Alternatively, an optimization can be performed (Wallach et al., 2009).

If the number of topics N is too small, resulting topics may be general and widespread, representing a large variety of competences. For example, in a database containing CVs from IT experts, programming skills may constitute a single topic and not be differentiated further. As a result, topic distributions of CVs are not very meaningful: The competences of two persons portrayed by CVs with a similar topic distribution may still differ substantially. On the other hand, the more topics, the more challenging it is for humans to grasp all word-topic-distributions and to interpret CV-topic-distributions. Moreover, if N is too large, resulting topics may be very similar to each other. Thus, (almost) the same competences can be represented by multiple topics, leading to interpretation difficulties. The competences of two persons portrayed by CVs with a largely differing topic distribution may still be similar. To determine a favorable number of topics N, evaluation methods for topic models (cf. Section 2.1) can be used. Then, the resulting topics can be analyzed with regard to their human interpretability. In particular, it can be checked whether competences of interest are represented by topics or pre-processing configuration and LDA application need to be refined.

3.5 Interpretation & Use of Results

Once the topic model has been constructed, the actual knowledge discovery can begin. The topic model provides both word-topic-distributions and CV-topic-distributions.

The word-topic-distributions can be analyzed to obtain an understanding of the subjects generally present in the CVs. On a more fine-grained level, analyses of each topic – in particular, of the words with the highest probability in each topic – can be conducted to allow for their interpretation. Ideally, many of the topics clearly represent specific competences (e.g., web development). It is to be expected that also other topics representing, for instance, university or school career are contained in the model. In any case, as long as topics are interpretable, they should be labelled accordingly. Preferably, multiple persons label topics independently and compare their assessments afterwards.

The CV-topic-distributions offer a succinct description of each CV's topics. They allow to analyze CVs with respect to contained topics, in particular with respect to the competences of the portrayed person. This is especially helpful when topics have already been labelled. Then, the competences of a person portrayed by a CV can be assessed rapidly by observing the respective CV-topic-distribution and taking into account the labels associated to the most prevalent topics. Such an assessment is useful in HRM (e.g., for swift decision support in regard to the relevance

of applicants for a job offer). Here, it is also important to note that using LDA, the topic distribution of a fresh CV can be determined quickly without re-running the whole model. Moreover, based on CV-topic-distributions, CVs may be categorized or tagged for future use. For example, all CVs which exhibit a proportion above 40% for a topic representing web development skills can be marked as relevant for future job offers in this area.

Furthermore, based on word-topic-distributions and CV-topic-distributions, techniques for post-processing LDA results can be applied. This includes in particular visualization approaches (e.g., Chaney and Blei, 2012; Sievert and Shirley, 2014) which assist analysts in gaining an overview and interpreting. For instance, LDAvis (Sievert and Shirley, 2014) provides clear illustrations of word-topic-distributions and offers to display terms particularly characteristic for a topic based on a relevance metric. This can be helpful to pinpoint rare but valuable competences occurring almost exclusively in a certain topic. If a specialist possessing this rare skill is required, the respective CV can then be retrieved quickly.

Obviously, there are further possibilities yet to be explored. We describe the following technique of a *topic-based search for CVs* as an exemplary idea. To facilitate this technique, in a first step, topics of interest and interpretable as competences are extracted from the LDA model and labelled. Based upon these topics, a query can be formulated which represents the desired competences in form of a search vector. For instance, if a Java developer with complementary competences in web development and web design is in demand, the emphasis may be 50% on Java programming, 30% on web development and 20% on web design. The search vector would thus include the values 0.5, 0.3 and 0.2 for topics representing Java programming, web development and web design respectively and 0 for all other topics. Then, the similarity between the search vector and the CV-topic-distributions of each CV can be calculated based on established similarity measures such as cosine similarity or Kullback-Leibler divergence for topics (Koltcov et al., 2014). The most similar CVs can be manually screened and promising candidates can be contacted for recruiting. Such a topic-based search possesses clear advantages compared to a usual keyword search (e.g., also on platforms such as *LinkedIn* and *Indeed*), which can be illustrated by the example above:

1) The topic-based search allows to search for actual competences and not just for words which may or may not represent these competences reasonably well. For instance, a CV may not contain the term "web development" but the portrayed person may still report experience in JavaScript, HTML and PHP. The respective CV would be deemed irrelevant by a keyword search, whereas the topic-based search acknowledges the competence in web development.

2) In the topic-based search, it is possible to put emphasis on an aspect and the mere occurrence of a keyword is not enough for a CV to be relevant. For example, the specification that Java skills should make up for 50% of a CV's topic distribution means that CVs indeed need to contain a lot of Java-related terms to be assessed as relevant in the topic-based search. In contrast, a keyword search for terms such as Java is not very promising because a large number of CVs will claim at least some competence in Java.

3) The topic-based search allows to specify a weighting between different competences. In the example, Java programming is weighted with 50%, web development with 30% and web design with 20%. However, in a simple keyword search, each keyword would be treated as equally important. Weighting allows to more accurately identify the candidates which fit the requirements best (e.g., that Java development skills are most important and the other competences are complementary).

4 Application and Evaluation

In this section, we first describe how we exemplarily applied the procedure presented in Sections 3.1-3.5. In addition to this demonstration of practical applicability, we also evaluate the feasibility of the results within Section 4.5.

4.1 Acquisition of CVs

For our exemplary application, we decided to focus on CVs from IT experts for the following reasons. First, IT experts often possess diverse competences (e.g., programming languages, software, ...) which they report in their CVs. A topic modeling approach adapted to CVs should be able to identify these competences and categorize them into interpretable topics. Second, focusing on a single area provides a particular challenge for the procedure. CVs portraying persons with completely different competences are relatively easy to distinguish. However, a procedure that provides good results even when applied to CVs which are quite similar to each other – as in this case, CVs from IT experts – is of greater practical usefulness because more fine-grained distinctions can be made. Third, projects in companies often require IT experts with specific competences and thus, this application context is highly important.

To obtain CVs from IT experts, we used a web crawler (cf. Section 3.1, c)). This choice was made to allow the acquisition of CVs from private homepages, which many IT freelancers maintain. The web crawler was fed with search terms commonly used in IT (e.g., “Java”) and including “CV”. Based on these search terms and starting from the *Google* search, the web crawler stored approximately 27,000 PDFs.

4.2 General Pre-processing

To ready the collection for text pre-processing, we had to resolve the challenges (C1) and (C2) described in Section 3.2. PDFs were stored in a MongoDB database and converted to TXT for further analyses.

In order to address (C1) the diverse languages present in the collection, we performed automatic language identification and partitioned the collection accordingly. To this end, we used a Java open source tool (Shuyo, 2014). On a manually inspected test sample of 100 documents, the approach did not produce any errors. We proceeded with German documents as those represented the largest proportion of the collection.

With regard to (C2), we observed that a quite large percentage of documents were not actually CVs. Thus, we manually classified documents into CVs and non-CVs to obtain a training dataset and built a classification model based on majority voting of the common classification methods logistic regression, support vector classification and random forests. The classification into CVs and non-CVs reached an accuracy of 95% on a test dataset of 1,291 documents and was applied to the remainder of the collection.

Overall, general pre-processing resulted in a database of 2,410 (presumed) CVs in German language which we used for further analyses. An exploratory data analysis was performed to obtain an overview. For instance, we determined the number of total (3,504,014) and unique (242,416) terms in the database and analyzed which terms occurred most frequently (all of them common stop words for German texts).

4.3 Text Pre-processing

The pre-processing routines with special characteristics in regard to CVs were set up as follows: In order to determine a comprehensive stop word list, we started by integrating established lists (Götze and Geyer, 2016; Salton and Buckley, 2018). Then, we modified this list by incorporating CV-specific stop words based on own reflections as well as an analysis of the 1,500 most frequent words in the database. The CV-specific stop words included in particular time specifications, legal forms of organizations and forms of address. Numbers were filtered as well. In contrast, terms such as “C” or “R” were removed from the list, as they represent ability in the respective programming languages in the given context.

To create n-grams, we used the NPMI-based method from the Python topic modeling library gensim (Řehůřek and Sojka, 2010). 2-grams were created for terms with a NPMI value of at least 0.5 and a joint occurrence frequency of at least 100. Analogously, 3-grams were created from 2-grams (e.g., “Microsoft Visual” and “Visual Studio” were combined to “Microsoft Visual Studio”) and so on. In this way, many n-grams were created, the most frequently used ones being “SQL Server”, “SAP R3” and “MS Office”.

Part-of-speech tagging was realized by the aggregated results of two taggers. We used TreeTagger (Schmid, 2018), a tagger based on a probabilistic Markov model pre-trained for the German language, and the Natural Language Toolkit (NLTK Project, 2018) tagger which was trained with the TIGER corpus (Institut für maschinelle Sprachverarbeitung, 2018). On a manually inspected test sample of 400 words, tagging in this way exhibited an accuracy of 96%. TreeTagger was also used for lemmatization.

For named entity recognition, the Stanford NER Tagger (Finkel et al., 2005) was used. Its results were then refined by publicly available lists of first names (Kolb, 2007; Michael, 2008) and a phone book for surnames, achieving a true positive rate of 97% in regard to filtered person names.

As suggested in Section 3.3, we fixed many of the pre-processing routines but let others vary

and experimented in order to achieve optimal results. To be more precise, we always – and in this order – removed formatting tags and special characters, tokenized and lowercased the CVs and removed stop words and words occurring only in few documents. We experimented with n-gram-creation (yes/no), lemmatization (yes/no), part-of-speech filtering (possibly filtering adjectives and/or verbs and/or adverbs) as well as the threshold for words to occur in too few documents (50/40/30/20/10). Overall, this resulted in 160 pre-processing configurations.

For each configuration, LDA models were generated using the gensim libary (Řehůřek and Sojka, 2010) and each number of topics $N \in \{25,50,75,100\}$, following (Wallach et al., 2009). Convergence was tested as suggested in (Hoffman et al., 2010). Subsequently, the generated LDA models were evaluated with respect to semantic coherence (cf. Section 2.1). We determined the configuration leading to the highest value of semantic coherence and used it for optimizing the LDA application (cf. Section 4.4). Thereafter, to verify the results, we re-ran the evaluation of all configurations with the optimized LDA model. The configuration leading to the highest value of semantic coherence (NPMI: 0.161) did not consider n-gram-creation and lemmatization, filtered adjectives as well as verbs and adverbs, and filtered all words occurring in less than 40 documents.

The results in regard to n-gram-creation and lemmatization may surprise at first. However, they are in line with previous research in other application contexts suggesting that LDA derives required semantic relations itself and stronger pre-processing reduces topic model quality (Schofield and Mimno, 2016). Similarly, filtering all parts of speech except nouns has also already been shown to provide strong results (Martin and Johnson, 2015) and is expected to be promising for CVs, with competences usually being described by nouns.

4.4 Application of LDA

The hyperparameters α and β were optimized similar to the pre-processing configuration, again following (Wallach et al., 2009). To determine the number of topics N, we generated LDA models for each $N \in \{2,4,6,...,100\}$ based on the optimal pre-processing configuration. We then analyzed semantic coherence for each N. This led to choosing N=42, a number of topics manageable for humans. Thus, the respective topic model was examined further with respect to interpretability and use of results.

4.5 Interpretation & Use of Results

The interpretation of the topic model was conducted separately by two human coders to account for human subjectivity. Consolidating the interpretation only required to settle minor wording differences. The word-topic-distributions of each topic were analyzed to obtain an overview of the topic model. We observed that topics could generally be categorized into three groups: Group A, the largest group, contained topics describing specific IT-related competences and consisted of 23 topics. The four topics in Group B described competences concerning business & management. The remaining 15 topics in Group C were related to university or school career

and, due to our focus on competences, not considered for further analysis. Some topics are shown exemplarily in Table 1. Thereby, a topic is represented by its seven words with highest probability in decreasing order and translations for German words are provided in square brackets. Please note that many English words are used frequently in German CVs, explaining their occurrence in topics.

ID	*Gr.*	*Topic (most probable words)*
1	A	java eclipse entwicklung[development] xml spring j2ee oracle
2	A	linux server administration unix system perl security
3	A	design adobe konzeption[conception] 3d web programmierung[programming] photoshop
4	A	entwicklung[development] web javascript php mysql css html
5	A	windows server ms microsoft support software office
6	A	c r analysis time solution network networks
7	B	management projekt[project] einführung[launch] analyse[analysis] projektmanagement[project management] business durchführung[execution]
8	C	university research school international european science german

Table 1. Exemplary Topics from each of the Groups A, B, C

Overall, the analysis yielded that most of the topics in Groups A and B are fine-grained topics clearly representing specific competences. More precisely, they do not describe competences rather general for CVs from IT experts such as programming skills, but more distinguishing competences such as programming skills in Java (cf. Topic 1). Employing the relevance metric of LDAvis (Sievert and Shirley, 2014), it was possible to differentiate topics even further and carve out competences highly characteristic for a topic. Usually, this concerned closely associated special frameworks, software or technologies. For instance, the most relevant words of Topic 1 (describing programming skills in Java) then were: jaxb, j2se, jpa, jax, ejb, hibernate, jms. All of them are Java-specific and occurred almost exclusively in Topic 1.

The results support that the topics possess a clear interpretation with respect to describing competences and are thus useful for HRM. Furthermore, based on these results, a topic-based search seemed promising. We labelled 21 of the 27 topics in Groups A and B with respect to the competence described (the remaining six topics were judged to not be as clearly interpretable and left out). Besides their use to rapidly assess the competences represented in a CV for HRM (cf. Section 3.5), the labels facilitated the topic-based search. Our prototypical implementation allows the specification of search queries as vectors containing the desired weight for each topic. Similarities between the search vector and the CV-topic-distributions of each CV are calculated based on Kullback-Leibler divergence (Koltcov et al., 2014). The most similar CVs are shown together with their topic distribution. Figure 2 illustrates the GUI of the prototypical implementation with the search query from Section 3.5 and the first search result (CV #545). Clicking on a search result opens the respective CV.

Figure 2. Topic-based Search for CVs

CV #545 portrayed a senior Java developer with skills in a large number of Java-related technologies (Spring, Struts, JUnit, JEE, ...). The CV also claimed a lot of work expertise in web development such as the programming of web frontends using HTML and CSS. To a lesser extent, competences in web design and respective software (e.g., Photoshop, Gimp) were reported as well. Thus, the CV fit the job offer represented by the search query exceptionally well. The analysis of the other top 10 CVs which were determined to be the best match for the search query yielded similar results.

For comparison, we also performed a keyword search using the search term *java AND "web development" AND "web design"*. Here, we observed all three advantages of a topic-based search outlined in Section 3.5: With respect to 1), the keyword search only yielded four results because few CVs actually followed the exact wording dictated by the search term. In particular, many suitable CVs such as the ones found by our topic-based search were neglected by the keyword search. Regarding 2), the problem of merely focusing on keywords became obvious as the CV of a manager who once had conducted a Java project was included in the four results of the keyword search, but did in fact not fit the job offer. Concerning 3), the lack of weighting showed when the CV of a web developer specializing in PHP, HTML and JavaScript with basic Java abilities was assessed as relevant. Overall, none of the results of the keyword search fit the job offer well. Our topic-based search thus produced clearly superior results in this setting.

We further specified eight more search queries and analyzed the respective CVs suggested by the topic-based search. In each case, the competences reported in the top CVs coincided with the competences called for by the search query. To conclude, the topic-based search worked very well in this application and seems fit to provide helpful decision support for HRM.

5 Conclusion, Practical Implications and Directions for Future Work

In this paper, a topic modeling procedure consisting of five steps with the aim of discovering knowledge from CVs has been presented. CV-specific characteristics are considered in each step. An exemplary application to CVs from IT experts suggests that clearly interpretable topics describing fine-grained competences (e.g., Java programming, web design) can be discovered. This information can be used to rapidly assess the contents of a CV, categorize CVs and identify promising candidates for job offers, thus providing decision support in HRM.

The presented procedure allows for proactive recruiting. It can, for instance, be applied in HRM similar to how professional social networks are currently used in the recruitment process to rapidly source candidates before subsequent steps such as job interviews are conducted. However, it is not restricted to members of these networks as the analyzed CVs may stem from any origin. Additionally, the presented topic-based search possesses advantages compared to a keyword search on these platforms. Moreover, the CV-topic-distributions in conjunction with labels can be used to categorize and tag CVs for future use. In this way, companies can construct and steadily extend a database of interesting CVs. Another promising idea for companies is to also include CVs of own employees to promote internal recruiting. In any case, HRM and IT departments need to cooperate as skills from both areas are required to achieve a successful application of the procedure.

While the paper at hand offers a detailed description of a procedure for knowledge discovery from CVs, there are also limitations which provide directions for further research. First, an application to CVs from a different context should be conducted to validate feasibility. Second, a topic-based search technique has been presented and evaluated in an exemplary setting. However, it should be further assessed, for instance by a more detailed comparison to alternatives (e.g., with the help of a HRM expert) and an application to real job offers from a company. Finally, CV-specific visualization approaches should be developed, allowing for an easier overview and use of the results of the procedure. They should be included in a tool facilitating and partly automating the five steps of the procedure to further its practical use.

6 References

Abel, F., Y. Deldjoo, M. Elahi and D. Kohlsdorf (2017). “Recsys challenge 2017: Offline and online evaluation”. In: *Proceedings of the 11th ACM RecSys*, pp. 372–373.

Aletras, N. and M. Stevenson (2013). “Evaluating topic coherence using distributional semantics”. In: *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pp. 13–22.

Allden, N. and L. Harris (2013). “Building a positive candidate experience: towards a networked model of e-recruitment” *Journal of Business Strategy (JBS)* 34 (5), 36–47.

Belford, M., B. Mac Namee and D. Greene (2018). "Stability of topic modeling via matrix factorization" *Expert Systems with Applications* 91, 159–169.

Blei, D. M. (2012). "Probabilistic topic models" *Communications of the ACM* 55 (4), 77–84.

Blei, D. M., A. Y. Ng and M. I. Jordan (2003). "Latent dirichlet allocation" *Journal of Machine Learning Research (JMLR)* 3 (Jan), 993–1022.

Boyd-Graber, J., D. Mimno and D. Newman (2014). "Care and feeding of topic models: Problems, diagnostics, and improvements" *Handbook of mixed membership models and their applications*, 225–255.

Breaugh, J. A. (2008). "Employee recruitment: Current knowledge and important areas for future research" *Human Resource Management Review (HRMR)* 18 (3), 103–118.

Chaney, A. J.-B. and D. M. Blei (2012). "Visualizing Topic Models". In: *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM 2012)*.

Chang, J., S. Gerrish, C. Wang, J. L. Boyd-Graber and D. M. Blei (2009). "Reading tea leaves: How humans interpret topic models". In: *Proceedings of the Neural Information Processing Systems Conference (NIPS 22)*, pp. 288–296.

Coffey, A. and P. Atkinson (1996). *Making sense of qualitative data: complementary research strategies:* Sage Publications, Inc.

Debortoli, S., O. Müller, I. Junglas and J. vom Brocke (2016). "Text mining for information systems researchers: an annotated topic modeling tutorial" *Communications of the Association for Information Systems (CAIS)* 39 (7), 111–135.

Dumont, T., P. Fettke and P. Loos (2016). "Towards multi-dimensional Clustering of Business Process Models using Latent Dirichlet Allocation". In: *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2016)*, pp. 69–80.

Fang, D., H. Yang, B. Gao and X. Li (2018). "Discovering research topics from library electronic references using latent Dirichlet allocation" *Library Hi Tech* 36 (3), 400–410.

Finkel, J. R., T. Grenager and C. Manning (2005). "Incorporating non-local information into information extraction systems by gibbs sampling". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*.

Gao, L. and N. Eldin (2014). "Employers' expectations: A probabilistic text mining model" *Procedia Engineering* 85, 175–182.

Gorbacheva, E., A. Stein, T. Schmiedel and O. Müller (2016). "The role of gender in business process management competence supply" *Business & Information Systems Engineering (BISE)* 58 (3), 213–231.

Götze, M. and S. Geyer (2016). *German stopwords*. URL: https://github.com/solariz/german_stopwords/blob/master/german_stopwords_full.txt (visited on 06/01/2018).

Guo, Y., S. J. Barnes and Q. Jia (2017). "Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation" *Tourism Management* 59, 467–483.

Gupta, V. and G. S. Lehal (2009). "A survey of text mining techniques and applications" *Journal of Emerging Technologies in Web Intelligence* 1 (1), 60–76.

Hendry, C. (2012). *Human resource management:* Routledge.

Hoffman, M., F. R. Bach and D. M. Blei (2010). "Online learning for latent dirichlet allocation". In: *Proceedings of the Neural Information Processing Systems Conference (NIPS 23)*, pp. 856–864.

Institut für maschinelle Sprachverarbeitung (2018). *TIGER corpus*. URL: http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html (visited on 06/02/2018).

Jauhiainen, T., M. Lui, M. Zampieri, T. Baldwin and K. Lindén (2018). "Automatic language Identification in texts: A survey" *arXiv preprint arXiv:1804.08186*.

Kolb, P. (2007). *Liste mit Vornamen*. URL: http://www.ling.uni-potsdam.de/~kolb/Vornamen.txt (visited on 06/02/2018).

Koltcov, S., O. Koltsova and S. Nikolenko (2014). "Latent dirichlet allocation: stability and applications to studies of user-generated content". In: *Proceedings of the 6th ACM Conference on Web Science*, pp. 161–165.

Lau, J. H., D. Newman and T. Baldwin (2014). "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pp. 530–539.

LinkedIn (2018). *About LinkedIn*. URL: https://about.linkedin.com/ (visited on 06/12/2018).

Martin, F. and M. Johnson (2015). "More efficient topic modelling through a noun only approach". In: *Proceedings of the Australasian Language Technology Association Workshop (ALTA 2015)*, pp. 111–115.

Michael, J. (2008). *Anredebestimmung anhand des Vornamens*. URL: https://www.heise.de/ct/ftp/07/17/182/ (visited on 06/02/2018).

Newman, D., J. H. Lau, K. Grieser and T. Baldwin (2010). "Automatic evaluation of topic coherence". In: *Proceedings of the 8th North American Chapter of the Association for Computational Linguistics*, pp. 100–108.

NLTK Project (2018). *Natural Language Toolkit*. URL: https://www.nltk.org/ (visited on 06/02/2018).

Řehůřek, R. and P. Sojka (2010). "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50.

Röder, M., A. Both and A. Hinneburg (2015). "Exploring the space of topic coherence measures". In: *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM 2015)*, pp. 399–408.

Salton, G. and C. Buckley (2018). *Stop word list 2*. URL: http://www.lextek.com/manuals/onix/stopwords2.html (visited on 06/01/2018).

Schmid, H. (2018). *TreeTagger - a part-of-speech tagger for many languages*. URL: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (visited on 06/02/2018).

Schofield, A. and D. Mimno (2016). "Comparing apples to apple: The effects of stemmers on topic models" *Transactions of the Association for Computational Linguistics* 4, 287–300.

Shuyo, N. (2010). *Language detection library*. URL: https://www.slideshare.net/shuyo/language-detection-library-for-java (visited on 06/01/2018).

Shuyo, N. (2014). *Language detection*. URL: https://github.com/shuyo/language-detection (visited on 06/01/2018).

Sievert, C. and K. Shirley (2014). "LDAvis: A method for visualizing and interpreting topics". In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70.

Spies, M. (2017). "Topic modelling with morphologically analyzed vocabularies" *Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and mechanics* 9 (1), 1–18.

Strohmeier, S. and F. Piazza (2013). "Domain driven data mining in human resource management: A review of current research" *Expert Systems with Applications* 40 (7), 2410–2420.

Wallach, H. M., D. M. Mimno and A. McCallum (2009). "Rethinking LDA: Why priors matter". In: *Proceedings of the Neural Information Processing Systems Conference (NIPS 22)*, pp. 1973–1981.

Weiss, S. M., N. Indurkhya, T. Zhang and F. Damerau (2010). *Text mining: predictive methods for analyzing unstructured information:* Springer Science & Business Media.

3.2 Paper 5: Explaining the Stars: Aspect-based Sentiment Analysis of Online Customer Reviews

Current Status	Full Citation
accepted and published in the 2019 Proceedings of the European Conference on Information Systems (06/2019)	Binder, M., B. Heinrich, M. Klier, A. Obermeier and A. Schiller (2019). "Explaining the Stars: Aspect-based Sentiment Analysis of Online Customer Reviews". In: *Proceedings of the 27th European Conference on Information Systems (ECIS)*, June 8-14, Stockholm-Uppsala, Sweden.

Summary

This paper addresses RQ5 by presenting a formally defined model for explaining and interpreting the overall star ratings of online customer reviews employing aspect-based sentiment analysis. To this end, a generalized ordered probit model using aspect-based sentiments as independent variables is used. Further, a likelihood-based pseudo R-squared measure is suggested to measure the explanatory power of the model. In this way, methodical issues associated with the ratings (in particular, their ordinal scale) are handled. The approach is evaluated using a large real-world dataset of restaurant reviews. It is assessed both methodically in comparison to alternative regression models (e.g., linear regression) as well as with respect to the provided results in regard to customer assessments and opinions. Moreover, implications for theory and practice are discussed.

Similar to Paper 4, this work employs a variety of existing concepts and methods from different fields, in particular natural language processing. The extraction of aspect-based sentiments is based on well-established methods for this task and in particular comprises dependency parsing. The proposed generalized ordered probit model and pseudo R-squared measure build on respective contributions suggested in the literature and are adjusted to the context at hand. As supported by the evaluation, the approach leads to results that are easy to interpret and provide valuable insights into customer assessments and opinions, revealing why specific customer ratings were assigned to a company or a competitor. These insights, in turn, allow for data-driven competitive advantages, for instance by providing decision support for the development of customer-centric solutions to improve customer satisfaction.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style. Moreover, terms only common in British English have been converted to corresponding American English terms.

The paper as published by AIS is available at: https://aisel.aisnet.org/ecis2019_rp/169/

Abstract:

The importance of online customer reviews for the success of products and services has been recognized in both research and practice. Therefore, the ability to explain and interpret customer assessments expressed by the assigned overall star ratings is an important and interesting research field. Existing approaches for explaining the overall star ratings, however, often do not address methodical issues associated with these ratings (e.g., ordinal scale). Moreover, they often ignore the review texts which contain valuable information on the customers' assessments of different aspects of the rated items (e.g., price or quality). To contribute to both research gaps, we propose a generalized ordered probit model using aspect-based sentiments as independent variables to explain the overall star ratings of online customer reviews. For measuring the explanatory power of our model, we suggest a likelihood-based pseudo R-squared measure. By evaluating our approach using a large real-world dataset of restaurant reviews we show, that, in contrast to other regression models, the generalized ordered probit model can address the methodical issues associated with the star ratings. Moreover, the evaluation shows that the results of the proposed model are easy to interpret and valuable for analyzing customer assessments.

Keywords: online customer reviews, explanatory model, aspect-based sentiment analysis, generalized ordered probit model

1 Introduction

In recent years, the number of internet users has increased from 1,024 million in 2005 up to 3,578 million in 2017 (ITU, 2017). This increase has considerably contributed to the rise of popular platforms such as *Amazon* (Linden et al., 2003) or *TripAdvisor* (Filieri et al., 2015) which, inter alia, provide access to online customer reviews (O'Mahony and Smyth, 2010). Online customer reviews can be an important instrument to reduce information asymmetries about offered products and services (Hu et al., 2008). They contain rich information about customers' assessments and opinions in form of user generated content (Ye et al., 2011) and typically consist of an overall star rating (e.g., 1 to 5 stars) and a textual part (Mudambi et al., 2014). The overall star ratings summarize the customers' general impressions of the rated items. The textual parts comprise further details on the customers' assessments, often towards different aspects of the rated items (e.g., service quality in a restaurant review), to justify and explain the associated overall ratings (Zhu et al., 2011). Indeed, literature already provides some approaches to analyze these textual assessments in terms of aspect-based sentiments (Schouten and Frasincar, 2016).

Online customer reviews may affect the economic success of products and services considerably (e.g., Chevalier and Mayzlin, 2006; Clemons et al., 2006; Minnema et al., 2016; Phillips et al., 2017; Ye et al., 2009; Ye et al., 2011; Zhu and Zhang, 2010). Research has shown that

besides high overall star ratings, positive feedback contained in the textual parts reviews yields, amongst others, higher sales volumes (Archak et al., 2007, 2011; Ghose and Ipeirotis, 2011). Even though the analysis of structural data, such as star ratings or metadata on the items, is predominantly focused by existing literature, the textual parts of reviews have been shown to comprise very valuable information (Ganu et al., 2013). In that line, some predictive models have been proposed (e.g., Goldberg and Zhu, 2006; Li et al., 2011; Pang and Lee, 2005; Qu et al., 2010) which aim to predict the star ratings based on review texts. However, these models mostly rely on latent variables which are hard to interpret as they do not necessarily represent the thematic aspects focused by the users when reviewing the item. Indeed, explaining and interpreting the overall star ratings based on such predictive models is not aimed at or possible. To make the rich information contained in the review texts accessible, an explanatory model is needed, which uses easy to interpret independent variables like aspect-based sentiments. Such an explanatory model enables the identification of causal relationships between the independent variables (i.e., aspect-based sentiments) and the dependent variable (i.e., the associated overall star rating) (Sainani, 2014).

Aspect-based sentiment analysis accounts for the review texts including the users' assessments of different aspects of the rated items in a methodically well-founded way (Jo and Oh, 2011; Schouten and Frasincar, 2016; Zhu et al., 2011). In that line, we use aspect-based sentiments contained in the review texts and propose an approach to explain and interpret the users' overall star ratings. We focus on the following research question:

> *How can aspect-based sentiments contained in the textual parts of online customer reviews be used to explain and interpret the associated overall star ratings?*

To answer this question, we aim at an explanatory model (cf. Shmueli, 2010; Shmueli and Koppius, 2011) to explain the associated overall star ratings based on easy to interpret aspect-based sentiments. We argue that the principles and the knowledge base of regression theory are adequate and valuable, providing well-founded methods to analyze and explain the associated overall star ratings of online customer reviews. In general, results of a regression analysis are easy to interpret as they allow to understand how the dependent variable (i.e., the overall star rating) changes on average, when the independent variables (i.e., the aspect-based sentiments) are varied (Myers, 1990). However, focusing on the given problem definition, the application of a regression analysis faces different methodical issues associated with the star ratings. Amongst others, these methodical issues arise from their ordinal scale (e.g., 1 to 5 stars as integer). To address such methodical issues and in contrast to existing approaches, we base our approach on a generalized ordered probit regression model. From a scientific point of view, the proposed approach aims to uncover the underlying reasoning of the overall star ratings as it uses interpretable aspect-based sentiments given in the review texts avoiding any latent variables. For practitioners, our model enables companies to gain a data-driven competitive advantage by being able to analyze the reasoning behind customer ratings and customer assessments. Such an explanation for the users' overall star ratings allows for customer orientation

based on the evidence and importance of different item aspects which are relevant for customer (dis)satisfaction. For example, businesses could focus their efforts on actions to improve on those aspects which influence users' (dis)satisfaction most. Thus, the presented approach provides a way to explain overall star ratings based on the review texts not yet targeted by existing approaches, resolves the associated methodical issues, and is relevant to research and practice.

The remainder of the paper is structured as follows: In the next section, we discuss both the related literature and the research gap. In Section 3, we step-by-step develop our model for explaining star ratings using aspect-based sentiments. In Section 4, we demonstrate and evaluate our approach using a large dataset of restaurant reviews. Section 5 depicts implications of our approach for theory and practice. Finally, we conclude, reflect on limitations and provide an outlook on further research.

2 Related Work and Research Gap

In this section, we analyze existing research which aims at *explaining* overall star ratings of online customer reviews using regression models. Thereby, we also consider works using structural and textual (item) data different from aspect-based sentiments as they might be interesting from a methodological point of view. Existing contributions with a sole predictive (or descriptive) perspective such as Li et al. (2011), Monett and Stolte (2016), Pang and Lee (2005), Qiu et al. (2018), Qu et al. (2010), Sharma et al. (2016), or Zhou et al. (2014) do not aim to explain or interpret the (overall) star ratings and are thus out of scope for our research. These works are not considered in the following.

In accordance with the guidelines of standard approaches to prepare the related work (e.g., Levy and Ellis, 2006; Webster and Watson, 2002), we searched the databases ScienceDirect, Google Scholar, ACM Digital Library, EBSCO Host, IEEE Xplore, and the AIS Library for the following search term and without posing a restriction on the time period: *("regression" and rating*) or ("regression" and review*) or ("regression" and "recommender")*. Additionally, we performed a forward and backward search starting from highly relevant papers. The papers found were manually screened based on title, abstract, keywords and summary. The 51 papers remaining after this first screening were analyzed in detail and 11 of them were identified as relevant for our work.

	Consideration of aspect-based sentiments	Addressing methodical issues (e.g., the ratings' ordinal scale)	Evaluation of the explanatory power of the model
Approaches considering structural (item) data			
Guo et al. (2016); Liu et al. (2017); Radojevic et al. (2017); Ye et al. (2014)	n/a	n/a	✓
Yang et al. (2018)	n/a	✓	n/a
Approaches considering textual (item) data			
Fu et al. (2013); Linshi (2014)	n/a	n/a	n/a
Debortoli et al. (2016); Xiang et al. (2015)	n/a	n/a	✓
Ganu et al. (2009); Ganu et al. (2013)	✓	n/a	n/a

Table 1. Existing Approaches for explaining the overall Star Ratings of Online Customer Reviews

Table 1 provides an overview of the identified papers. They contribute to the problem of modeling the overall star ratings of online customer reviews using regression models with different sets of independent variables (i.e., structural (item) data or textual (item) data). The respective approaches are grouped depending on the characteristic of these independent variables (highlighted by different shades and subheadings). The first column of Table 1 states whether aspect-based sentiments are considered. The second column indicates whether the proposed regression models address methodical issues relevant in the context of explaining overall star ratings. For example, it is necessary to consider the fact that the dependent variable (i.e., the overall star rating) is ordinally scaled (i.e., discrete and ordered) (Debortoli et al., 2016). The third column states whether the explanatory power of the regression model is evaluated using a well-founded quality measure (e.g., the explained variance).

Guo et al. (2016), Liu et al. (2017), Radojevic et al. (2017) and Ye et al. (2014) use regression models with structural data as independent variables to model the overall star ratings of reviews and evaluate the explanatory power of their models by calculating (adjusted) R-squared values. Radojevic et al. (2017) propose a linear multi-level regression model for overall star ratings, using structural data regarding the items (e.g., *price* or *free internet*) and the users (e.g., regarding nationality or travel experience) as independent variables. Guo et al. (2016), Liu et al. (2017) and Ye et al. (2014) use sub-ratings explicitly given by the users (e.g., room experience and service on a 5-point Likert scale). Thereby, Guo et al. (2016) and Liu et al. (2017) analyze the relationships between explicitly given sub-ratings as independent variables and the overall rating as dependent variable in the hotel domain. Ye et al. (2014) investigate the relationship

between price as independent variable and given sub-ratings for service quality or value as dependent variable. All four works – Guo et al. (2016), Liu et al. (2017), Radojevic et al. (2017) and Ye et al. (2014) – provide first insights in the underlying reasons for customer assessments in online customer reviews. However, in none of these works aspect-based sentiments in the review texts are used. Instead, Guo et al. (2016), Liu et al. (2017) and Ye et al. (2014) rely on explicitly given sub-ratings. In reality such explicitly given multi-ratings represent an exceptional case limiting these approaches to some extent. Moreover, all four works use common linear regression models which do not address the methodical issues that arise when explaining the overall star ratings. In particular, the ordinal scale of the star ratings is not considered. Neglecting such methodical issues may lead to significant misspecifications and thus invalid results. Yang et al. (2018) are the only ones to account for the methodical issue of ordinally scaled overall ratings. They introduce an ordinal regression model to infer the overall star ratings from structural location-based data of items (i.e., hotels). Their aim is to explain a hotel's guest assessments (given by the average rating of the hotel) based on information about the hotel's location (e.g., accessibility to points of interest or the location's surrounding environment). The approach relies on structural data regarding the location and the authors do not aim at using review texts or aspect-based sentiments. Additionally, they do not assess the explanatory power of their model, which is a challenging problem, as there are no standard quality measures for the presented ordinal regression model. Summing up, the approaches using structural data are hampered in their applicability (assumption that sub-ratings are given) and/or by the missing consideration of the methodical issues associated with the star ratings (e.g., the ordinal scale) and/or the respective evaluation of the explanatory power of the model. Additionally, they do not take advantage of the review texts or aspect-based sentiments.

Indeed, there also exist approaches using independent variables derived from textual (item) data to explain the star ratings of online customer reviews. Fu et al. (2013) and Linshi (2014) propose linear regression models to explain the associated star ratings. Thereby, Fu et al. (2013) employ word counts based on the review texts as independent variables. Linshi (2014) use document vectors from a codeword Latent Dirichlet Allocation (LDA) which is able to distinguish different topics based on the connotation (good vs. bad) of the co-occurring words (e.g., good food vs. bad food). However, in both works, the authors use linear regression models which do not account for the methodical issues associated with the overall star ratings like their ordinal scale. Additionally, they do not further investigate the explanatory power of the proposed regression models. Debortoli et al. (2016) and Xiang et al. (2015) indeed analyze the explanatory power of their regression models based on the review texts. Debortoli et al. (2016) – similar to Linshi (2014) – use document vectors from a LDA based on the review texts as explanatory variables. They provide a multinomial logistic regression model for explaining the associated overall star ratings. To assess the explanatory power of their model, the deviance explained is stated. Xiang et al. (2015) propose a linear regression model based on the factor loadings from a factor analysis of the review texts. The explanatory power is assessed in terms of the adjusted R-squared measure. The methodical issues, however, are not addressed in both approaches, as the ordinal

scale of the star ratings is neglected. In addition, document vectors from a LDA (Debortoli et al., 2016) or factor loadings (Xiang et al., 2015), respectively, do not necessarily account for (different) sentiments. For example, different sentiments may be contained in one single topic or factor (e.g., one topic or factor concurrently containing statements for good and bad food) or one sentiment may be distributed over different topics or factors. This weakens the interpretability resp. validity of the results. To conclude, the approaches for explaining the overall star ratings of reviews discussed in this paragraph do not address the methodical issues associated with the overall star ratings. In particular, the ordinal scale of the star ratings is neglected. Moreover, they do not account for aspect-based sentiments.

Ganu et al. (2009) and Ganu et al. (2013) show that aspect-based sentiments contained in review texts can be used to improve recommender systems. Both papers generally focus on predicting a user's star rating for a restaurant based on his or her previous ratings for other restaurants and the ratings of all other users. However, in minor parts of the papers (i.e., Section 3.3 of Ganu et al. (2009) and Section 3.2 of Ganu et al. (2013)) regression models for inferring the associated overall star ratings using aspect-based sentiments are discussed. These regressions are based on sentence types, represented as (aspect, sentiment)-pairs assigned to every sentence. To construct the sentence types, each sentence of the review texts is classified according to one aspect it most probably refers to (e.g., food, service or miscellaneous). Additionally, a sentiment label (e.g., positive, neutral or negative) is assigned to each sentence. On this basis, multivariate regression models for the associated overall star ratings are proposed using sentence type fractions in the review texts as independent variables. More precisely, a sentence type fraction is calculated as the percentage of sentences of that type contained in the review text. Ganu et al. (2009) use a linear and Ganu et al. (2013) a quadratic regression model. Both, however, focus on using aspect-based sentiments to improve recommender systems but do not aim at explaining and interpreting the associated overall star ratings. Therefore, they do not further investigate the explanatory power of the proposed regression models (e.g., in terms of coefficients of determination). Additionally, the allocation of sentiment labels is equivalent to a classification instead of a more fine-grained representation of the sentiments as numerical values. Finally, the authors apply common regression models, which do not address the methodical issues associated to the star ratings (e.g., the ordinal scale).

To conclude, there are very interesting contributions regarding modelling the overall star ratings of online customer reviews which can serve as a basis for further research. To uncover the causal relationships between aspect-based sentiments contained in review texts and the associated overall star ratings, an explanatory model is needed. However, existing literature lacks an explanatory model using aspect-based sentiments to explain the associated overall star ratings which addresses the occurring methodical issues (e.g., ordinal scale of the star ratings). Furthermore, the explanatory power of (different sets of) aspect-based sentiments has not been investigated yet. Due to the methodical issues arising, amongst others from the ordinal scale of the star ratings, this is particularly challenging.

3 A Model to explain Star Ratings

To address this research gap, we propose an explanatory model for overall star ratings with respect to aspect-based sentiments, which addresses the methodical issues associated with the star ratings. We first introduce the basic idea of our approach. Then, we outline a generalized ordered probit model for the analysis of star ratings. Finally, we propose a likelihood-based pseudo R-squared measure for assessing the explanatory power of aspect-based sentiments in this context.

3.1 Basic Idea of our Approach

Our aim is to explain the overall star ratings of textual reviews based on the associated aspect-based sentiments. To do that, first, an adequate regression model addressing the methodical issues for modelling star ratings has to be established. These issues result in particular from both the ordinal scale of star ratings and the characteristics of aspect-based sentiments. Then, the explanatory power of different aspect-based sentiments can be assessed using this model.

Our approach is based on the ordered probit model (McKelvey and Zavoina, 1975). To adequately represent star ratings, we follow a two-step approach. First, an underlying model for continuous preferences instead of discrete star ratings is established (Greene and Hensher, 2010). Then, a non-linear transformation of the underlying preferences onto the rating scale is used. More precisely, the ratings are modelled by dividing the underlying continuous preference variable into intervals of different size.

To elaborate why this two-step approach is proposed, we discuss different methodical issues for modelling star ratings. Thereby, we compare the ordered probit model to a linear regression model because the latter is commonly used in literature (cf. Section 2). First and crucially, an ordered probit model accounts for the ordinal scale of the star ratings, whereas a linear regression model does not and thus might lead to significant misspecifications. To achieve an accurate representation, an explanatory model has to reflect *uneven distances within the (ordinal) rating scale*. For instance, on a scale from 1 to 5 a rating of 4 might, on average, be much closer to a rating of 5 with respect to the underlying preference than to a rating of 3 (Greene and Hensher, 2010). A linear regression model is not able to cope with this issue, whereas the ordered probit model accounts for uneven distances within the rating scale by assigning preference intervals of different sizes to the ratings. Further, a model for star ratings must cope with a *non-normal distribution of the rating errors* (due to the star ratings being discrete) and with *heteroscedasticity of the ratings* (due to the bounded scale of the star ratings). In contrast to a linear regression model, our proposed approach addresses these issues by estimating unbounded continuous preferences in a first step. Finally, *varying impacts of the aspect-based sentiments* over the rating scale might occur. For instance, in the context of a restaurant review, a poor service (e.g., due to an unfriendly waiter) may easily lead to assigning the lowest rating, but a pleasant service alone will in general not be sufficient to assign the highest rating. This can be taken into account

by generalizing the ordered probit model to allow varying coefficients of the aspect-based sentiments.

3.2 Generalized Ordered Probit Model to analyze Aspect-based Sentiments

We consider a set of $M \in \mathbb{N}$ textual reviews. Each review is associated with a star rating r on a discrete scale from 1 to a maximal rating of $K \in \mathbb{N}$. This is the common review structure observed for popular platforms such as *Amazon* or *TripAdvisor* (with K=5 or K=10 for most platforms). For each review, we take into account $A \in \mathbb{N}$ different item aspects relevant regarding the associated star rating. To give an example, in a restaurant review possible item aspects might be food quality or service quality. For instance, in the review "The food was great" a strongly positive sentiment towards the aspect food quality is expressed. More generally, we analyze the sentiment $s_a \in \mathbb{R}$ towards each item aspect $a \in 1, \ldots, A$. In this way, a numerical value is assigned to the sentiment s_a. Overall, for review i (with $i \in \{1, \ldots, M\}$) this results in aspect-based sentiments $s_1^i, \ldots, s_A^i \in \mathbb{R}$ and an associated star rating $r^i \in \{1, \ldots, K\}$.

In our two-step approach, first, preferences $R_*^i \in \mathbb{R}$ are modelled using the aspect-based sentiments $s_1^i, , ., s_A^i$. Later, the preferences are transformed into ratings in a non-linear way. According to the classical ordered probit model, the underlying preferences are given by

$$R_*^i = \beta_1 s_1^i + \ldots + \beta_A s_A^i + \epsilon, \quad (1)$$

where $\beta_1, \ldots, \beta_A$ denote the parameters with respect to the aspect-based sentiments $s_1^i, \ldots, s_A^i$ and $\epsilon \sim N(0,1)$ denotes the random error term of the underlying linear preference model reflecting the ambiguity contained in textual reviews (Mudambi et al., 2014). To account for the uncertainty stemming from the error term, we also introduce a discrete random variable $R^i \in \{1, \ldots, K\}$ to estimate the actual rating r^i in the i-th review. In the underlying linear preference model, the intercept term can be omitted since flexible threshold terms $\theta_1 < \ldots < \theta_{K-1} \in \mathbb{R}$ are used to transform the preferences into ratings, i.e., $R^i = 1$ for $R_*^i \leq \theta_1, R^i = 2$ for $\theta_1 < R_*^i \leq \theta_2, \ldots, R^i = K$ for $R_*^i > \theta_{K-1}$.

The parameters $\beta_1, \ldots, \beta_A$ and the thresholds $\theta_1, \ldots, \theta_{K-1}$ have to be estimated according to the classical ordered probit model. To give an example, consider a set of restaurant reviews on a rating scale from 1 to 5 addressing only the sentiments towards food quality and service. Then, an exemplarily resulting model might be given by the preference model $R_*^i = 1.0 \cdot s_{food}^i + 0.5 \cdot s_{service}^i + \epsilon$ (i.e., $\beta_1 = 1.0$ and $\beta_2 = 0.5$) and the non-linear transformation $R^i = 1$ if $R_*^i \leq -2.5 (= \theta_1), R^i = 2$ if $-2.5 < R_*^i \leq -0.8 (= \theta_2), \ldots, R^i = 5$ if $R_*^i > 3.2 (= \theta_4)$ onto the rating scale.

Those parameters are fitted by maximizing the log-likelihood of the model. According to the preference model in Equation (1) and the transformation onto the rating scale as introduced above, it is given by

$$\log L(\beta_1,..,\beta_A,\theta_1,..,\theta_{K-1}) = \sum_{i=1}^{M}\sum_{j=1}^{K} Z_{ij} \log[\Phi(\theta_j - \beta_1 s_1^i - \ldots - \beta_A s_A^i) - \Phi(\theta_{j-1} - \beta_1 s_1^i - \ldots - \beta_A s_A^i)], \quad (2)$$

where $Z_{ij} = 1$ if $r^i = j$, $Z_{ij} = 0$ otherwise, $\theta_0 := -\infty$, $\theta_K := +\infty$ and Φ denotes the cumulative distribution function of the standard normal distribution. That is, the likelihood of a rating j in the i-th review is given by $P(R^i = j) = P(R^i \leq j) - P(R^i \leq j-1)$ in the model, which means, by the difference in the cumulative probability to the next lowest rating.

In Equation (2), $P(R^i \leq j \,|s_1^i,\ldots,s_A^i) = P(R_*^i \leq \theta_j | s_1^i,\ldots,s_A^i) = \Phi(\theta_j - \beta_1 s_1^i - \ldots - \beta_A s_A^i)$ is assumed. In other words, the parameters $\beta_1,\ldots,\beta_A$ are independent of the rating value j ('Parallel Lines Assumption'). However, for example, a positive price-sentiment towards an item may have different impacts: Its impact might be stronger when the rating is at least mediocre on a rating scale from 1 to 5 (i.e., on $P(R^i \geq 3) = 1 - P(R^i \leq 2)$), whereas it might be lower when the associated rating is very good (i.e., on $P(R^i = 5) = 1 - P(R^i \leq 4)$). More generally, the Parallel Lines Assumption has to be tested for each aspect-based sentiment $s_a \in \{s_1,\ldots,s_A\}$. If it does not hold for s_a, a relaxed version

$$P(R^i \leq j \,|s_1^i,\ldots,s_A^i) = P(R_*^i \leq \theta_j | s_1^i,\ldots,s_A^i) = \Phi(\theta_j - \ldots - \beta_a^j s_a^i - \ldots) \quad (3)$$

with different coefficients β_a^j has to be used.

To test the Parallel Lines Assumption for sentiment s_a, the Bayesian Information Criterion (BIC) can be used (Schwarz, 1978). The assumption holds if $\log(M)\,(K-2) > 2 \cdot \left(\log L(\widehat{\beta_{G_a}}, \widehat{\theta_{G_a}}) - \log L(\hat{\beta},\hat{\theta})\right)$, where $\hat{\beta}$, $\hat{\theta}$ and $\widehat{\beta_{G_a}}$, $\widehat{\theta_{G_a}}$ are the maximum likelihood estimates for the classical version and the relaxed version G_a for sentiment s_a, respectively. Since the BIC takes into account the sample size M, it copes with the problem that for large samples, the model with more degrees of freedom often "falsely" gives distinctly higher likelihoods due to overfitting. As the sample sizes for the analysis of textual reviews are typically very high, the BIC is generally a well-suited measure in this context. Overall, our ordered probit model is generalized to varying coefficients for every aspect-based sentiment violating the Parallel Lines Assumption.

3.3 Measure to assess the Explanatory Power for the proposed Model

In the following, we propose a measure to assess the explanatory power of different aspect-based sentiments for our generalized ordered probit model. To do so, we assess the explained variability by different aspect-based sentiments in the underlying linear preference model. Thereby, the variability explained by the underlying preference model (i.e., its R-squared value) can be identified with its likelihood. More precisely, in this case the R-squared value can be evaluated by

$$\mathcal{R}^2 = 1 - \left[\frac{L_{Null-Model}}{L_{Preference-Model}}\right]^{2/M}, \tag{4}$$

where $L_{Preference-Model}$ denotes the likelihood of the fitted preference model (Maddala, 1983). Similarly, $L_{Null-Model}$ denotes the likelihood of a preference model restricted to $\beta = 0$. That is, the null model yields a constant preference regardless of the aspect-based sentiments. For the proposed generalized model, this identification of the R-squared value matches the explained variability in each preference model for $R^i \leq j$ and thus provides a well-founded overall estimate of the variability explained by the underlying generalized preference model.

However, our generalized ordered probit model for star ratings includes an additional variance, since the exact preferences underlying the assigned star ratings are unknown. That is, the likelihood of the underlying preference model is not directly accessible. To cope with this issue and to take into account the additional variance of the preference distribution, we propose to rescale the measure to have a maximum value of 1 for the generalized ordered probit model. In Nagelkerke (1991), this rescaling of the R-squared measure was already proposed for models that are fitted by maximum likelihood estimation in general, but in our generalized ordered probit model it is especially suited. Since our approach indeed includes underlying linear preference models, the measure inherits the precise foundation in Equation (4) when applied to our generalized ordered probit model. Overall, in our context the proposed measure is given by

$$\mathcal{R}^2_{Nagelkerke} = \frac{1-\left[\frac{L_{Null-Model}}{L_{Gen.Ord.Prob.-Model}}\right]^{2/M}}{1-L_{Null-Model}{}^{2/M}}. \tag{5}$$

This Nagelkerke pseudo R-squared measures how likely our generalized ordered probit model based on aspect-based sentiments is, compared to a null-model that does not factor in the aspect-based sentiments at all (i.e., restricting all coefficients of the aspect-based sentiments to zero in Equation (2)). In that way, it assesses the variability explained by the underlying preference model (Veall and Zimmermann, 1992). Having established an R-squared-type measure for the proposed model, we are able to evaluate the proposed model on different subsets of reviews and thereby gain valuable insights on the impact of different aspect-based sentiments.

4 Evaluation

In this section we evaluate our proposed model on a large dataset of restaurant reviews. First, we discuss the reasons for selecting the dataset and describe its preparation. Then, we methodically evaluate our approach in comparison to alternative models on our real-world dataset. Finally, we present the results of our proposed model for selected sentiment aspects.

4.1 Case Selection and Preparation of the Dataset

To evaluate our approach, we use a large real-world dataset of reviews for restaurants in New York City from 2010-2017 provided by an established web portal for online customer reviews

regarding local businesses, especially restaurants. Overall, the dataset consists of 2.4 million textual restaurant reviews and their associated star ratings. The characteristics of the dataset are summarized in Table 2. Thereby, the density of available reviews (calculated as the number of reviews divided by the product of the numbers of users and items) and the skewness of the rating distribution ('J-shaped') are in line with previous literature (e.g., Askalidis et al., 2017; Debortoli et al., 2016; Huang et al., 2004). Since these characteristics are typical for online customer reviews and since the dataset is large enough to analyze different sentiment aspects (each with a sufficient number of reviews), we selected this real-world dataset to apply and evaluate our model.

First, aspect-based sentiments have to be extracted from the reviews in the dataset. This step is necessary to apply the proposed generalized ordered probit model. However, it is not part of our contribution of the paper at hand (thus, it is described as dataset preparation). To extract sentiments from text, well-established methods exist (Agarwal et al., 2015; Liu, 2012; Taboada et al., 2011). Thereby, supervised learning approaches and dictionary-based approaches can be distinguished (Liu, 2012). Since supervised learning approaches require manual labelling of a large number of reviews, we decided to use a dictionary-based approach as in (Taboada et al., 2011). It is, however, important to note that generally supervised learning approaches may also be used to determine the inputs for our proposed model. For our evaluation, we applied separate sentiment dictionaries for different aspects in the restaurant context. This allowed us to account for varying sentiment orientations depending on the referred aspect. For example, the word "low" has a positive sentiment when referring to the price, whereas its sentiment orientation is negative for other aspects (e.g., "low food quality").

For our evaluation and without any loss of generality, we considered the aspects price, service, food quality, ambience, food quantity and miscellaneous. These aspects are broadly consistent with literature (e.g., Kiritchenko et al., 2014), but generally, additional aspects or separations (such as food quality vs. food quantity) may also be included as inputs for our model. To account for these different aspects in our analysis, we determined the referred aspect for every word expressing a sentiment in the reviews. Therefore, we used a list of index words for each considered aspect. Then, we applied the Stanford NLP Dependency Parser (Schuster and Manning, 2016), as in Kiritchenko et al. (2014) and Agarwal et al. (2015), to match sentiment words appearing in the review texts with the corresponding index words. For example, in the sentence "The *waitress* was *friendly*." The sentiment word *friendly* is matched with the index word *waitress*, which refers to the aspect service. Moreover, we aggregated the mean sentiment for each aspect accounting for intensified, weakened and negated contexts (Taboada et al., 2011). The implementation was done in Python. Finally, to avoid unstable results by an explanatory model, multicollinearity between the extracted aspect-based sentiments was tested to be sufficiently low. This is underlined by a variance inflation factor (VIF) of 1.12 in Table 2 (i.e., max. 1-1/1.12=11% of an aspect-based sentiment can be explained by sentiments towards other aspects) (Mansfield and Helms, 1982; O'brien, 2007).

Characteristics of the dataset	
# of users / restaurants	583’815 / 18’507
# of textual reviews and ratings	2’396’643
# of users with high review count (>50)	5’146
# of restaurants with high review count (>100)	6’197
Considered aspect-based sentiments	price, service, food quality, ambience, food quantity, and miscellaneous
Multicollinearity between the aspect-based sentiments measured by the VIF	1.12

Table 2. Characteristics of the Dataset

4.2 Methodical Evaluation of our Approach

Having prepared the dataset, the sentiments $s_1^i, \ldots, s_6^i$ (towards price, service, food quality, ambience, food quantity, miscellaneous) and the associated rating r^i ($\in \{1,\ldots,5\}$) are given for each review ($\in \{1, \ldots, 2'396'643\}$). Based on this real-world dataset, we evaluate the ability of different approaches to address the methodical issues discussed in Section 3.1. More precisely, we compare the ordered probit model and its proposed generalized version to a linear regression model because the latter is commonly used in literature to model and explain star ratings (cf. Section 2).

For the classical ordered probit model we get, according to Equation (1), the preference model

$$R_*^i = \beta_1 s_1^i + \ldots + \beta_6 s_6^i + \epsilon,$$

with $\epsilon \sim N(0,1)$ and the strictly non-linear transformation onto the rating scale

$$R^i = 1 \text{ for } R_*^i \leq \theta_1, R^i = 2 \text{ for } \theta_1 < R_*^i \leq \theta_2, \ldots, R^i = 5 \text{ for } R_*^i > \theta_4.$$

The proposed generalized ordered probit model can formally be written as

$$R^i \leq j \text{ if } \beta_1^j s_1^i + \beta_2^j s_2^i + \ldots + \beta_6^j s_6^i + \epsilon \leq \theta_j \text{ for } j = 1,2,3,4$$

with $R^i \in \{1, \ldots, 5\}$, $\epsilon \sim N(0,1)$ and different coefficients $\beta_a^1, \beta_a^2, \beta_a^3, \beta_a^4$ instead of one fixed coefficient β_a for the aspect-based sentiments $s_a \in \{s_1, \ldots, s_6\}$ that violate the Parallel Lines Assumption.

Using a linear regression the ratings are modelled as

$$R^i = \theta_0 + \beta_1 s_1^i + \ldots + \beta_6 s_6^i + \epsilon$$

with an intercept θ_0 and an error term $\epsilon \sim N(0, \sigma^2)$.

As already discussed in Section 3.1., these models differ in their ability to address the methodical issues for modelling star ratings with respect to aspect-based sentiments. In the following, we evaluate these three models regarding the four methodical issues discussed in Section 3.1:

First, we examined whether *uneven distances within the (ordinal) rating scale* exist on the dataset. Therefore, we analyzed the overall sentiment (defined as $s_1 + s_2 + \ldots + s_6$) of each review in the dataset. More precisely, we determined the average value of the overall sentiment $s_1 + s_2 + \ldots + s_6$ over all reviews grouped by the assigned star rating. Having determined these values, the distance between two star ratings can be identified with the difference in the average overall sentiment expressed in the corresponding reviews. Thereby, for example, the increase in this value from a 4-star to a 5-star review was detected to be less than half compared to all other adjacent star ratings. More precisely, the standardized differences (to have an average value of 1) in the overall sentiments amount to 1.1 (1 to 2 stars), 1.3 (2 to 3 stars), 1.1 (3 to 4 stars) and only 0.5 (4 to 5 stars). In that line, indeed uneven distances can be detected on our dataset. Thus, the assumption of even distances within the (ordinal) rating scale made by the linear regression model is not met. In contrast, the classical and the generalized ordered probit model can cope with uneven distances by assigning preference intervals of different sizes to the ratings.

Second, we determined whether a *non-normal distribution of the rating errors* occurs on our dataset. Therefore, we performed a Kolmogorov-Smirnov test (Massey, 1951) of the normality assumption for the linear regression model, which failed on the dataset. The test gave a vanishing probability that the cumulative distribution of the error term stems from a normal distribution ($p < 10^{-16}$). Hence, the assumption of normally distributed errors made by the linear regression model is not valid. In contrast, the (generalized) ordered probit models do not assume a specific distribution of the rating errors.

Third, we examined whether *heteroscedasticity of the ratings* is an issue in our dataset. This can be detected by comparing the linear model to a relaxed version with a scalable error variance $\beta_v \widehat{R^i}$ instead of a fixed error variance σ^2, where β_v denotes the additional variance parameter and $\widehat{R^i}$ the estimated rating. Adding the variance parameter β_v leads to an improvement of $3'620$ in the BIC which reveals the presence of heteroscedasticity. Hence, the assumption of homoscedasticity of the rating in the linear regression model is not met. In contrast, the classical and the generalized ordered probit model are not hampered by such an assumption and thus can handle the occurring heteroscedasticity of the ratings.

Finally, we tested whether *varying impacts of the aspect-based sentiments* can be detected in our dataset. To uncover possible varying impacts, we compared (similarly to above) the differences within the rating scale, but separately for different aspect-based sentiments. Thereby, for instance, the standardized differences in the service sentiment amount to 1.5 (1 to 2 stars), 1.1 (2 to 3 stars), 0.9 (3 to 4 stars) and 0.5 (4 to 5 stars). This indicates that the aspect-based sentiments indeed have significantly varying impacts since, for instance, the service sentiment differs over-proportionally between 1- and 2-star ratings (1.5 vs. distance 1.1 overall, as detected by analyzing overall uneven distances above). That is, a model assuming a constant coefficient for each aspect-based sentiment, such as the linear regression model, is strongly limited in its validity. To verify that our proposed model captures these different impacts, we also compared

the classical ordered probit model to a generalized version by the respective BIC values. Thereby, also significant varying impacts over the rating scale were detected by a difference in the BIC value of 2'686. Since Raftery (1995) defined differences bigger than 10 already as 'very strong evidence' for the model with the lower BIC value, the proposed generalized version is more valid.

Overall, the methodical evaluation above shows that indeed all of the methodical issues discussed in Section 3.1 occur on our dataset. Our proposed model is able to address these issues, whereas the classical ordered probit model does not account for varying impacts of the aspect-based sentiments and the linear regression model does not resolve any of the discussed issues. Table 3 summarizes the results.

	Accounts for uneven distances within the (ordinal) rating scale	**Allows for non-normal distribution of the rating errors**	**Accounts for heterosce-dasticity of the ratings**	**Accounts for varying impacts of the aspect-based sentiments**	**BIC (relative to the Generalized Ordered Probit Model)**
Ordered Probit Model	✓	✓	✓	n/a (constancy assumed)	2'686
Generalized Ordered Probit Model	✓	✓	✓	✓	-
Linear Regression Model	n/a (even dis-tances as-sumed)	n/a (normal dis-tribution as-sumed)	n/a (homosce-dasticity as-sumed)	n/a (constancy assumed)	39'029
Empirical evidence for methodical issues in our dataset	Related standardized differences are significantly uneven (from 0.5 to 1.3)	Kolmogorov-Smirnov test rejects normal distribution assumption ($p < 10^{-16}$)	Additional variance parameter in linear model leads to a more valid model (i.e. higher BIC)	Impacts of certain senti-ments (e.g., service senti-ment) differ significantly between 1- and 2-star ratings	

Table 3. Comparison of different Regression Models on the Dataset

To further evaluate the considered regression models, we also compared the relative quality of these models. Therefore, the values of the BIC relative to the generalized ordered probit model are also stated in Table 3. As mentioned in Section 3.2, the BIC accounts for the sample size and thus is suited for large datasets such as our dataset of restaurant reviews. Thereby, the values of the BIC indicate that the proposed generalized ordered probit model is methodically much better suited to explain the star ratings on our dataset than a classical ordered probit and especially a linear regression model.

4.3 Results for selected Aspect-based Sentiments

In this section, we present and discuss the results for selected aspect-based sentiments based on our dataset. Since our main contribution addresses methodical issues on explaining the star ratings and due to length restrictions, we limited ourselves to the (three most frequently referred) aspect-based sentiments for price, service and food quality.

At first, we built our model based on the subset of reviews expressing sentiments towards all three of these aspects (and do not address any of the other extracted aspects). Price, service and food quality sentiment have different impacts over the rating scale (i.e., violate the Parallel Lines Assumption), which underlines the relevance of our proposed generalized ordered probit model. This model is given by

$$R^i \leq j \text{ if } \beta^j_{price} s^i_{price} + \beta^j_{service} s^i_{service} + \beta^j_{food} s^i_{food} + \epsilon \leq \theta_j \text{ for } j = 1,2,3,4.$$

Coefficients	$j = 1$	$j = 2$	$j = 3$	$j = 4$
θ_j (threshold)	-0.89	-0.07	0.78	1.59
β^j_{price} (price sentiment)	0.11	0.16	0.17	0.11
$\beta^j_{service}$ (service sentiment)	0.28	0.29	0.28	0.22
β^j_{food} (food quality sentiment)	0.29	0.38	0.43	0.34

Table 4. Coefficients in the Generalized Ordered Probit Model (based on Reviews that address the Price, Service and Food Quality Sentiment)

The coefficients of the model are provided in Table 4. The Nagelkerke pseudo R-squared (cf. Equation (5)) is 44%. All coefficients were highly statistically significant ($p < 10^{-7}$). Overall, we noticed that the price sentiment has a lower impact than the service or food quality sentiment. One reason might be that the price level of the restaurant is often known prior to the visit while the service quality and the food quality are experienced during the stay. We found that the food quality sentiment indeed has the strongest impact on the overall preferences (in terms of the assigned star ratings). Surprisingly though, the sentiment towards service has a similarly strong impact on the preferences for parts of the rating scale. In particular, in the lowest rating category, the service sentiment has an (almost) equally strong impact as the food quality sentiment (0.28 vs 0.29 in Table 4). This indicates that for poorly rated restaurants a lacking service quality is equally bad as a low food quality. Notably, we would not have been able to detect

that characteristic using a linear regression model or the classical ordered probit model. In the latter models, fixed coefficients are estimated, whereas in our proposed model the ratio of the coefficients for food quality sentiment and service sentiment vary significantly (for example, 0.29/0.28 = 1.04 for $j = 1$ vs. 0.34/0.22 = 1.55 for $j = 4$).

To further evaluate the benefits of the proposed model, we compared it to the classical ordered probit model, which does not allow for varying impacts. Thereby, the Nagelkerke pseudo R-squared is 42% for the ordered probit model. That is, on a general level the explanatory power is nearly the same as for our model. However, to gain more detailed insights in the differences in validity, we analyzed how well the two models are able to explain the ratings for different parts of the rating scale. Thereby, the results might be biased by the skewed rating distribution ('J-shaped'). To eliminate this bias, we randomly sampled an equal number of 1,000 reviews for each rating category and built the models on this sample. Thereby, the proposed model had significantly higher explanatory power measured by the Nagelkerke pseudo R-squared for high and low ratings: In detail, 71% vs. 63% for rating 1, 38% vs. 27% for rating 2, 24% vs. 29% for rating 3, 45% vs. 44% for rating 4 and 69% vs. 63% for rating 5. Except of the average rating of 3, the proposed model explains the ratings more accurately for all rating categories. Overall, this indicates that the proposed generalized ordered probit model outperforms the classical ordered probit model by additionally addressing varying impacts of the aspect-based sentiments (cf. Table 3).

Besides the reviews addressing all three aspect-based sentiments, there are reviews which address only one or two of them. In that line different subsets of reviews can be identified (based on the addressed aspect-based sentiments). To examine and compare how well our proposed model is able to explain the overall star ratings on these subsets, we evaluated the respective Nagelkerke pseudo R-squared values given in Table 5. All coefficients in all models were highly statistically significant ($p < 10^{-7}$).

Aspects addressed in the reviews	Nagelkerke pseudo R-squared by our proposed model
Price / Service / Food	20% / 49% / 32%
Price & Service / Price & Food / Service & Food	48% / 35% / 43%
Price & Service & Food	44%

Table 5. Nagelkerke pseudo R-squared based on Reviews addressing different Sentiment Aspects

By establishing the Nagelkerke pseudo R-squared as an estimator for the explained variability in the generalized ordered probit model, we have an evaluation measure to compare the explanatory power of aspect-based sentiments on different subsets of reviews. That is, we have a replacement for the R-squared measure in a methodically sound model for star ratings and thus get a more valid comparison of the different subsets, compared to using a linear regression model.

Thereby, we found that for reviews addressing only one of these three sentiments, the service sentiment even explains the most variability in the star ratings (49% vs. 32% for the food quality sentiment and 20% for the price sentiment). This indicates that reviews only addressing the service often express a definite sentiment towards that aspect (e.g., complaints about unfriendly service). For reviews that address the food quality sentiment we found that the ones containing additional sentiment aspects explain the star ratings better (44% vs. 32% with food quality sentiment alone). This indicates that one-dimensional reviews towards the food quality often do not discuss additional aspects that influence their overall preference towards the restaurant. For reviews addressing the service sentiment we detected a different effect. The ones containing additional sentiments tended to explain the star ratings slightly worse in comparison (44% vs. 49% with service sentiment alone). According to the first observation, this might be due to the fact that reviews addressing mainly the service often express a definite sentiment towards that aspect which strongly affects the associated overall rating, while reviews addressing multiple aspects might only mention the service for the sake of completeness. Overall, using our approach we detected significantly varying impacts of the aspect-based sentiments both within the rating scale and (in terms of the explained variability) based on the combination of aspects addressed in the reviews.

5 Implications for Theory and Practice

In contrast to existing approaches for explaining the star ratings of online customer reviews, our approach takes advantage of the valuable information contained in aspect-based sentiments which are measured in the review texts. Furthermore, it addresses the methodical issues which emerge during the explanation of overall star ratings, particularly due to their ordinal scale. In that way, by applying our approach the impact of aspect-based sentiments on the associated overall star ratings can be explained and interpreted in a methodically well-founded way. Proposing a generalized ordered probit model and allowing the consideration of different sets of aspect-based sentiments, our approach can provide a detailed level of analysis. For example, as indicated by the evaluation of the model on a large real-world dataset of restaurant reviews, valuable insights about varying impacts of aspect-based sentiments on the overall star ratings can be discovered. Finally, having established a quality criterion for the proposed model in form of a R-squared type measure, our approach is able to compare the strength of the relationships between different sets of aspect-based sentiments and the associated overall star ratings.

From a methodical point of view, the results presented in Section 4 indicate that our proposed model is methodically better suited than a linear regression, which is commonly used in state-of-the-art approaches, as well as a classical ordered probit model to explain the overall star ratings of online customer reviews. This is especially supported by the fact that our dataset is large enough to be representative and exhibits characteristics typical for online customer reviews (cf. Section 4.1). Moreover, the methodical issues occurring when explaining overall star ratings (i.e., uneven distances within the (ordinal) rating scale, non-normal distribution of the

rating errors, heteroscedasticity of the ratings and varying impacts of the aspect-based sentiments; cf. Table 3) are addressed by our proposed approach. Overall, the relative BIC values in Table 3 are all significantly large, which indicates, that compared to the alternative models our approach is able to explain some additional substantial part of the variation of star ratings.

Our proposed model can help practitioners to gain a data-driven competitive advantage by using the aspects and the associated sentiments to analyze why specific customer ratings were assigned to their company or a competitor. Based on these insights the company can innovate its business model and further develop customer-centric solutions adding business value. This competitive advantage can be achieved in diverse areas of application such as decision support, quality management or marketing. Using the detailed information about the aspects influencing the customer assessment, businesses can focus their efforts on actions in a more effective and target-oriented way. For example, the use of financial, infrastructural and human resources can be improved with respect to customer demands. In quality management, our approach allows to identify reasons for a possible drop in customer satisfaction and thus enables suitable countermeasures. Using our approach, marketing analysts can study the reasons for customer (dis)satisfaction on a detailed level. This allows them to ensure customer orientation by considering client needs and meeting their major priorities.

6 Conclusion, Limitations and Future Work

Explaining the underlying reasoning for the overall star ratings of online customer reviews is an important issue in both research and practice. In this paper, we present an approach to explain and interpret the overall star ratings of online customer reviews using aspect-based sentiments contained in review texts. The proposed approach contributes to existing research by allowing for a detailed and interpretable understanding of the customer assessment of products and services. We propose a generalized ordered probit model and a Nagelkerke pseudo R-squared measure to explain the overall star ratings using aspect-based sentiments. Existing approaches lack such an explanatory regression model addressing the methodical issues associated with the star ratings and assessing the explanatory power for the model. A formal definition of the approach was provided, and it was evaluated on a large real-world dataset of restaurant reviews. The evaluation was conducted in two steps. In a first step, we methodically evaluated our approach by comparing the proposed model to alternative regression models on our dataset. Therein, we showed, that our approach is able to address the methodical issues occurring when explaining overall star ratings of online customer reviews. In a second step, we presented the results of our proposed model for selected sentiment aspects. Thereby, our approach yields interpretable results and detailed relationships between aspect-based sentiments and the overall star ratings have been uncovered.

Nevertheless, our work also has limitations which may constitute the starting point for future research. In this paper we focused on evaluating the explanatory power of given sets of aspect-

based sentiments. Future research could explore the usage of sentiments towards automatically extracted, but interpretable aspects. Furthermore, the approach was applied to a large real-world dataset from the restaurant domain. The fact that the dataset that our dataset is large and exhibits typical characteristics for online customer reviews suggests that the results will apply in other domains in a similar way. Nevertheless, future research could evaluate it on further datasets from other domains. Finally, further evaluations and methodical extensions (e.g., considering additional structural data such as given sub-ratings or item data) could also provide interesting insights regarding the explanation of star ratings of online customer reviews.

7 References

Agarwal, B., N. Mittal, P. Bansal and S. Garg (2015). "Sentiment analysis using common-sense and context information" *Computational intelligence and neuroscience* 2015, 715–730.

Archak, N., A. Ghose and P. G. Ipeirotis (2007). "Show me the money!: deriving the pricing power of product features by mining consumer reviews". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 56–65.

Archak, N., A. Ghose and P. G. Ipeirotis (2011). "Deriving the pricing power of product features by mining consumer reviews" *Management Science* 57 (8), 1485–1509.

Askalidis, G., S. J. Kim and E. C. Malthouse (2017). "Understanding and overcoming biases in online review systems" *Decision Support Systems* 97, 23–30.

Chevalier, J. A. and D. Mayzlin (2006). "The effect of word of mouth on sales: Online book reviews" *Journal of Marketing Research (JMR)* 43 (3), 345–354.

Clemons, E. K., G. G. Gao and L. M. Hitt (2006). "When online reviews meet hyperdifferentiation: A study of the craft beer industry" *Journal of Management Information Systems (JMIS)* 23 (2), 149–171.

Debortoli, S., O. Müller, I. A. Junglas and J. Vom Brocke (2016). "Text Mining for Information Systems Researchers: An Annotated Topic Modeling Tutorial" *Communications of the Association for Information Systems (CAIS)* 39, 110–135.

Filieri, R., S. Alguezaui and F. McLeay (2015). "Why do travelers trust TripAdvisor? Antecedents of trust towards consumer-generated media and its influence on recommendation adoption and word of mouth" *Tourism Management* 51, 174–185.

Fu, B., J. Lin, L. Li, C. Faloutsos, J. Hong and N. Sadeh (2013). "Why people hate your app: Making sense of user feedback in a mobile app store". In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1276–1284.

Ganu, G., N. Elhadad and A. Marian (2009). "Beyond the stars: improving rating predictions using review text content". In: *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, pp. 1–6.

Ganu, G., Y. Kakodkar and A. Marian (2013). "Improving the quality of predictions using textual information in online user reviews" *Information Systems* 38 (1), 1–15.

Ghose, A. and P. G. Ipeirotis (2011). "Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 23 (10), 1498–1512.

Goldberg, A. B. and X. Zhu (2006). "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization". In: *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pp. 45–52.

Greene, W. H. and D. A. Hensher (2010). *Modeling Ordered Choices. A Primer:* Cambridge University Press.

Guo, Y., S. Barnes and Q. Jia (2016). "Mining Meaning from Online Ratings and Reviews: Tourist Satisfaction Analysis Using Latent Dirichlet Allocation" *Tourism Management* 59, 467–483.

Hu, N., L. Liu and J. J. Zhang (2008). "Do online reviews affect product sales? The role of reviewer characteristics and temporal effects" *Information Technology and Management* 9 (3), 201–214.

Huang, Z., H. Chen and D. Zeng (2004). "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering" *ACM Transactions on Information Systems (TOIS)* 22 (1), 116–142.

ITU (2017). *World telecommunication/ICT indicators database:* International Communication Union.

Jo, Y. and A. H. Oh (2011). "Aspect and sentiment unification model for online review analysis". In: *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pp. 815–824.

Kiritchenko, S., X. Zhu, C. Cherry and S. Mohammad (2014). "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 437–442.

Li, F., N. Liu, H. Jin, K. Zhao, Q. Yang and X. Zhu (2011). "Incorporating reviewer and product information for review rating prediction". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pp. 1820–1825.

Linden, G., B. Smith and J. York (2003). "Amazon. com recommendations: Item-to-item collaborative filtering" *IEEE Internet Computing* (1), 76–80.

Linshi, J. (2014). *Personalizing Yelp star ratings: A semantic topic modeling approach.* Yelp Dataset Challenge Winner. URL: https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf (visited on 06/24/2019).

Liu, B. (2012). "Sentiment analysis and opinion mining" *Synthesis Lectures on Human Language Technologies* 5 (1), 1–167.

Liu, Y., T. Teichert, M. Rossi, H. Li and F. Hu (2017). "Big data for big insights: Investigating language-specific drivers of hotel satisfaction with 412,784 user-generated reviews" *Tourism Management* 59, 554–563.

Maddala, G. S. (1983). *Limited-dependent and qualitative variables in econometrics:* Cambridge University Press.

Mansfield, E. R. and B. P. Helms (1982). "Detecting Multicollinearity" *The American Statistician* 36 (3a), 158–160.

Massey, F. J. (1951). "The Kolmogorov-Smirnov Test for Goodness of Fit" *Journal of the American Statistical Association (JASA)* 46 (253), 68–78.

McKelvey, R. D. and W. Zavoina (1975). "A statistical model for the analysis of ordinal level dependent variables" *The Journal of Mathematical Sociology* 4 (1), 103–120.

Minnema, A., T. H. A. Bijmolt, S. Gensler and T. Wiesel (2016). "To keep or not to keep: effects of online customer reviews on product returns" *Journal of Retailing* 92 (3), 253–267.

Monett, D. and H. Stolte (2016). "Predicting Star Ratings based on Annotated Reviews of Mobile Apps". In: *Federated Conference on Computer Science and Information Systems (FedCSIS 2016)*, pp. 421–428.

Mudambi, S. M., D. Schuff and Z. Zhang (2014). "Why aren't the stars aligned? An analysis of online review content and star ratings". In: *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS 2014)*, pp. 3139–3147.

Myers, R. H. (1990). *Classical and modern regression with applications:* Duxbury Press.

Nagelkerke, N. J. D. (1991). "A Note on a General Definition of the Coefficient of Determination" *Biometrika* 78 (3), 691–692.

O'brien, R. M. (2007). "A Caution Regarding Rules of Thumb for Variance Inflation Factors" *Quality & Quantity* 41 (5), 673–690.

O'Mahony, M. P. and B. Smyth (2010). "Using readability tests to predict helpful product reviews". In: *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pp. 164–167.

Pang, B. and L. Lee (2005). "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pp. 115–124.

Phillips, P., S. Barnes, K. Zigan and R. Schegg (2017). "Understanding the impact of online reviews on hotel performance: an empirical analysis" *Journal of Travel Research* 56 (2), 235–249.

Qiu, J., C. Liu, Y. Li and Z. Lin (2018). "Leveraging sentiment analysis at the aspects level to predict ratings of reviews" *Information Sciences* 451, 295–309.

Qu, L., G. Ifrim and G. Weikum (2010). "The bag-of-opinions method for review rating prediction from sparse text patterns". In: *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 913–921.

Radojevic, T., N. Stanisic and N. Stanic (2017). "Inside the rating scores: A multilevel analysis of the factors influencing customer satisfaction in the hotel industry" *Cornell Hospitality Quarterly* 58 (2), 134–164.

Raftery, A. E. (1995). "Bayesian model selection in social research" *Sociological Methodology*, 111–163.

Sainani, K. L. (2014). "Explanatory versus predictive modeling" *PM&R* 6 (9), 841–844.

Schouten, K. and F. Frasincar (2016). "Survey on aspect-level sentiment analysis" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 28 (3), 813–830.

Schuster, S. and C. D. Manning (2016). "Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 23–28.

Schwarz, G. (1978). "Estimating the dimension of a model" *The Annals of Statistics* 6 (2), 461–464.

Sharma, R. D., S. Tripathi, S. K. Sahu, S. Mittal and A. Anand (2016). "Predicting online doctor ratings from user reviews using convolutional neural networks" *International Journal of Machine Learning and Computing* 6 (2), 149.

Shmueli, G. (2010). "To explain or to predict?" *Statistical Science* 25 (3), 289–310.

Shmueli, G. and O. R. Koppius (2011). "Predictive analytics in information systems research" *MIS Quarterly*, 553–572.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll and M. Stede (2011). "Lexicon-Based Methods for Sentiment Analysis" *Computational Linguistics* 37 (2), 267–307.

Veall, M. and K. Zimmermann (1992). "Pseudo-R2's in the ordinal probit model" *The Journal of Mathematical Sociology* 16 (4), 333–342.

Xiang, Z., Z. Schwartz, J. H. Gerdes Jr and M. Uysal (2015). "What can big data and text analytics tell us about hotel guest experience and satisfaction?" *International Journal of Hospitality Management* 44, 120–130.

Yang, Y., Z. Mao and J. Tang (2018). "Understanding guest satisfaction with urban hotel location" *Journal of Travel Research* 57 (2), 243–259.

Ye, Q., R. Law and B. Gu (2009). "The impact of online user reviews on hotel room sales" *International Journal of Hospitality Management* 28 (1), 180–182.

Ye, Q., R. Law, B. Gu and W. Chen (2011). "The influence of user-generated content on traveler behavior: An empirical investigation on the effects of e-word-of-mouth to hotel online bookings" *Computers in Human Behavior* 27 (2), 634–639.

Ye, Q., H. Li, Z. Wang and R. Law (2014). "The influence of hotel price on perceived service quality and value in e-tourism: An empirical investigation based on online traveler reviews" *Journal of Hospitality & Tourism Research* 38 (1), 23–39.

Zhou, L., S. Ye, P. L. Pearce and M.-Y. Wu (2014). "Refreshing hotel satisfaction studies by reconfiguring customer review data" *International Journal of Hospitality Management* 38, 1–10.

Zhu, F. and X. Zhang (2010). "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics" *Journal of Marketing* 74 (2), 133–148.

Zhu, J., H. Wang, M. Zhu, B. K. Tsou and M. Ma (2011). "Aspect-based opinion polling from customer reviews" *IEEE Transactions on Affective Computing* 2 (1), 37–49.

4 Automated Planning of Process Models

This section contains three papers covering automated planning of process models, the third focal point of the dissertation, including the research questions RQ6-RQ8. In Section 4.1, the automated construction of the control flow patterns parallel split and synchronization is treated (RQ6). Section 4.2 comprises an automated planning approach for adapting process models to needs for change in advance (RQ7). Finally, Section 4.3 discusses the construction of multi-actor process models based on an automated planning approach (RQ8).

4.1 Paper 6: Automated Planning of Process Models: The Construction of Parallel Splits and Synchronizations

Current Status	Full Citation
accepted and published (10/2019) in Issue 125 of *Decision Support Systems*	Heinrich, B., F. Krause and A. Schiller (2019). "Automated Planning of Process Models: The Construction of Parallel Splits and Synchronizations". *Decision Support Systems (DSS)* 125, Paper ID 113096.

Summary

This paper addresses RQ6 by providing an automated planning approach for the construction of parallel splits and synchronizations. To this end, concepts which allow the identification of the set of feasible parallelizations including complex parallelizations (e.g., nested parallelizations and parallelizations with an arbitrary length of path segments) are proposed. Additionally, a concrete algorithm enabling the actual automated construction of parallel splits and synchronizations in process models is introduced. The approach is evaluated according to key properties such as completeness, correctness and computational complexity. Furthermore, its practical applicability as well as its practical utility are verified. To do so, the approach is applied in a project at a financial services provider in a naturalistic ex post evaluation as well as within several real-world processes of different companies in various contexts.

The work builds heavily on concepts from AI planning, for instance, belief state tuples and belief states, nondeterministic belief state-transition systems, applicability and planning graphs, which are subsumed under the notion "planning domain". In particular, planning graphs can be constructed using methods from AI planning. On this foundation, the paper develops novel concepts in form of definitions (defining dependencies between actions) and theorems (stating how these dependencies relate to parallelizations) as well as a novel method in form of an algorithm. Enabling the automated construction of parallel splits and synchronizations, the paper furthers business process agility. Moreover, in multiple real-world scenarios, applying the approach leads to additional, feasible parallelizations being constructed (in comparison to manual planning). Consequently, business process flexibility is increased. Additionally, the constructed parallelizations enhance the decision-making aspect of process models by allowing to select a beneficial way for process execution (e.g., based on temporal, economic and resource criteria constraints).

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style.

The paper as published by Elsevier is available at: https://doi.org/10.1016/j.dss.2019.113096

Abstract:

Efficient business processes play a major role in the success of companies. Business processes are captured and described by models that serve, for instance, as a starting point for implementing processes in a service-oriented way or for performance analysis. To support process modelers via methods and techniques (e.g., algorithms) in an automated manner, several research fields such as process mining and automated planning of process models have emerged. In particular, the aim of the latter research field is to enable the automated construction of process models using planning techniques. To this end, an automated construction of control flow patterns in process models is necessary. However, this task currently remains a widely unsolved issue for the central patterns parallel split and synchronization.

We introduce novel concepts, which, in contrast to existing approaches, allow the construction of complex parallelizations (e.g., nested parallelizations and parallelizations with an arbitrary length of path segments) and are able to identify the set of feasible parallelizations. Moreover, we propose an algorithm facilitating the automated construction of parallel splits and synchronizations in process models. Our approach is evaluated according to key properties such as completeness, correctness and computational complexity. Furthermore, both the practical applicability within several real-world processes of different companies in various contexts as well as the practical utility of our approach are verified. The presented research expands the boundaries of automated planning of process models, adds more analytical rigor to automatic techniques in the context of business process management and contributes to control flow pattern theory.

Keywords: business process modeling, automated planning of process models, control flow patterns, business process management

1 Introduction

The way a company defines and handles its business processes is of paramount importance for the company's success; this has been acknowledged in both science and practice over the previous years (Chang, 2016; vom Brocke and Mendling, 2018) and has started and stimulated research fields such as business process management (BPM). A business process can be defined as the "specific ordering of work activities across time and space, with a beginning, an end, and clearly identified inputs and outputs" (Davenport, 1993, p. 5). BPM focuses on capturing, implementing, analyzing and optimizing a company's business processes. In this regard, several research fields in BPM such as process mining (Augusto et al., 2018; IEEE Task Force on Process Mining, 2012; van der Aalst, 2016), automated (web) service selection and composition (Lemos et al., 2016; Xu et al., 2016) and automated planning of process models (Heinrich et al., 2018; Hoffmann et al., 2012) have emerged in order to support business analysts and process modelers via methods and algorithms. In particular, the focus in this paper lies on the research

field automated planning of process models, which aims to enable the automated construction of process models using planning algorithms (Heinrich et al., 2015; Heinrich et al., 2018; Heinrich and Schön, 2015, 2016; Henneberger et al., 2008; Hoffmann et al., 2012; Lautenbacher et al., 2009).

The automated construction of process models can be understood as a planning problem (Ghallab et al., 2004) with the objective to arrange process model components in a feasible order based on an initial state, a set of available actions as well as conditions for goal states. The input data for this planning can, for instance, be obtained by fresh modeling of actions, extracting actions from existing process models or a conceptualization of (web) services to represent the corresponding actions (Bortlik et al., 2018). Furthermore, interfaces of process modeling tools may be used (cf. *Evaluation*). A fundamental challenge of the automated construction of process models is to cope with control flow patterns describing the control flow of a process. More precisely, in order to plan sophisticated process models, not only a specific sequence of actions but also the control structures representing these patterns have to be constructed in an automated manner.

This general problem of planning an entire process model including control flow patterns is decomposed into sub problems to address a sub problem in-depth. Parallel splits (sometimes also called AND-splits) and their corresponding synchronizations capture elementary aspects of processes and thus are assessed to be central patterns (Soffer et al., 2015; van der Aalst et al., 2003; van der Aalst and ter Hofstede, 2005). Parallelizations are also deemed highly relevant when aiming to represent complex process flows (cf., e.g., examples in (He et al., 2008; Russell et al., 2016) and the discussion below). Furthermore, uncovering and representing the concurrent behavior of a system has long been assessed as valuable in many application contexts (Cheikhrouhou et al., 2015; Cook and Wolf, 1998) and parallelizations are crucial, for instance, to reduce execution times of processes and service compositions (Alrifai et al., 2012). Besides the relevance discussed by researchers, in several projects with different companies, we observed that almost all of the processes incorporated many parallelized actions. For example, in a cooperation with a European financial services provider in which over 600 core business processes were analyzed, over 90% of these processes contained at least one parallelization while around 33% contained more than five. Our analyses of these processes showed that the parallelizations served different reasons such as reducing total required execution time, increasing throughput and allowing a relatively constant workload of employees and a high utilization of resources (due to the reduction of waiting time). In this vein, parallelizations offer valuable decision support, as parallelizations enhance the decision-making aspect of process models (Hasić et al., 2018; Kummer et al., 2016): They allow to select a beneficial way for process execution (e.g., in terms of execution time). Moreover, in some cases, they were necessary to ensure legal and regulatory compliance (e.g., to realize a dual principle). Furthermore, they improved organizational flexibility. For instance, they enabled a concurrent process execution by different organizational units and, due to reduced execution times, a quicker response to

external events (e.g., customer complaints). This illustrates the practical importance of parallelizations in process models.

Addressing both the scientific and practical relevance, in this paper we will concentrate on the so far widely unsolved issue of an automated construction of parallel splits and synchronizations in process models. The contributions are as follows:

- Concepts are developed allowing the construction of complex parallelizations (including nested parallelizations and an arbitrary length of path segments within parallelizations) and the set of all feasible parallelizations while not constructing infeasible parallelizations. These concepts are independent of a concrete modeling language and can cope with possibly infinite sets of world states and large domains. This guarantees a maximum of compatibility with existing approaches and process modeling languages.
- Based on these concepts, we propose a novel algorithm for the automated construction of parallel splits and synchronizations in process models.
- The presented algorithm is implemented into a prototype which is evaluated in real-use situations.

The remainder of the paper is organized as follows: The next section contains the background of our research. Here, the theoretical background, the related work and the underlying planning domain are presented. Thereafter, we answer the key research question of how parallel splits and synchronizations can be constructed in an automated manner by proposing concepts and providing a concrete algorithm. The approach is illustrated by means of a running example. In the subsequent section, the concepts and the algorithm are evaluated according to key properties such as completeness, correctness and computational complexity. Furthermore, they are implemented into a prototype and their practical applicability within several real-world processes of different companies in various contexts as well as their practical utility are assessed. Finally, the last section summarizes the results, discusses limitations and provides an outlook for future research.

2 Background

In this section, we describe the theoretical background of our research based on the discussion by Soffer et al. (2015) and present related work and the research gap. Thereafter, we outline the underlying planning domain.

2.1 Theoretical Background

Business process models are critical when designing, realizing and analyzing business processes (Reijers and Mendling, 2011; Russell et al., 2016; van der Aalst, 2013; vom Brocke and Mendling, 2018). Imperative models representing business processes usually consist of at least two types of components: actions and control flow patterns. These control flow patterns can be

seen as a theory for clarifying the process flow, with a control flow pattern being a proposition which expresses how processes can be executed, or, more precisely, which control flows can exist in processes (Russell et al., 2006; van der Aalst and ter Hofstede, 2005). On the one hand, control flow patterns are abstract concepts striving to show the process flow independently of a concrete modeling language; on the other hand, modeling languages provide a concrete representation for control flow patterns (van der Aalst and ter Hofstede, 2005). The basic control flow patterns are sequence, exclusive choice, simple merge, parallel split and synchronization (Migliorini et al. 2011; Russell et al., 2006; Russell et al., 2016; van der Aalst et al., 2003). Control flow patterns allow to abstract from an individual process execution: In this regard, a parallel split specifies that a single route of execution is split into two or more sequences of actions (called 'path segments'), where all actions in these different path segments can be executed concurrently (Russell et al., 2006; Russell et al., 2016; van der Aalst et al., 2003). However, the actions in different path segments originating from a parallel split do not necessarily have to be executed in parallel from a temporal perspective (van der Aalst et al., 2003), although it is generally feasible to do so. Further, a synchronization represents a point where two or more path segments of arbitrary length originating from previous parallel splits converge into a single subsequent path (Russell et al., 2016). This conceptualization regarding parallel splits and synchronizations also holds for so called nested parallelizations. Such a nested parallelization occurs when one or more parallelizations and their corresponding actions are contained in a path segment of another parallelization.

To further substantiate this conceptualization, the process state (denoted by its state variables; cf. Definition 1 in *Planning Domain*) has to be considered. In this way, potential inconsistencies can be avoided, ensuring the feasibility of parallel splits, synchronizations and their state transitions (cf., e.g., Wang and Kumar, 2005). The well-known ACID properties (Haerder and Reuter, 1983) serve as reference to address this feasibility. More precisely, a synchronization merging two or more path segments (originating from a previous parallel split) requires that all actions in these path segments have been executed (Russell et al., 2016), while conflicts have to be avoided. For instance, when the same state variables are changed concurrently in different path segments, this represents a violation to the ACID-principle isolation, thus creating a conflict when trying to synchronize the path segments and their resulting states. In detail, while due to the potential concurrency of path segments leading to a synchronization, different actual execution routes are enabled (e.g., due to different possible temporal orders of actions), all of these routes need to result in the same state when synchronized. This holds due to two reasons: First, the state before the parallel split is equal. Second, it is necessary to be able to continue with the process independently of the actual execution route taken before synchronization (Soffer et al., 2015). Furthermore, as processes may be executed many times with different initial states, both control flow patterns as well as states (and its state variables) denoted by a process model should be able to deal with possibly infinite sets of world states and large domains as well as respective data types used by the state variables (Heinrich et al., 2015).

Based on these theoretical considerations with regard to control flow patterns, and in particular parallelizations, much work has been carried out to analyze control flow patterns in terms of different aspects such as inclusion in workflow modeling languages and corresponding tools (e.g., Russell et al., 2016), reconstruction of control flow in processes via process mining (e.g., Wen et al., 2009), empirical evidence and applications in real-world processes (e.g., Russell et al., 2006), and automated verification of control flow (patterns) (e.g., Wynn et al., 2009). In the same vein, approaches for the automated planning of process models can also be seen as contribution to control flow pattern theory by analyzing and evaluating whether control flow patterns can be constructed correctly in an automated manner. Based upon this, sequences of actions as well as control flow patterns can be constructed in order to plan sophisticated process models. To this end, concepts and algorithms for the automated construction of control flow patterns need to be provided. In this paper, we contribute to this research by presenting concepts and an algorithm that constructs both parallel splits and synchronizations in an automated manner while considering the theoretical conceptualization of parallelizations discussed above.

2.2 Related Work and Research Gap

We structure existing approaches for the automated identification or construction of parallelizations according to the BPM lifecycle phases process modeling, process implementation, process execution and process analysis (Wetzstein et al., 2007). While our research focuses on the *process modeling* phase, we have also included relevant approaches from other phases, as such approaches may possibly be interesting.

In the *process modeling* phase, so far only the approach of Hoffmann et al. (2012) discusses the automated construction of process models including parallelizations. However, the authors do not aim to provide concepts of how to construct parallelizations and do not present a concrete algorithm for the construction of parallelizations. Moreover, they use a heuristic approach in model-based software development, and thus their approach does not provide all feasible parallelizations.

Automated web service composition can be seen as part of the phases *process implementation* and *process execution* and is partly based on planning techniques (Bertoli et al., 2001; Bonet and Geffner, 2001; Deokar and El-Gayar, 2011). Heinrich et al. (2012) analyze multiple approaches (Bertoli et al., 2006; Bertoli et al., 2010; Binder et al., 2009; Constantinescu et al., 2004; Lécué et al., 2009; Meyer and Weske, 2006; Pathak et al., 2006; Pistore et al., 2005) in detail regarding the construction of control flow patterns: Focusing on parallel splits and synchronizations, most of these approaches state that two actions can be parallelized if they do not contradict each other. However, these approaches do not define concepts and thus do not specify when exactly an action is contradicting another action. This would be necessary to provide a concrete automated planning algorithm for the construction of parallelizations. Only Meyer and Weske (2006) state a formal concept to parallelize two actions, which is based on preconditions and effects not being in conflict. However, using this approach and focusing on two

actions means that the length of each path segment within parallelizations is limited to only one action (cf. Meyer and Weske, 2006). Moreover, construction of complex parallelizations such as nested parallelizations is not supported. Additionally, due to its heuristic nature, the authors do not aim to provide the set of feasible parallelizations. Furthermore, large sets of world states and large domains as well as respective numerical data types and also other large data types of state variables are not treated. Other authors in these phases propose to calculate so called dependency coefficients for each action and suggest to parallelize two actions if their dependency coefficients are the same (Omer, 2011; Omer and Schill, 2009; Rathore and Suman, 2015; Vanitha et al., 2012). Dependency coefficients represent how many actions are dependent on the considered action or how many actions the considered action is dependent on. However, similarly to (Meyer and Weske, 2006), the parallelized path segments are synchronized in any case after at most one action per path segment. Furthermore, nested parallelizations are not supported, and the approaches are heuristic. Additionally, large sets of world states as well as respective numerical data types and other large data types of state variables are not treated. The same holds for a similar approach proposed by Madhusudan and Uttamsingh (2006) which divides a sequence of actions into sets of actions that can be parallelized based on precedence constraints.

Further research related to our work is associated with the phase *process analysis*. In process mining, data about executed processes is stored in logs and used to enable the *re*construction of process models. For instance, Hwang and Yang (2002) present an approach in which process log data can be used to reconstruct the underlying process model and thus also control flow patterns such as parallel splits. The reconstruction of parallel splits and synchronizations in this research field is based on the execution order of actions discovered in the logs. Most approaches state that two actions are parallel if they appear in any order (see, e.g., van der Aalst et al., 2004; van der Aalst, 2012; Wen et al., 2007). This is of heuristic nature and a non-sufficient criterion, as, for instance, two actions may be executed in any order but not in parallel and at the same time because the same executing person (resource) is required for both actions. Other approaches also use logs with explicit timestamps enabling the identification of actions which were actually executed simultaneously (Weijters et al., 2006; Wen et al., 2007) or detecting overlapping actions (Wen et al., 2009). However, their intention and the presented algorithms are different to our research goal, since process mining focuses on the *re*construction of models for already *existing* processes. Therefore, these works do not aim to provide an approach for an automated construction of parallelizations in newly planned process models and thus do not present concepts to support this task. Moreover, as they rely on logs from existing process executions, these works do not deal with infinite sets of world states and large domains as well as respective data types used by the state variables. Further, Jin et al. (2016) propose an approach for refactoring process models and including parallelizations in the refactored process models. They do so by applying techniques from process mining and determining relations between actions, allowing to identify actions which can be parallelized. However, the authors strive to refactor *existing* process models and thus do not aim to construct parallelizations in newly

planned process models. Additionally – as Jin et al. (2016, pp. 464–465) state – their approach cannot guarantee that the resulting process models are sound structured, which makes the manual intervention of a modeler necessary when applying the approach. This impedes an automated construction of parallelizations by means of an algorithm. Furthermore, the presented approach strictly relies on petri nets and is thus dependent on a concrete modeling language.

To sum up: In the literature there are several valuable contributions regarding an automated identification or construction of parallel splits and synchronizations which could serve as a basis for our research. However, there is a research gap which can be stated in terms of the following relevant aspects (cf. Section *Theoretical Background*) not addressed by existing approaches (cf. Table 1):

(A1) Concepts stating how to construct feasible parallelizations in newly planned process models need to be provided. These concepts have to allow the construction of complex parallelizations, which means, the support of nested parallelizations and an arbitrary length of path segments within parallelizations. The concepts must ensure the consistency of the state transitions resulting from a parallelization and must be formally and clearly defined.

(A2) Possibly infinite sets of world states and large domains as well as respective large data types of state variables have to be treated.

(A3) The set of feasible parallelizations has to be provided while preventing infeasible parallelizations.

(A4) The approach needs to be independent of a concrete modeling language.

(A5) A concrete algorithm for an automated construction of parallelizations in newly planned process models has to be provided.

Phase	Works	(A1)	(A2)	(A3)	(A4)	(A5)
Process Modeling	Hoffmann et al. (2012)	✕	✕	○	✓	✕
Process Implementation & Process Execution	Bertoli et al. (2006); Bertoli et al. (2010); Binder et al. (2009); Constantinescu et al. (2004); Lécué et al. (2009); Pathak et al. (2006); Pistore et al. (2005)	✕	○	✕	○	✕
	Meyer and Weske (2006)	○	✕	○	✓	○
	Madhusudan and Uttamsingh (2006); Omer (2011); Omer and Schill (2009); Rathore and Suman (2015); Vanitha et al. (2012)	○	✕	○	✓	○

Process Analysis	van der Aalst (2012); van der Aalst et al. (2004); Weijters et al. (2006); Wen et al. (2007); Wen et al. (2009)	✕	○	○	○	✕
	Jin et al. (2016)	✕	✕	○	✕	✕
✓: considered; ✕: not considered; ○: partly considered						

Table 1. Overview of Related Work

2.3 Planning Domain

Based on control flow pattern theory, when planning process models, we have to cope with an abstraction from individual process executions. Therefore, the realizations of state variable values are not determined at the moment of planning and belief states instead of world states need to be considered (Ghallab et al., 2004). Here, a belief state represents possibly infinite sets of world states. When working with belief states it is common to deal with a nondeterministic planning problem and to refer to a nondeterministic planning domain. Both guarantee a maximum of compatibility with existing approaches in the literature (Bertoli et al., 2001, 2006; Heinrich et al., 2015; Heinrich and Schön, 2015, 2016; Sycara et al., 2003) and allow an acceptance and use of our approach. Central for the nondeterministic planning domain is the nondeterministic belief state-transition system. It is based on the notion of a belief state tuple, which is defined as follows:

Definition 1 (*belief state tuple*). A *belief state tuple p* is a tuple consisting of a *belief state variable v(p)* and a subset *r(p)* of its predefined *domain dom(p)*, which is written as *p:=(v(p),dom(p),r(p))*. The domain, *dom(p)*, specifies which values can generally be assigned to *v(p)*. The set $r(p) \subseteq dom(p)$ is called the *restriction* of *v(p)* and contains the values that can be assigned to *v(p)* in this specific belief state tuple *p*.

According to this definition, each belief state variable *v(p)* has a predefined data type (for example 'double') specifying the predefined domain *dom(p)*. Additionally, restrictions *r(p)* can be defined for each belief state variable *v(p)*. A restriction can either be described by logical expressions defining a set of values or an explicit enumeration of values. The notion of a belief state tuple is used in the formal definition of a nondeterministic belief-state transition system presented in the following. It is given in terms of its belief states, its actions and a transition function which describes how the application of actions leads from one belief state to possibly many belief states (Bertoli et al., 2006; Ghallab et al., 2004; Heinrich et al., 2009).

Definition 2 (*nondeterministic belief state-transition system*). Let BST be a finite set of belief state tuples. A *nondeterministic belief state-transition system* is a tuple $\Sigma = (BS, A, R)$, where

- $BS \subseteq 2^{BST}$ is a finite set of *belief states*. An element of BS, a belief state, is a subset of the finite set of belief state tuples BST, containing every belief state variable one time at the most.
- A is a finite set of *actions*. Each action $a \in A$ is a triple consisting of the action name and

two sets, which we will write as $a := (name(a), precond(a), effects(a))$. The set $precond(a) \subseteq BST$ are the *preconditions* of a and the set $effects(a) \subseteq BST$ are the *effects* of a. The term *preconditions* (including inputs) denotes everything an action needs to be applied, including tangible and non-tangible entities (e.g., data, materials, components), general conditions (e.g., time slot when an action is applicable) and resources (e.g., staff, machines). The term *effects* (including outputs) denotes everything an action provides, deallocates or alters after it was applied, including tangible and non-tangible entities, general conditions and resources.[1]

- An action a is *applicable* in a belief state bs iff $\forall w \in precond(a)\ \exists u \in bs: v(w)=v(u) \wedge r(w) \cap r(u) \neq \emptyset$. In other words, a is applicable in bs iff all belief state variables in $precond(a)$ also exist in bs and the respective restrictions of the belief state variables intersect.
- $R: BS \times A \longrightarrow 2^{BS}$ is the *transition function*. The transition function associates to each belief state $bs \in BS$ and to each action $a \in A$ the set $R(bs, a) \subseteq BS$ of next belief states.

According to Definition 2, a state variable of the preconditions and effects is defined as belief state tuple that consists of the name of the state variable, its domain and a set of values, all of which can be assigned to the state variable in a specific world state (according to an individual process execution). From a process modeling perspective, this is a natural way to express certain preconditions and effects of actions and allows to represent possibly infinite sets of world states.

Definition 3 ((*non-*)*determinism in state space*). An action a is *deterministic* in a belief state bs iff $|R(bs, a)| = 1$. It is *nondeterministic* if $|R(bs, a)| > 1$. If a is applicable in bs, then $R(bs, a)$ is the set of belief states that can be reached from bs by applying a.

Based on both Definitions 2 and 3, a planning graph can be generated by means of several existing algorithms that progress from an initial belief state to goal belief states (see for example (Bertoli et al., 2001, 2006; Heinrich and Schön, 2015; Sycara et al., 2003)). Here, a planning graph is defined as:

Definition 4 (*planning graph*). A *planning graph* is an acyclic, bipartite, directed graph $G = (N, E)$ with the set of nodes N and the set of edges E. Henceforth, the set of nodes N consists of two partitions: The set of action nodes $Part_A$ and the set of belief state nodes $Part_{BS}$. Each node $bs \in Part_{BS}$ represents one distinct belief state from the set BS of belief states in the planning graph. Each node $a \in Part_A$ represents an action from the set A of actions in the planning

[1] To give an example: With the help of preconditions, data entities such as securities order data entities as well as bank employees (human resources) can be specified which are needed to apply an action "process buying order". Its effects specify, for example, that the securities order data entities are altered and the previously allocated bank employees are deallocated.

graph. The planning graph starts with one explicit initial belief state $bs_{init} \in BS$ and ends with one to possibly many goal belief states $bs_{goal_j} \in BS$.

Given Definition 4, a planning graph may consist of one to many paths. Here, a path is defined as:

Definition 5 (*path*). A *path* in a planning graph is a sequence $(bs_{init}, a_1, bs_2, a_2, \ldots, a_n, bs_{n+1})$ of belief state nodes and action nodes starting with the initial belief state and ending in exactly one goal belief state with each action being represented one time at the most.

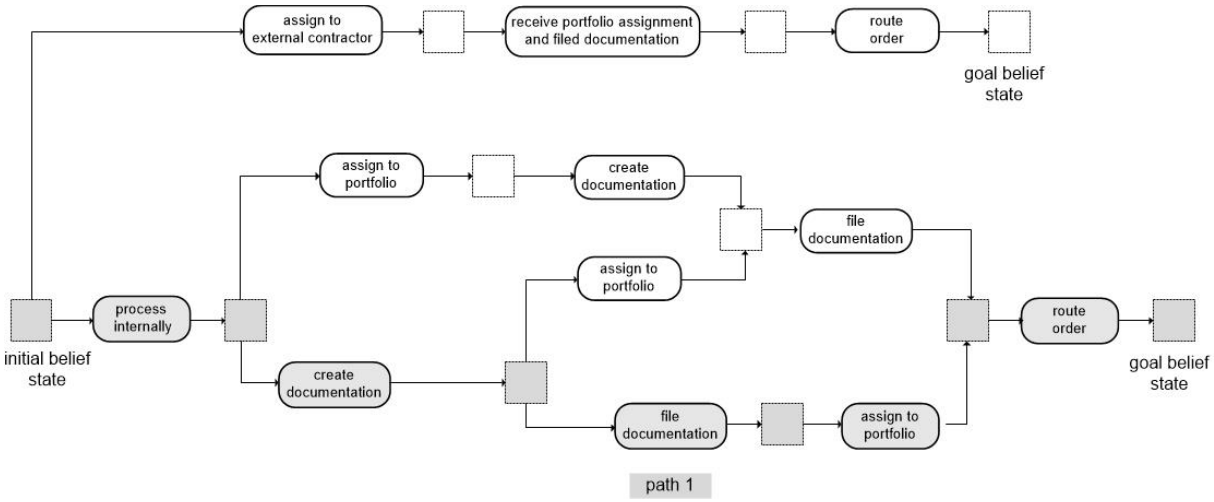

Figure 1. Excerpt of the Order Management of a Financial Services Provider

To illustrate the above definitions of a planning domain and to introduce a running example, Figure 1 shows an excerpt of the real-world order management of a financial services provider. Here, the (internal or external) processing of an incoming order is performed. The full planning graph from which this example is taken can be found in the *Evaluation*. In our case the graph is planned by applying the approach suggested by Bertoli et al. (2006); however other approaches such as (Heinrich et al., 2009) are also feasible and provide the same graph. If a manually constructed graph (respectively, process model) is available, our approach may be applied as well to allow the construction of (additional) parallelizations for such models. The specification of the initial belief state and the condition for a belief state to be a goal belief state are given in Table 2.

initial belief state	{(order state, state, {passed}), (order price, double+, double+), (order amount, int+, int+), (internal processing, state, {unknown}), (documentation state, state, {not created}), (portfolio assignment, boolean, {false})}
condition for goal belief state	{(order state, state, {routed})}

Table 2. Initial Belief State and Condition for Goal Belief State

In the initial belief state, an order has already been placed in terms of an order state, a price and an amount. The condition for a belief state to be a goal belief state of the presented excerpt

represents that the order has been routed. Several actions are necessary before an order can be routed. The company can decide to mandate an external contractor (*assign to external contractor*) that provides a package which encapsulates all needed actions (*receive portfolio assignment and filed documentation*). After running these actions, the order can be routed (*route order*) to reach a goal belief state. If the company chooses not to mandate the external contractor, the action *process internally* enables the execution of three tasks which have to be completed before the order can be routed: *assign to portfolio*, *create documentation* and *file documentation*. The planning graph exhibits four possible sequences of actions to reach a goal belief state starting from the initial belief state and thus contains four paths (cf. Definition 5). In the following Table 3, we present the actions of one of the paths (marked in grey as path 1 in Figure 1) according to Definition 2. The remaining paths and actions are analogously annotated.

Action	Preconditions	Effects
process internally	{(internal processing, state, {unknown})}	{(internal processing, state, {true})}
create documentation	{(internal processing, state, {true}), (documentation state, state, {not created})}	{(documentation state, state, {created})}
file documentation	{(internal processing, state, {true}), (documentation state, state, {created})}	{(documentation state, state, {filed})}
assign to portfolio	{(internal processing, state, {true}), (portfolio assignment, boolean, {false})}	{(portfolio assignment, boolean, {true})}
route order	{(order state, state, {passed}), (order price, double+, double+), (order amount, int+, int+), (portfolio assignment, boolean, {true}), (documentation state, state, {filed})}	{(order state, state, {routed})}

Table 3. Order Management: Annotation of the Actions of Path 1

In path 1, the company chooses to process the order internally (action *process internally*), setting the value of the belief state variable `internal processing` to "true". Internal processing enables the creation of a documentation (action *create documentation*). This creation is represented by the belief state variable `documentation state` whose value is altered from "not created" to "created". After the documentation is created, it is filed. Therefore, the action *file documentation* requires the value "created" of `documentation state` and transforms it into "filed". Finally, the portfolio needs to be updated (action *assign to portfolio*), which alters the value of the belief state variable `portfolio assignment` to "true". Until now, the order could not be routed (action *route order*), since this requires a filed documentation as well as an existent portfolio assignment as represented by the preconditions of *route order*. Applying *route order* leads to the value of the belief state variable `order state` changing from

"passed" to "routed". As this also represents the condition for a goal belief state, *route order is* the last action applied in the path.

3 Approach for the Automated Construction of Parallelizations

In this section, we present our concepts and algorithm for the automated construction of parallel splits and synchronizations. Figure 2 illustrates the approach on an abstract level by showing which part of the paper represents existing knowledge, which concepts we introduce and how the algorithm works.

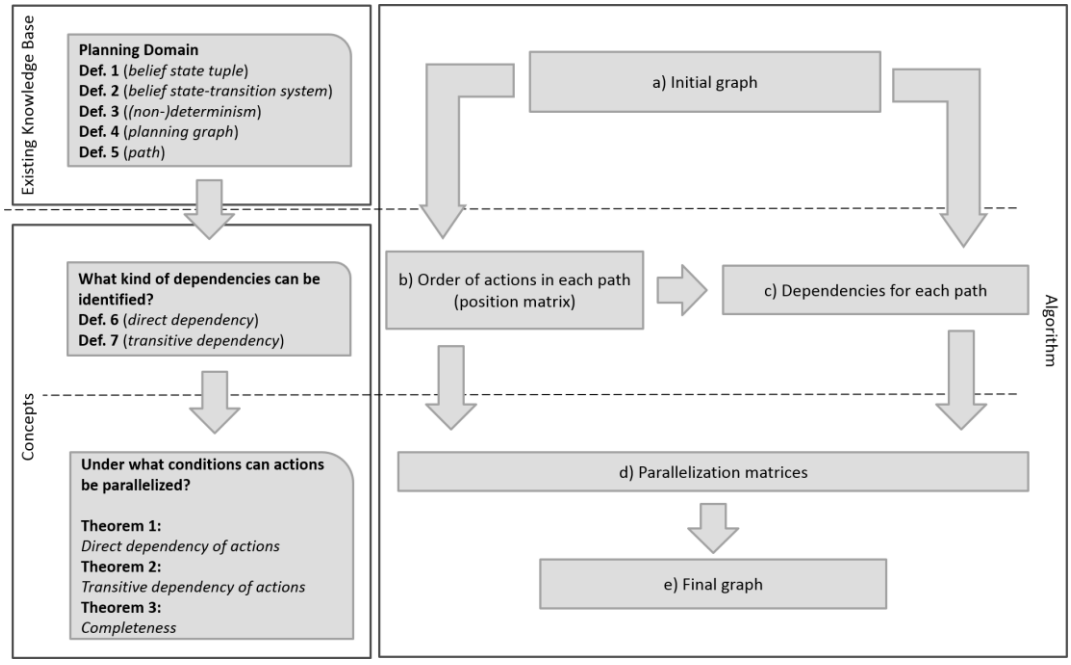

Figure 2. Overview of our Approach for the Construction of Parallelizations

We build our research on both the planning domain and planning graph (cf. Definitions 4 and 5; area a) in Figure 2), which can be constructed by existing algorithms. The graph contains all sequences of actions starting from the initial belief state and resulting in goal belief states. To provide a complete and correct solution to the problem of constructing the set of feasible parallelizations in a graph, we state concepts ("dependencies", cf. section *Concepts*) that describe conditions under which actions can be parallelized. To be more precise, we will first define "direct dependencies" between actions (cf. Definition 6). We will then show the connection of this notion to parallelizing actions. However, these direct dependencies will prove insufficient to construct the set of all feasible parallelizations, especially more complex parallelizations such as nested parallelizations. Therefore, we will introduce the concept of "transitive dependency" of actions (cf. Definition 7), critically complementing direct dependencies and enabling a correct and complete construction of parallelizations (cf. Theorems 1-3). More precisely, we will

prove that if and only if neither of these dependencies occur, the regarded actions can indeed be parallelized.

An algorithm stating how to analyze these dependencies and how to construct all feasible parallelizations is described in the section *Algorithm*. For this analysis, it needs to be taken into account which action is succeeding another action in a certain path of the planning graph. To this end, our algorithm creates a position matrix representing the order of actions in each path of the planning graph (cf. area b) in Figure 2). Using this matrix and the identified dependencies (cf. area c) in Figure 2), parallelization matrices for each path of the planning graph can be constructed. These matrices show which actions are directly or transitively dependent on each other and which actions can be parallelized (cf. area d) in Figure 2) based on the respective path. When combined, the parallelization matrices therefore indicate every feasible parallelization and enable the construction of the final graph (cf. area e) in Figure 2) containing all parallelizations.

3.1 Concepts

The first idea to identify actions that can be parallelized is to compare the preconditions and effects of actions in a path. If this analysis shows that the effects of two compared actions are not disjoint from each other, or that the effects of one action intersect with the preconditions of the other action, we call this a direct dependency of both actions in the following.

Definition 6 (*direct dependency* ⇠): Let $(bs_{init},a_1,bs_2,a_2,...,a_n,bs_{n+1})$ be a path in the planning graph and let a_i and a_j be actions in this path with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1,\dots,n-1\}$, $j \in \{2,\dots,n\}$. The action a_j is *directly dependent* on the action a_i (denoted by $a_i \dashleftarrow a_j$) iff:

$$\left(v(effects(a_i)) \cap \left(v\left(precond(a_j)\right) \cup v\left(effects(a_j)\right)\right)\right)$$

$$\cup \left(v\left(effects(a_j)\right) \cap v(precond(a_i))\right) \neq \varnothing$$

Here, $v(\dots)$ denotes the belief state variables of the tuples of the regarded set.

To illustrate Definition 6, consider the actions *process internally* and *create documentation* from the running example above. The effects of *process internally* and the preconditions of *create documentation* have the belief state variable internal processing in common. Therefore, these actions are directly dependent in every path containing both actions. This definition can be used to gain information about feasible parallelizations via the following theorem.

Theorem 1: Let $(bs_{init},a_1,bs_2,a_2,...,a_n,bs_{n+1})$ be a path in the planning graph and let a_i and a_j be actions in this path with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1,\dots,n-1\}$, $j \in \{2,\dots,n\}$.

a) If a_j is directly dependent on a_i (i.e., $a_i \dashleftarrow a_j$), a_i and a_j cannot be parallelized.

b) If a_j is *not* directly dependent on a_i and $j = i + 1$ (i.e., a_j is *directly* succeeding a_i), a_i and a_j can be parallelized.

Theorem 1 as well as all following theorems are proven in the *Supplement*. This theorem enables the construction of parallelizations with respect to directly adjacent actions. However, in order to construct complex parallelizations (including nested parallelizations and parallelizations with an arbitrary length of path segments), non-adjacent actions have to be analyzed as well. For that purpose, direct dependencies are not a sufficient concept, because it might or might not be correct to parallelize such actions that are not directly dependent. Therefore, we have to state under which additional concept it is feasible to parallelize two non-adjacent actions.

Definition 7 (*transitive dependency*): Let $p = (bs_{init},a_1,bs_2,a_2,...,a_n,bs_{n+1})$ be a path in the planning graph and let a_i and a_j be actions in p with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1, \dots, n-2\}$, $j \in \{3, \dots, n\}$. The action a_j is *transitively dependent* on the action a_i in p iff there is a set $A_k = \{a_{k_1}, \dots, a_{k_m}\} \subseteq \{a_{i+1}, \dots, a_{j-1}\}$, $A_k \neq \emptyset$, such that $a_i \dashleftarrow a_{k_1} \dashleftarrow \dots \dashleftarrow a_{k_m} \dashleftarrow a_j$.

A transitive dependency in a path can be seen as a continuous chain of direct dependencies among a non-empty subset of actions in that path, leading from one action to another. Evidently, the concrete ordering of actions in a path plays a crucial role for transitive dependency: The actions $a_{k_1}, \dots, a_{k_m}$ that result in a transitive dependency of an action a_j on an action a_i in a path p might, even if they are contained in a path p', fail to do so in p' due to being in a different ordering (for example, in p', one of the actions $a_{k_1}, \dots, a_{k_m}$ may be executed after a_j). This underlines the need of a path-wise definition of transitive dependency. Definition 7 can be used to gain information about feasible parallelizations via the following theorem.

Theorem 2: Let $p = (bs_{init},a_1,bs_2,a_2,...,a_n,bs_{n+1})$ be a path in the planning graph and let a_i and a_j be actions in p with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1, \dots, n-2\}$, $j \in \{3, \dots, n\}$.

a) If a_j is transitively dependent on a_i, the actions a_i and a_j cannot be parallelized based on p.

b) If a_j is neither directly nor transitively dependent on a_i, the actions a_i and a_j can be parallelized.

Focusing only on a single path, we might at first "miss out" (from a graph-wise perspective) a certain parallelization by not parallelizing transitively dependent actions (cf. Theorem 2a)), if these actions are not dependent on each other in another path of the planning graph. However, the respective parallelization is then constructed based on the analysis of that path:

Theorem 3 (*completeness*): Let G be a planning graph consisting of the paths $p_1,...,p_k$. Suppose the actions $a_1, \dots, a_n$ represented in G can be parallelized. By analyzing direct and transitive dependencies in all paths $p_1,...,p_k$, the parallelization of $a_1,...,a_n$ is constructed.

This result finalizes the development of our concepts. Thus, the set of feasible parallelizations including nested parallelizations and parallelizations consisting of path segments with more than one action can be constructed based on our formally defined concepts of direct dependency, transitive dependency and completeness.

3.2 Algorithm

In this section, we present an algorithm which builds on the concepts and allows to construct complete graphs while also being computationally efficient (cf. Section *Evaluation*). Let P be the set of all paths contained in the planning graph G (as planned by existing approaches; e.g., Bertoli et al., 2006; Heinrich et al., 2009). For each $p \in P$ we define a *parallelization matrix* M_p. The purpose of a parallelization matrix is to show which actions can be parallelized based on the respective path. To this end, our algorithm fills the parallelization matrices with entries determining whether to allow or to prohibit parallelization based on the concepts from the previous section. The family $(M_p)_{p \in P}$ then indicates all feasible parallelizations of the whole graph. The pseudo code of the algorithm is shown in the Table 4 (an extended version with comments is available in the *Supplement*). The algorithm relies on four steps, which are exemplified in the following by our running example:

```
1  Vector allActions:= new Vector()
2  [][] positionMatrix:= new int [#actionsInGraph][#pathsInGraph]
3  for all p ∈ (1 ≤ p ≤ #pathsInGraph)
4    for all i ∈ (1 ≤ i ≤ p.length)
5        if (a[i][p] ∉ allActions) then
6            allActions.add(a[i][p])
7        end if
8        positionMatrix[allActions.getIndex(a[i][p])][p] = i
9    end for
10 end for
11 Vector ParaMatrices:= new Vector()
12 for all p ∈ (1 ≤ p ≤ #pathsInGraph)
13  [][]ParaMatrix:= new String[allActions.length][allActions.length]
14  ParaMatrices.insertElementAt(ParaMatrix, p)
15 end for
16 for all p ∈ (1 ≤ p ≤ #pathsInGraph)
17  for all i ∈ (2 ≤ i ≤ allActions.length)
18      if (positionMatrix[i][p]=0) then
19          continue
20      end if
21      for all j ∈ (i-1 ≥ j ≥ 1)
22          if (positionMatrix[j][p]=0) then
23              continue
24          end if
25          if (ParaMatrices.elementAt(p).[i][j] ≠ ddep) then
26              if(v(effects(a[i])) ∩ (v(precond(a[j])) ∪ v(effects(a[j]))) ≠∅ ∨ v(precond(a[i])) ∩ v(effects(a[j])) ≠∅) then
27                  for all a ∈ (p ≤ a ≤ #pathsInGraph) do
28                      if (positionMatrix[i][a]=0 ∨  positionMatrix[j][a]=0) then
29                          continue
30                      end if
31                      ParaMatrices.elementAt(a).[i][j] ← ddep
32                  end for
33              else
34                  if (|positionMatrix[i][p]-positionMatrix[j][p]| = 1) then
35                      ParaMatrices.elementAt(p).[i][j] ← para
```

```
36                  end if
37              end if
38          end if
39      end for
40  end for
41 end for
42 for all p ∈ (1 ≤ p ≤ #pathsInGraph)
43  for all i ∈ (3 ≤ i ≤ p.length)
44      for all j ∈ (i-2 ≥ j ≥ 1)
45          pos_i:= allActions.getindex(a[i][p])
46          pos_j:= allActions.getindex(a[j][p])
47          if(ParaMatrices.elementAt(p).[Max(pos_i,pos_j)][Min(pos_i,pos_j)]) ≠ ddep) then
48              for all k ∈ (i > k > j)
49                  pos_k:= allActions.getindex(a[k][p])
50                  if((ParaMatrices.elementAt(p).[Max(pos_i,pos_k)][Min(pos_i,pos_k)]) = (ddep ∨ tdep))
51                  ∧ (ParaMatrices.elementAt(p).[Max(pos_j,pos_k][Min(pos_j,pos_k)]) = (ddep ∨ tdep))) then
52                      (ParaMatrices.elementAt(p).[Max(pos_i,pos_j)][Min(pos_i,pos_j)]) ←tdep
53                      break for
54                  end if
55              end for
56          end if
57          if(ParaMatrices.elementAt(p).[Max(pos_i,pos_j][Min(pos_i,pos_j] ≠(ddep ∨ tdep)) then
58              ParaMatrices.elementAt(p).[Max(pos_i,pos_j][Min(pos_i,pos_j]←para
59          end if
60      end for
61  end for
62 end for
```

Table 4. Pseudocode of our Algorithm

1) A list of the actions in the graph and a position matrix, containing the position of each action in each path, is generated (line 1-10). To this end, first the actions of the graph are determined in the order in which they appear (line 3-7): this means, all actions of a first path (in our example, *process internally, create documentation, file documentation, assign to portfolio, route order*; cf. Figure 1) are followed by the actions in other paths that were not part of the first path (*assign to external contractor, receive portfolio assignment and filed documentation*)[2]. Then, the position matrix containing the position of every action in each path of G is generated (line 8). The rows represent the actions (in the order identified before), the columns correspond to the different paths. For our example with the four paths $p1$, $p2$, $p3$ and $p4$, this yields the following position matrix:

$$\begin{array}{c} \\ pi \\ cd \\ fd \\ ap \\ ro \\ ae \\ re \end{array} \begin{array}{c} \begin{array}{cccc} p1 & p2 & p3 & p4 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & - \\ 2 & 2 & 3 & - \\ 3 & 4 & 4 & - \\ 4 & 3 & 2 & - \\ 5 & 5 & 5 & 3 \\ - & - & - & 1 \\ - & - & - & 2 \end{pmatrix} \end{array}.$$

Abbreviation	Action
pi	process internally
cd	create documentation
fd	file documentation
ap	assign to portfolio
ro	route order
ae	assign to external contractor
re	receive portfolio assignment and filed documentation

Here, " $-$ " denotes that the action is not part of the respective path.

[2] A different order of the paths does not lead to different sets of feasible parallelizations.

2) A set of (at first, empty) parallelization matrices is constructed (lines 11-15). The rows and columns of every parallelization matrix M_p represent all actions contained in G ordered by their position as identified in step 1). Each entry determines a row-column-combination and therefore an action-action-combination. For our example, this means that four (one for each path) parallelization matrices M_1 to M_4 are generated, each row and column representing one of the seven actions contained in the graph.

3) The algorithm examines – for all paths – the direct dependencies between pairs of actions in the respective path (lines 16-41)[3]. Whenever a direct dependency is identified in a path p (line 26), it is inserted in the respective entry in M_p. The concept of direct dependency is path-overarching, so that additionally, to reduce computing time, an identified direct dependency is also inserted into all entries corresponding to these two actions in the subsequent paths (lines 27-32). Following Theorem 1a), direct dependencies prohibit parallelization. When actions are *not* directly dependent, it is examined whether one of the actions is directly succeeding the other action in the considered path (lines 33-37). This is done via the position matrix. If this is indeed the case, the potential parallelization is noted in the corresponding entry in M_p (line 35), which is justified by Theorem 1b). In our example, the analysis of the direct dependencies starts with the actions *process internally* and *create documentation*. The effects of *process internally* and the preconditions of *create documentation* have the belief state variable internal processing in common (cf. Table 3 for an overview of preconditions and effects), resulting in a direct dependency of those two actions. Therefore, this direct dependency is inserted in the parallelization matrices M_1 to M_3, since both actions are applied in the first three paths. The algorithm then examines *file documentation* and *create documentation* (directly dependent, because the effects of both actions contain the belief state variable documentation state), *file documentation* and *process internally* (directly dependent due to the common belief state variable internal processing), *assign to portfolio* and *file documentation* (not directly dependent because of no common belief state variable) etc. and inserts the respective entries in the parallelization matrices.

4) The transitive dependencies are worked out (necessarily path-wise; lines 42-62). Only actions which are not directly dependent and which are not directly succeeding each other remain to be examined, reducing computing time. The algorithm searches for a set A_k of actions as in the definition of transitive dependency (Definition 7). This is done in a special proceeding order to guarantee that all dependencies required to examine a certain transitive dependency have already been determined beforehand (cf. for-loops in lines 43, 44 and 48). More precisely, the algorithm at first searches for a transitive dependency by adjacent actions (for example between action 1 and action 3 by action 2). Thereafter, the algorithm searches for transitive dependencies between non-adjacent actions (so that, e.g., for examining the transitive dependency of action

[3] Only one entry for each pair of actions is required, so in this and the following steps, just a triangular matrix needs to be considered and without loss of generality, all entries above the main diagonal can be disregarded (cf. for-loops, e.g., line 21). Also, only matrix entries for actions that actually appear in the respective path need to be filled out (lines 18-24, lines 28-29).

4 on action 1, the potential transitive dependency between action 1 and action 3 can be already used). Every transitive dependency is noted in the parallelization matrix of the considered path, prohibiting parallelization (cf. Theorem 2a); line 52). If neither direct nor transitive dependency is discovered between two actions in a path, the actions can be parallelized (cf. Theorem 2b); lines 57-58). In our example, the first potential transitive dependency that the algorithm analyzes is the one between *create documentation* and *assign to portfolio* in path 1 (cf. Figure 1), since it is already known that *file documentation* is directly dependent on *process internally*. However, *assign to portfolio* is not dependent on *file documentation* (which is the action applied in-between *assign to portfolio* and *create documentation*), and thus *create documentation* and *assign to portfolio* are not transitively dependent in path 1. Thus, the possibility to parallelize *assign to portfolio* and *create documentation* is included in M_1. The algorithm proceeds by examining the pair *route order* and *process internally* (in path 1). Here, *route order* is directly dependent on *assign to portfolio*, which itself is directly dependent on *process internally*, leading to a transitive dependency of *route order* and *process internally* that is entered in M_1. In this way, the algorithm identifies all transitive dependencies for all paths.

Based on the parallelization matrices, the process model containing all feasible parallelizations can be constructed by iterating over all parallelization matrices, including each feasible parallelization (as indicated in of the matrices) in the process model and removing redundant parallelizations. Thus, Theorem 3 is considered. For the running example, the completed parallelization matrices are

$$M_1 = M_2 = M_3 = \begin{array}{c} \\ pi \\ cd \\ fd \\ ap \\ ro \\ ae \\ re \end{array} \begin{array}{c} \begin{array}{ccccccc} pi & cd & fd & ap & ro & ae & re \end{array} \\ \begin{pmatrix} - & - & - & - & - & - & - \\ ddep & - & - & - & - & - & - \\ ddep & ddep & - & - & - & - & - \\ ddep & para & para & - & - & - & - \\ tdep & ddep & ddep & ddep & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \end{pmatrix} \end{array} \quad M_4 = \begin{array}{c} \\ pi \\ cd \\ fd \\ ap \\ ro \\ ae \\ re \end{array} \begin{array}{c} \begin{array}{ccccccc} pi & cd & fd & ap & ro & ae & re \end{array} \\ \begin{pmatrix} - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ - & - & - & - & tdep & - & - \\ - & - & - & - & ddep & ddep & - \end{pmatrix} \end{array}.$$

They are used to construct the final graph (depicted in Figure 3) including all feasible parallelizations.

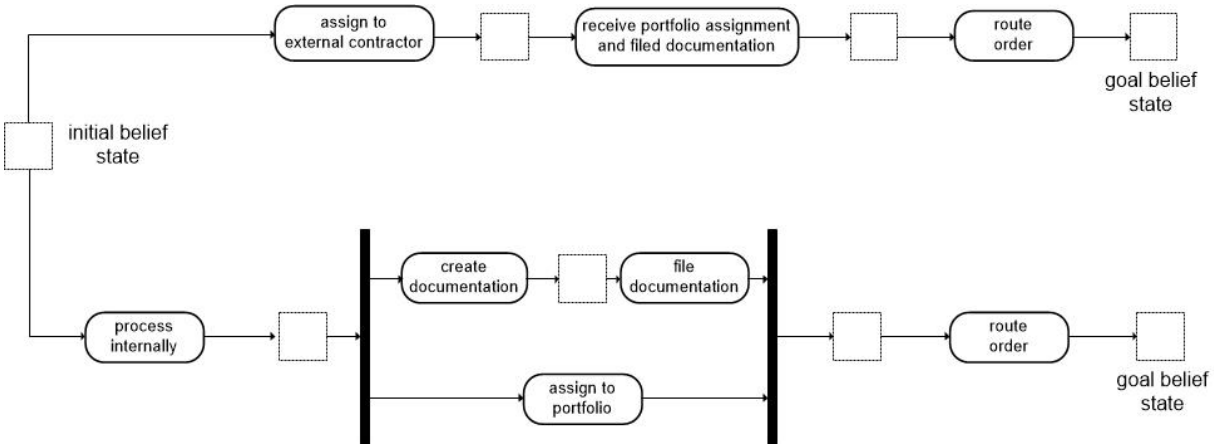

Figure 3. Final Graph resulting from the Application of the Algorithm to the Running Example

4 Evaluation

The presented approach was evaluated as shown in this section.

4.1 Analysis of the Algorithm Properties

We mathematically evaluated the algorithm in terms of the key properties termination, completeness and computational complexity and summarize the results in the following (proofs and calculations are available in the *Supplement*).

Termination: The algorithm terminates.

Correctness/Completeness: The algorithm leads to complete and correct parallelization matrices: Every required entry is inserted and there is no entry that would allow an infeasible parallelization or prohibit a feasible parallelization.

Computational Complexity: When evaluating the computational complexity of our algorithm, we considered the worst-case-scenario as is usual. The following results were achieved: Given a planning graph in which each path has n actions and each action has m preconditions and m effects, the asymptotic time complexity of our algorithm is $O(n^3)$ and $O(m^2)$. This polynomial run time underlines the computational efficiency (cf. Arora and Barak, 2009; Cobham, 1965) and thus practical applicability of the algorithm. We did not evaluate the computational complexity of our algorithm in comparison to competing algorithms since it solves a heretofore unsolved problem (cf. aspects (A1) – (A5) in *Related Work and Research Gap*).

4.2 Operational Evaluation

To examine its technical feasibility and practical applicability (Prat et al., 2015), we examined our approach with respect to the following three evaluation questions:

(*E1*) Can the algorithm be realized in a prototypical implementation?

(*E2*) Can the algorithm be applied to real-world processes and how can the necessary input data (i.e., the specification of actions, initial belief state and conditions for goal belief states) be obtained?

(*E3*) Which output results from the application of the algorithm to real-world processes?

In regard to (*E1*), a Java implementation of an existing algorithm for the automated construction of planning graphs (Bertoli et al., 2006) served as a basis for our work. This implementation allows the import of actions, initial belief states and conditions for goal belief states specified in form of XML files. We extended the implementation to incorporate the presented algorithm for the automated construction of parallelizations. The validity of the prototype was ensured by means of structured tests using the JUnit framework and planning test process models. At the end of the test phase, the implementation did not exhibit any errors. This result supports the technical feasibility of the algorithm and provides "proof by construction" (Hevner et al., 2004; Nunamaker et al., 1991) [4].

With respect to (*E2*) we analyzed the algorithm in-depth in different real-use situations using our prototypical implementation[5]. In the following, we exemplarily focus on one of these real-world processes referring to the order management of a European financial services provider (the running example used above is part of this process as well). More precisely, this process addresses the execution of security orders where several steps including check routines have to be modeled (cf. Figure 4). In the past, this process had to be (re)designed several times due to new services, new regulations or changing organizational requirements (for example, when outsourcing parts of the process to external service providers). To evaluate our approach we focused on the previous redesigns of this process and analyzed whether it is possible to apply the approach in these redesign situations and to what extent the results of the automated planning match with manually designed parallelizations.

In order to apply the algorithm, we conducted two steps: First, we obtained the necessary input data. To do so, a set of actions was extracted based on former process models in the area of security order management. This could be done easily and in an automated manner via the financial services provider's process modeling tool (ARIS toolset) which features a XML interface. Such an interface can be used in order to export actions to our prototype. In the area of security order management, about 200 different actions including their preconditions and effects were imported from the ARIS toolset and verified. Besides, a small number of additional actions was modeled manually. Moreover, the initial belief state and conditions for goal belief

[4] In this context, a web interface for the implementation capable of planning process models in an automated manner has been prepared. It can be accessed using the following link: http://www-sempa.ur.de/

[5] The prototype was run on an Intel Core i7-2600 3.40 GHz running Windows 7, 64 Bit and Java 8, Build-Version 1.8.0_05-b13.

states were specified in cooperation with the financial services provider. Then, the process models were planned using the prototype. This second step took less than two seconds in case of the order management process model.

Concerning (*E3*), we examined the output. Figure 4 shows an entire planned process model[6]. Here, our algorithm constructed two parallelizations which were also part of the manually designed process model. The first parallelization is constructed after the action *proof stock*, where the actions *enter quantity* and *determine market value* are parallelized. The second parallelization refers to our running example. Here, the action *assign to portfolio* is parallelized to the actions *create documentation* and *file documentation*. The assessment underlined the applicability and feasibility of the algorithm in all redesign situations of the security order management process.

To further address the evaluation questions, the presented approach was applied in additional real-use situations from various application contexts and different companies. These applications are discussed in the *Supplement*. The analysis of the evaluation questions (*E1*)-(*E3*) supported the technical feasibility and practical applicability of the presented approach. Table 5 summarizes the results.

Evaluation Question	**Result**
(*E1*) Can the algorithm be realized in a prototypical implementation?	The algorithm was implemented and successfully integrated into a prototype for the automated planning of process models.
(*E2*) Can the algorithm be applied to real-world processes and how can the necessary input data (i.e., the specification of actions, initial belief state and conditions for goal belief states) be obtained?	The algorithm was applied in several real-use situations of various application contexts and different companies. The analyzed situations included up to 278 actions and 189 states in the planning graph and are of a medium to large size. This is also reflected in the number of paths of the different planning graphs which ranges up to over 1.2 million (due to the various orders the actions can appear in). The necessary input data could, for example, be obtained by the XML interface of an existing modeling tool. Our algorithm was able to cope with the required data types and could be applied in all situations without restrictions. The run time of the algorithm varied – depending on the size and complexity of the processes - from a few milliseconds up to around 12.5 minutes.

[6] Note that in this figure, the two paths of our running example have been merged before the action *route order*, since the process model is represented as a UML activity diagram without state nodes. This diagram type was the modelling notation preferred by the financial services provider.

(*E3*) Which output results from the application of the algorithm to real-world processes?	The algorithm constructed parallelizations for each of the real-world processes. For a significant number of processes, complex parallelizations (e.g., nested parallelizations) were constructed. The algorithm provided the manually constructed parallelizations and further, additional feasible parallelizations.

Table 5. Results with regard to the Evaluation Questions (E1)-(E3)

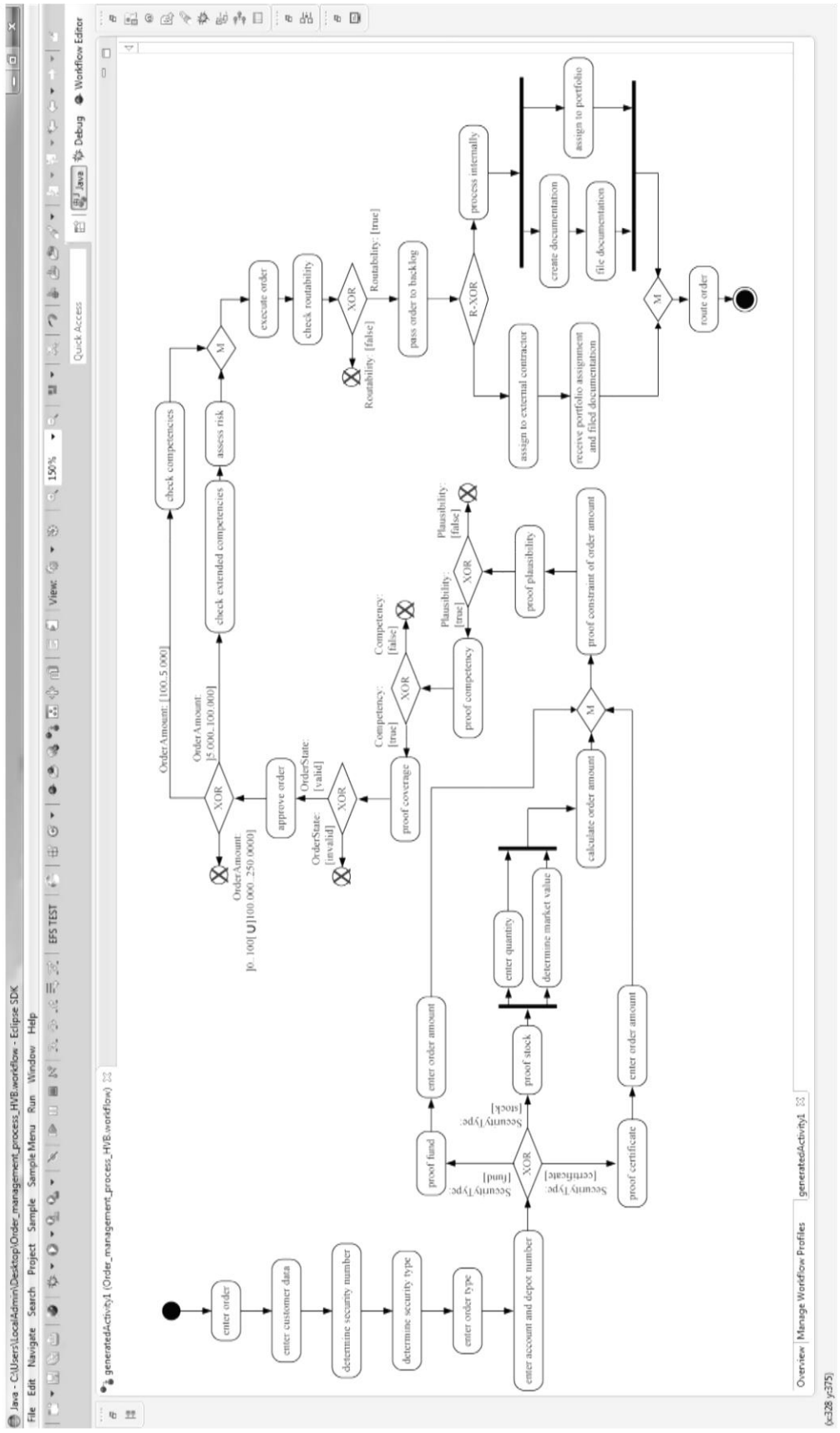

Figure 4. Planned Model of the Order Management Process (Screenshot Prototype)

4.3 Practical Utility

We further assessed the practical utility (Prat et al., 2015) of our approach by means of a naturalistic ex post evaluation (Venable et al., 2012). Its application resulted in the construction of the parallelizations already contained in the (existing) manually designed process models as well as additional feasible parallelizations and consequently increased flexibility by definition (cf. van der Aalst, 2013). Thereby, flexibility by definition represents the ability to consider alternative execution routes at planning time (in our context, facilitated by feasible parallelizations). This capability is of practical use for decision support because alternative execution routes can be assessed based on economic and resource criteria constraints. Subsequently the most beneficial execution route can be selected for process execution. For instance, in this way, an execution route with favorable execution time may be chosen when necessary. The real-use situation of this naturalistic evaluation is presented in the following Table 6 (Sun and Kantor, 2006; Venable et al., 2012).

General setting	Extensive project at a European financial services provider aiming for an improved transparency of costs, execution times and capacities with regard to core business processes
Available data and systems	Detailed information as well as key economic indicators such as total cost, total required execution time and personnel requirements for a large number of business processes and the actions covered by these processes; provided by process experts and executives of the financial services provider
Involved people	Multiple organizational units of the European financial services provider and their employees (business and process experts, executives)
Hypothesis	Realizing a previously non-identified feasible parallelization should reduce total costs and total required execution times while increasing resource utilization, as long as the necessary resources for concurrent execution are available. This should also help in the prevention of errors and claims occurring during process execution.

Table 6. Real Environment analyzed in the Naturalistic Evaluation

Similar to Siha and Saad (2008), we exemplarily discuss two selected cases in the context of the "Contracting wealth management customer" process (cf. Table C.1 in *Supplement*) in Table 7.

Subprocess	**Managing depot conditions**	**Handling non-executed security paper orders**
Description of the subprocess	Customer inquiries lead to changed depot conditions which are issued by the respective employees in charge. These change requests are stored in a list, which has to be worked through by different organizational units of the financial services provider to complete the needed change.	A variety of problems results in non-executed security paper orders issued by employees in charge of the financial services provider. These orders need to be rectified, forwarded and executed.
Organizational units involved	Advisors / multiple regional service divisions / processing department / process management department	Advisors / regional service division / commerce, sales and deposits units / processing department / process management department / financial market services
Issue	The previously existing sequential execution of actions occurring when, for instance, a customer opened a deposit account had resulted in a significant time gap between the opening and the completion of the respective inquiry. This, in turn, had led to customer complaints and repeated effort of the employees in charge.	Discussions with different organizational units revealed that for certain actions, it had not been clear which unit was in charge. Time delays resulting from the sequential execution of these actions had resulted in long execution times and many unnecessary internal inquiries and reworks. This in turn had led to claims of customers because overdue security paper orders had been deleted erroneously.
Improvement potential	A clear division of responsibility between the different organizational units of the financial services provider allowed a (previously not identified) concurrent execution of actions (i.e., nested parallelizations). The feasibility of this concurrent execution of actions with respect to economic criteria and resource constraints was confirmed by experts in a workshop based on which the employees in charge were informed and trained.	Our analysis showed that, as long as different organizational units were responsible for some of the actions, a parallelization of these actions was not only feasible, but highly beneficial. A workshop with the respective organizational units (including, e.g., the sales, commerce and deposits units) was conducted to ensure that the proposed concurrent action execution would be possible based on economic criteria and resource constraints. Thereby, it was also ensured that each organizational unit was only in charge of the actions it was capable for.

Results	The concurrent execution of previously sequentially executed actions could be realized. In this way, a large number of time delays and repeated efforts could be avoided. A 50%-reduction in occurrence of these aspects led to saving 20% of total required execution time. For the employees, this amounted to an average reduction of at least 12 minutes of working time per process execution. Additionally, realizing the improved feasible execution route including the concurrent execution of actions resulted in an optimization potential for cost savings of 1.2 full time equivalents p.a.	The concurrent execution of actions allowed an improved workload efficiency and thus an optimization potential for cost savings amounting to 1.42 full time equivalents p.a. Furthermore, due to a reduction of the total required execution time, the aforementioned claims could be reduced or even avoided.

Table 7. Selected Cases in the Naturalistic Evaluation

Overall, our approach demonstrated its practical utility in the analyzed real-use situations with respect to the criterion flexibility by definition. Several in-depth analyses and discussions with executives and employees supported that realizing the identified concurrent execution of actions (e.g., in nested parallelizations) was feasible and beneficial based on economic criteria and resource constraints. After workshops with the involved organizational units of the financial services provider, selected execution routes including the concurrent execution were applied. In this way, total required execution times were reduced, resource utilization was increased and errors and claims could be reduced. In these real-use situations, an improved decision support provided by our approach was realized.

5 Conclusion, Limitations and Further Research

In this paper, we introduced concepts stating how to construct parallel splits and synchronizations in newly planned process models in an automated manner. Compared to existing works, our approach supports the construction of all feasible parallelizations in a process model, including complex parallelizations such as nested parallelizations. Based on our formally defined concepts, we presented a concrete algorithm for this task. We implemented the approach into a software prototype to show its applicability. Moreover, the presented approach allows the consideration of large data types and planning independently of a concrete modeling language. This means that applicability for various notations such as UML activity diagrams, BPMN diagrams and Event-driven Process Chains is supported.

The main findings from our research for control flow pattern theory are as follows. To begin with, the presented concepts support the foundations of control flow pattern theory regarding the patterns parallel split and synchronization and allow to show that both patterns can indeed be constructed feasibly and in an automated manner. Second, the theoretical understanding of parallel splits and synchronizations was furthered, compared to existing approaches: Thereby, interestingly, it was proven that for two or more actions to be parallelized, other actions have to be analyzed as well (due to potential transitive dependency). Third, we showed that actions which are directly or transitively dependent cannot be parallelized. This adds rigor to statements prevalent in literature that actions may not be "in conflict" or similar descriptions (cf., e.g., Weber et al., 2010). Fourth, it was proven that in contrast to existing concepts (e.g., based on the order of actions), the absence of dependency is indeed a sufficient criterion for actions to be parallelized.

Building on these insights, our work offers major findings for the research field automated planning of process models. We believe that by addressing the presented research gap, it significantly expands the boundaries of the research field. In particular, the proposed concrete algorithm for an automated construction of all feasible parallelizations in newly planned process models forms an indispensable component of a comprehensive approach for an automated planning of process models.

Additionally, there are implications for applying our approach in practice as well. Parallelizations are, amongst other purposes, used to reduce execution times and costs while increasing workload efficiency and resource utilization. This optimization potential can be leveraged by applying our approach which allows the construction of additional parallelizations, thus increasing flexibility by definition. In this way, our approach provides valuable decision support. To reflect such implications in more detail: First, proposing alternative feasible parallelizations opens the door for discussions with process managers and executives as specific and detailed models are on the table, which can be explored and assessed regarding their organizational feasibility. Second, such discussions and what-if scenarios are in particular very fruitful – as the experiences in our cooperations show – in cases where existing process models have to be adapted to new company-internal or external (e.g., new regulations) requirements. Third, because the run times to plan models were short, some preconditions and effects of actions, especially the ones which specify resources and organizational responsibilities, could be altered. In this way, new ways and alternatives to overcome traditional organizational constraints could be provided. Fourth, when process models are realized by (web) services, our approach can provide valuable input. For instance, the process models constructed by our approach can be used by service selection approaches. This means, planned process models including different feasible parallelizations can be assessed regarding both their potential service implementation and resulting Quality-of-service values (e.g., overall cost or availability) which supports to choose beneficial execution routes (cf., e.g., Bortlik et al., 2018; Heinrich and Mayer, 2018).

However, our research also possesses some limitations that should be addressed in future work. First, our approach constructs parallelizations for planning graphs without cycles (cf. Definitions 4 and 5). This limitation could be resolved by analyzing the (sub)paths within a cycle once and separately, allowing the construction of parallelizations while considering arbitrary cycles. Further advanced control flow patterns and their combination with parallelizations have to be examined in a similar way. Second, when applying the approach in real-use situations, noisy preconditions or effects may occur and influence dependencies between actions. To address this issue, multiple plannings with different preconditions and/or effects of respective actions can be initiated. Based on this, it can be evaluated whether the noise influences the resulting process model and a feasible process model can be chosen. Third, paths consisting of ordered actions as input can be provided by multiple approaches. Thus, work should be carried out to transfer our approach to related research fields such as web service composition and process model verification which may also benefit from our work. For instance, currently we work on an enhancement of an existing (web) service composition and selection approach by considering feasible parallelizations of services during runtime of a process. Moreover, future work should analyze how our approach can be applied to manually constructed process models to allow the construction of (additional) parallelizations for such models. Our approach forms an appropriate foundation for this as well as for the aforementioned enhancements and thus serves as a suitable basis for further research.

6 References

Alrifai, M., T. Risse and W. Nejdl (2012). "A hybrid approach for efficient Web service composition with end-to-end QoS constraints" *ACM Transactions on the Web (TWEB)* 6 (2), 7.

Arora, S. and B. Barak (2009). *Computational complexity: a modern approach:* Cambridge University Press.

Augusto, A., R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella and A. Soo (2018). "Automated discovery of process models from event logs: Review and benchmark" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 31 (4), 686–705.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2001). "Planning in nondeterministic domains under partial observability via symbolic model checking". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*: Morgan Kaufmann, pp. 473–478.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2006). "Strong planning under partial observability" *Artificial Intelligence* 170 (4), 337–384.

Bertoli, P., M. Pistore and P. Traverso (2010). "Automated composition of web services via planning in asynchronous domains" *Artificial Intelligence* 174 (3), 316–361.

Binder, W., I. Constantinescu and B. Faltings (2009). "Service invocation triggers: a lightweight routing infrastructure for decentralised workflow orchestration" *International Journal of High Performance Computing and Networking* 6 (1), 81–90.

Bonet, B. and H. Geffner (2001). "GPT: a tool for planning with uncertainty and partial information". In: *Proceedings of the Workshop on Planning with Uncertainty and Partial Information (ICAI 2001)*, pp. 82–87.

Bortlik, M., B. Heinrich and M. Mayer (2018). "Multi User Context-Aware Service Selection for Mobile Environments" *Business & Information Systems Engineering (BISE)* 60 (5), 415–430.

Chang, J. F. (2016). *Business process management systems. Strategy and implementation:* CRC Press.

Cheikhrouhou, S., S. Kallel, N. Guermouche and M. Jmaiel (2015). "The temporal perspective in business process modeling. A survey and research challenges" *Service Oriented Computing and Applications* 9 (1), 75–85.

Cobham, A. (1965). "The intrinsic computational difficulty of functions". In: *Proceedings of the 1964 Congress for Logic, Methodology, and the Philosophy of Science*: North Holland Publishing Co., pp. 24–30.

Constantinescu, I., B. Faltings and W. Binder (2004). "Large scale, type-compatible service composition". In: *IEEE International Conference on Web Services*, pp. 506–513.

Cook, J. E. and A. L. Wolf (1998). "Event-based detection of concurrency". In: *ACM SIGSOFT Software Engineering Notes*, pp. 35–45.

Davenport, T. H. (1993). *Process innovation: reengineering work through information technology:* Harvard Business Press.

Deokar, A. V. and O. F. El-Gayar (2011). "Decision-enabled dynamic process management for networked enterprises" *Information Systems Frontiers* 13 (5), 655–668.

Ghallab, M., D. Nau and P. Traverso (2004). *Automated planning: theory & practice:* Elsevier.

Haerder, T. and A. Reuter (1983). "Principles of transaction-oriented database recovery" *ACM Computing Surveys (CSUR)* 15 (4), 287–317.

Hasić, F., J. de Smedt and J. Vanthienen (2018). "Augmenting processes with decision intelligence: Principles for integrated modelling" *Decision Support Systems (DSS)* 107, 1–12.

He, Q., J. Yan, H. Jin and Y. Yang (2008). "Adaptation of web service composition based on workflow patterns". In: *International Conference on Service-Oriented Computing (ICSOC 2008)*, pp. 22–37.

Heinrich, B., M. Bolsinger and M. Bewernik (2009). "Automated planning of process models: the construction of exclusive choices". In: *Proceedings of the 30th International Conference on Information Systems (ICIS 2009).*

Heinrich, B., M. Klier and S. Zimmermann (2012). "Automated Planning of Process Models-Towards a Semantic-based Approach". In *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pp. 169–194: IGI Global.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models. Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

Heinrich, B. and M. Mayer (2018). "Service selection in mobile environments: considering multiple users and context-awareness" *Journal of Decision Systems (JDS)* 27 (2), 92–122.

Heinrich, B., A. Schiller and D. Schön (2018). "The cooperation of multiple actors within process models: an automated planning approach" *Journal of Decision Systems (JDS)* 27 (4), 238–274.

Heinrich, B. and D. Schön (2015). "Automated Planning of Context-aware Process Models". In: *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*.

Heinrich, B. and D. Schön (2016). "Automated Planning of Process Models: The Construction of Simple Merges". In: *Proceedings of the 24th European Conference on Information Systems (ECIS 2016)*.

Henneberger, M., B. Heinrich, F. Lautenbacher and B. Bauer (2008). "Semantic-based Planning of Process Models". In: *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2008)*, pp. 1677–1689.

Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design science in information systems research" *MIS Quarterly* 28 (1), 75–105.

Hoffmann, J., I. Weber and F. M. Kraft (2012). "SAP speaks PDDL: Exploiting a software-engineering model for planning in business process management" *Journal of Artificial Intelligence Research* 44, 587–632.

Hwang, S.-Y. and W.-S. Yang (2002). "On the discovery of process models from their instances" *Decision Support Systems (DSS)* 34 (1), 41–57.

IEEE Task Force on Process Mining (2012). "Process mining manifesto". In: *Business Process Management Workshops*: Springer Berlin Heidelberg, pp. 169–194.

Jin, T., J. Wang, Y. Yang, L. Wen and K. Li (2016). "Refactor business process models with maximized parallelism" *IEEE Transactions on Services Computing* 9 (3), 456–468.

Kummer, T.-F., J. Recker and J. Mendling (2016). "Enhancing understandability of process models through cultural-dependent color adjustments" *Decision Support Systems (DSS)* 87, 1–12.

Lautenbacher, F., T. Eisenbarth and B. Bauer (2009). "Process model adaptation using semantic technologies". In: *13th Enterprise Distributed Object Computing Conference Workshops (EDOCW 2009)*, pp. 301–309.

Lécué, F., A. Delteil, A. Léger and O. Boissier (2009). "Web service composition as a composition of valid and robust semantic links" *International Journal of Cooperative Information Systems* 18 (1), 1–62.

Lemos, A. L., F. Daniel and B. Benatallah (2016). "Web service composition: a survey of techniques and tools" *ACM Computing Surveys (CSUR)* 48 (3), 33.

Madhusudan, T. and N. Uttamsingh (2006). "A declarative approach to composing web services in dynamic environments" *Decision Support Systems (DSS)* 41 (2), 325–357.

Meyer, H. and M. Weske (2006). "Automated service composition using heuristic search". In *Business Process Management*, pp. 81–96: Springer.

Migliorini, S., M. Gambini, M. La Rosa and A. H. M. ter Hofstede (2011). *Pattern-based evaluation of scientific workflow management systems*. Technical Report. Queensland University of Technology.

Nunamaker, J. F., M. Chen and T. D. M. Purdin (1991). "Systems development in information systems research" *Journal of Management Information Systems (JMIS)* 7 (3), 89–106.

Omer, A. M. (2011). "A framework for Automatic Web Service Composition based on service dependency analysis". Dissertation. TU Dresden.

Omer, A. M. and A. Schill (2009). "Web service composition using input/output dependency matrix". In: *Proceedings of the 3rd Workshop on Agent-oriented Software Engineering Challenges for Ubiquitous and Pervasive Computing*, pp. 21–26.

Pathak, J., S. Basu, R. Lutz and V. Honavar (2006). "Parallel Web Service Composition in MoSCoE: A Choreography-based Approach". In: *4th IEEE European Conference on Web Services*, pp. 3–12.

Pistore, M., P. Traverso, P. Bertoli and A. Marconi (2005). "Automated synthesis of composite BPEL4WS web services". In: *IEEE International Conference on Web Services*, pp. 293–301.

Prat, N., I. Comyn-Wattiau and J. Akoka (2015). "A taxonomy of evaluation methods for information systems artifacts" *Journal of Management Information Systems (JMIS)* 32 (3), 229–267.

Rathore, M. and U. Suman (2015). "An Inheritance based Service Execution Planning Approach using Bully Election Algorithm". In: *Proceedings of the International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*.

Reijers, H. A. and J. Mendling (2011). "A study into the factors that influence the understandability of business process models" *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 41 (3), 449–462.

Russell, N., A. H. M. ter Hofstede, W. M. P. van der Aalst and N. Mulyar (2006). "Workflow control-flow patterns: A revised view" *BPM Reports, BPMcenter.org* (0622).

Russell, N., W. M. P. van der Aalst and A. H. M. ter Hofstede (2016). *Workflow Patterns. The Definitive Guide:* MIT Press.

Siha, S. M. and G. H. Saad (2008). "Business process improvement. Empirical assessment and extensions" *Business Process Management Journal (BPMJ)* 14 (6), 778–802.

Soffer, P., Y. Wand and M. Kaner (2015). "Conceptualizing routing decisions in business processes. Theoretical analysis and empirical testing" *Journal of the Association for Information Systems (JAIS)* 16 (5), 345.

Sun, Y. and P. B. Kantor (2006). "Cross-Evaluation. A new model for information system evaluation" *Journal of the American Society for Information Science and Technology* 57 (5), 614–628.

Sycara, K., M. Paolucci, A. Ankolekar and N. Srinivasan (2003). "Automated discovery, interaction and composition of semantic web services" *Journal of Web Semantics* 1 (1), 27–46.

van der Aalst, W. M. P. (2012). "Process mining: Overview and opportunities" *ACM Transactions on Management Information Systems (TMIS)* 3 (2), 7.

van der Aalst, W. M. P. (2013). "Business process management. A comprehensive survey" *ISRN Software Engineering* (507984).

van der Aalst, W. M. P. (2016). *Process mining. Data science in action.* Second edition: Springer.

van der Aalst, W. M. P., A. P. Barros, B. Kiepuszewski and A. H. M. ter Hofstede (2003). "Workflow patterns" *Distributed and Parallel Databases* 14 (1), 5–51.

van der Aalst, W. M. P. and A. H. M. ter Hofstede (2005). "YAWL. Yet another workflow language" *Information Systems* 30 (4), 245–275.

van der Aalst, W. M. P., T. Weijters and L. Maruster (2004). "Workflow mining: Discovering process models from event logs" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 16 (9), 1128–1142.

Vanitha, V., V. Palanisamy and K. Baskaran (2012). "Automatic Service Graph Generation for Service Composition in Wireless Sensor Networks" *Procedia Engineering* 30, 591–597.

Venable, J., J. Pries-Heje and R. Baskerville (2012). "A comprehensive framework for evaluation in design science research". In: *Proceedings of the 7th International Conference on Design Science Research in Information Systems: Advances in Theory and Practice (DESRIST 2012)*, pp. 423–438.

vom Brocke, J. and J. Mendling (eds.) (2018). *Business Process Management Cases. Digital Innovation and Business Transformation in Practice:* Springer International Publishing.

Wang, J. and A. Kumar (2005). "A framework for document-driven workflow systems". In: *International Conference on Business Process Management*, pp. 285–301.

Weber, I., J. Hoffmann and J. Mendling (2010). "Beyond soundness: on the verification of semantic business process models" *Distributed and Parallel Databases* 27 (3), 271–343.

Weijters, A., W. M. P. van der Aalst and A. A. De Medeiros (2006). "Process mining with the heuristics miner-algorithm" *Technische Universiteit Eindhoven, Tech. Rep. WP* 166, 1–34.

Wen, L., W. M. P. van der Aalst, J. Wang and J. Sun (2007). "Mining process models with non-free-choice constructs" *Data Mining and Knowledge Discovery* 15 (2), 145–180.

Wen, L., J. Wang, W. M. P. van der Aalst, B. Huang and J. Sun (2009). "A novel approach for process mining based on event types" *Journal of Intelligent Information Systems* 32 (2), 163–190.

Wetzstein, B., Z. Ma, A. Filipowska, M. Kaczmarek, S. Bhiri, S. Losada, J.-M. Lopez-Cob and L. Cicurel (2007). "Semantic Business Process Management: A Lifecycle Based Requirements Analysis". In: *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) in conjunction with the 3rd European Semantic Web Conference (ESWC 2007)*, pp. 7–17.

Wynn, M. T., H. M.W. Verbeek, W. M. P. van der Aalst, A. H. M. ter Hofstede and D. Edmond (2009). "Business process verification-finally a reality!" *Business Process Management Journal (BPMJ)* 15 (1), 74–92.

Xu, Y., J. Yin, S. Deng, N. N. Xiong and J. Huang (2016). "Context-aware QoS prediction for web service recommendation and selection" *Expert Systems with Applications* 53, 75–86.

Supplement

A. Analysis of the Concepts

Theorem 1: Let $(bs_{init}, a_1, bs_2, a_2, \ldots, a_n, bs_{n+1})$ be a path in the planning graph and let a_i and a_j be actions in this path with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1, \ldots, n-1\}$, $j \in \{2, \ldots, n\}$.

a) If a_j is directly dependent on a_i (i.e., $a_i \dashleftarrow a_j$), a_i and a_j cannot be parallelized.

b) If a_j is *not* directly dependent on a_i and $j = i + 1$ (i.e., a_j is *directly* succeeding a_i), a_i and a_j can be parallelized.

Proof. a) Let a_j be directly dependent on a_i (cf. Definition 6). Then, at least one of the following three statements has to be true:

(1) $v(effects(a_i)) \cap v(precond(a_j)) \neq \emptyset$

(2) $v(effects(a_i)) \cap v(effects(a_j)) \neq \emptyset$

(3) $v(effects(a_j)) \cap v(precond(a_i)) \neq \emptyset$

Let us assume that the actions a_i and a_j can indeed be parallelized. Then, a concurrent execution of the actions needs to be possible (Russell et al., 2016). However, either of these statements leads to a contradiction to the ACID-principle isolation and thus to inconsistencies. Statement (1) induces a read-write collision when executing both actions concurrently, and so does statement (3). Similarly, statement (2) causes a write-write collision. Hence, a_i and a_j cannot be parallelized.

b) The action a_i is applicable in the belief state bs_i. After applying a_i, the belief state bs_{i+1} is reached in which a_j is applicable. This state bs_{i+1}, generated by means of the transition function, is based on the belief state bs_i and the effects of the applied action a_i. To be more precise, a belief state tuple whose variable is not contained in the effects of a_i remains unchanged, while if it is contained in the effects of a_i it is changed accordingly. When a_j has been carried out, we obtain in an analogous way the belief state bs_{i+2}. Let a_j be not directly dependent on a_i (cf. Definition 6). It follows that $v(effects(a_i)) \cap v(precond(a_j)) = \emptyset$. This shows that $precond(a_j)$ consists only of belief state tuples whose variable is not contained in the effects of a_i, and such belief state tuples do not change in the transition from bs_i to bs_{i+1}. Therefore, a_j can be applied in the belief state bs_i (since it is applicable in bs_{i+1}). We obtain a belief state bs' by the value of the transition function $R(bs_i, a_j)$.

From Definition 6 it further follows that $v(effects(a_j)) \cap v(precond(a_i)) = \emptyset$. Thus, analogous to the argument above, $precond(a_i)$ consists only of belief state tuples whose variables are not contained in the effects of a_j and these belief state tuples do not change via the transition

from bs_i to bs'. Hence, a_i can be applied in the belief state bs' (since it is applicable in bs_i). By means of the value of the transition function $R(bs', a_i)$, we obtain a belief state bs''.

Furthermore, it is also feasible to execute the actions $a_i$and a_j concurrently: Both actions are applicable in bs_i and because of the definition of direct dependency (Definition 6), no read-write or write-write collision occurs which would result in an inconsistency when taking the ACID-principle isolation into account.

To fulfil the criteria for parallelized actions stated by Russell et al. (2016), it remains to show that the order in which the actions a_i and a_j are executed has no effect on the resulting belief state. To this end, we consider an arbitrary belief state tuple of bs_i. By Definition 6, its variable cannot be contained in both the effects of a_i and a_j. Thus, there are three possibilities: Either its variable is not contained in the effects of a_i and a_j, or it is contained in the effects of a_i but not of a_j, or it is contained in the effects of a_j but not of a_i. In either case, the order in which a_i and a_j are executed has no influence on the resulting belief state, leading to $bs_{i+2} = bs''$ (and this is also equal to the resulting belief state when executing concurrently).

■

Theorem 2: Let p =(bs_{init},a_1,bs_2,a_2,...,a_n,bs_{n+1}) be a path in the planning graph and let a_i and a_j be actions in p with $i < j$ (i.e., a_j is succeeding a_i), $i \in \{1, \dots, n-2\}$, $j \in \{3, \dots, n\}$.

a) If a_j is transitively dependent on a_i, the actions a_i and a_j cannot be parallelized based on p.

b) If a_j is neither directly nor transitively dependent on a_i, the actions a_i and a_j can be parallelized.

Proof. a) By the definition of a transitive dependency (Definition 7), there exists a set $A_k \neq \emptyset$, $A_k = \{a_{k_1}, \dots, a_{k_m}\} \subseteq \{a_{i+1}, \dots, a_{j-1}\}$ in p such that $a_i \dashleftarrow a_{k_1} \dashleftarrow \dots \dashleftarrow a_{k_m} \dashleftarrow a_j$. When trying to rearrange the actions to enable the parallelization of a_i and a_j, the order of a_i and a_{k_1} is fixed: Because they are directly dependent, a_i and a_{k_1} cannot be parallelized (cf. Theorem 1a)), and applying a_{k_1} before a_i would be a violation to the requirement that the parallelization is supposed to be possible in p. The same argument can be made for a_{k_1} and a_{k_2}, ..., up to a_{k_m} and a_j (because of the respective direct dependencies between these actions). These facts put together show the necessity to apply a_j after a_i. Thus, because of their fixed order, a_i and a_j cannot be parallelized based on p (Russell et al., 2016).

b) Denote by X the set of actions which contains every action between a_i and a_j, including a_i and a_j. Denote by $D \subset X$ the subset which contains a and every action that is directly or transitively dependent on a_i, and by $X \backslash D$ its complement.

The basic idea of the proof is to construct a feasible path p' in which a_i and a_j are directly succeeding each other, and to then apply Theorem 1b).

1. The first part of p' is a copy of p, until (but excluding) the action a_i. Denote the belief state (which coincides in both paths) after the last action of this part with bs^0.

2. For the second part: The actions in $X\backslash D$ are applied in the same order as in p. This is possible: Denote by $S \subset X\backslash D$ the set of actions for which this is not feasible, and let $s \in S$ be the first such action. This means that the preconditions of s are not met in the belief state in which s should occur in p'. Because of the structure of the construction, this means that an action $d \in D$ (that has now been left out in p') must have altered the respective belief state variable in p, preceding s there. So in p, s is directly dependent on d, leading to $s \in D$, contradiction. Therefore, S is empty.

3. Denote the belief state directly after the execution of the last action of $X\backslash D$ (which is a_j) with bs^1. The action a_i is applicable in bs^1 (it was applicable in bs^0 and no action in p' after bs^0 until bs^1 can have the effect of changing a variable which is part of the preconditions (or effects) of a_i).

4. Now the rest of the actions in D can be applied, in the same order as in p.

Proof: Denote by $T \subset D$ the set of actions for which this is not feasible, and let $t \in T$ be the first such action. This means that the preconditions of t are not fulfilled in the belief state bs^t in which t should occur in p'. Consider a belief state variable z which has a value that causes t to be not applicable in bs^t. Now, look for the last action in p after bs^0 and before the execution of t which alters this variable z.

Case 1: There is no such action. Then, the aforementioned situation can only occur when there is an action x in $X\backslash D$ that changes z and is applied after t in p (it is obviously applied prior to t in p'). This, however, leads to $t \dashleftarrow x$ in p and therefore to a contradiction to $x \in X\backslash D$.

Case 2: There is such an action. Denote it by q. Note that necessarily $q \in X\backslash D$ because otherwise q would also certainly be the last action in p' before bs^t changing z, and that q has definitely already been applied in p', because all actions in $X\backslash D$ are applied before the first application of an action in D.

Case 2.1: After q but before bs^t, an action $r \in D$ has changed z in p'. Certainly, in p, r is applied prior to q. Thus, $z \in v(effects(r)) \cap v(effects(q))$ leads to $r \dashleftarrow q$ in p and to a contradiction to $q \in X\backslash D$.

Case 2.2: After q, an action $r \in X\backslash D$ has changed z in p'. However, q was defined to be the last action in p after bs^0 and before the execution of t which alters this variable z, so that in p, t is executed before r, leading to $t \dashleftarrow r$ and a contradiction to $r \in X\backslash D$ because $t \in D$.

Considering all cases we can conclude $T = \emptyset$.

5. The belief state in p' after the application of the last action in D coincides with the belief state in p after a_j:

The belief state bs^0 is the same. If, afterwards, two actions are both in D or in $X \backslash D$, their order is the same in p and in p'. If one action d is in D and the other action x is in $X \backslash D$, their order might differ in p and p': It is possible that d is applied before x in p, but applied after x in p'. However, they cannot change the same variable (otherwise $d \dashleftarrow x$ in p, so $x \in D$).

Thus the rest of the actions can be applied just as in p and, when completed, the path p' is just the path p with possibly a reordering of a few (non-dependent) actions.

In p', a_i and a_j are directly succeeding each other and therefore, application of Theorem 1b) delivers the desired result.

■

Theorem 3 (*completeness*): Let G be a planning graph consisting of the paths p_1,...,p_k. Suppose the actions $a_1, \ldots, a_n$ represented in G can be parallelized. By analyzing direct and transitive dependencies in all paths p_1,...,p_k, the parallelization of a_1,...,a_n is constructed.

Proof. As a_1,...,a_n can be parallelized, a feasible path p in which all actions in $S = \{a_1, \ldots, a_n\}$ are parallelized needs to result. We show that this path p is indeed constructed based on the analysis of direct and transitive dependencies: A path p' exists in the planning graph such that p' equals p with the exception that the actions in S are planned in sequence instead of being parallelized (the exact order of the actions in S in p' is not important for the further argumentation). Consider any pair of actions $(a_i, a_j) \in S \times S, i \neq j$. Let (w. l. o. g.) a_j be succeeding a_i in p'. Certainly, the action a_j is not directly dependent on a_i (otherwise, a_i and a_j could not be parallelized). As this holds for any such pair in $S \times S$, a_j is also not transitively dependent on a_i in p'. Thus, our concepts allow parallelizing a_i and a_j. Since (a_i, a_j) was an arbitrary pair of actions in $S \times S$ with $i \neq j$, parallelization of all actions in S is allowed in the path p'. Thus, analyzing direct and transitive dependencies in p' results in the construction of the path p in which all actions in S are parallelized.

■

B. Analysis of the Algorithm Properties

Termination: The algorithm terminates.

Proof. To see that our algorithm terminates, one has to keep two facts in mind:

(F1) The planning graph consists of a finite number of paths

(F2) Each path of the planning graph contains only a finite number of actions

Neither the number of paths in the planning graph nor the number of actions in a particular path is changed in the course of the algorithm.

Since the statements in line 6 and 10 (cf. pseudo code in *Appendix E*) terminate obviously, we need to begin our analysis with the for-loop starting in line 11. Because of (F1) and (F2), the number of iterations of the for-loops in line 11 and 12 is finite. Each iteration terminates because only simple set operations are made, so that altogether the for-loop in line 11 terminates.

The for-loop starting in line 22 terminates due to (F1) and the simple operations in its iterations.

Now we consider the for-loop starting in line 30. It again has only finitely many iterations because of (F1). AllActions.length is finite as well due to (F1) and (F2), so that the for-loop starting in line 35 is only invoked a finite number of times. The following statements terminate obviously. The for-loop initiated in line 45 once again has a finite number of iterations because of (F1), and contains only simple operations, leading to its termination. Since line 56-57 terminate evidently, the whole for-loop starting in line 30 terminates.

Finally we need to analyze the for-loop in line 66. Because of (F1) and (F2), the number of iterations of the three for-loops starting in line 66, 68 and 70 is finite. The next statement we need to consider is the for-loop starting in line 85, which has only finitely many iterations for the same reason. The rest of the statements terminate obviously, leading to the termination of the for-loop starting in line 66 and therefore the termination of the whole algorithm.

■

Correctness/Completeness: The algorithm leads to complete and correct parallelization matrices: Every required entry is inserted and there is no entry that would allow an infeasible parallelization or prohibit a feasible parallelization.

Proof. It suffices to show the result for an arbitrary path p of the planning graph.

At first, the algorithm searches for direct dependencies between all actions in p. If a direct dependency is found, the algorithm inserts the symbol $ddep$ in the entries (rows and columns corresponding to the pair of actions) in the parallelization matrix M_p (and the parallelization matrices of following paths which contain both actions), regardless of whether the actions are directly succeeding each other or not (line 50 in the algorithm). The symbol $ddep$ prohibits parallelization (cf. Theorem 1a)). If no direct dependency is found, the algorithm inserts the symbol $para$ in the corresponding entry, if the considered actions are directly succeeding each

other (line 57). This means that parallelization of the actions is allowed in p, and the correctness of this statement is proved in Theorem 1b).

Afterwards, the algorithm attempts to complete M_p by looking for transitive dependencies between actions. Only pairs of actions that are applied in p but do not yet have an entry in M_p need to be considered. If a transitive dependency is found, the symbol $tdep$ is inserted (line 91), prohibiting parallelization as justified by Theorem 2a).

If, after analyzing all possible transitive dependencies, a pair of actions in p still does not have an entry in M_p (that is, the actions are not directly succeeding each other and they are neither directly nor transitively dependent in p), the symbol $para$ is inserted (line 101). This symbol allows parallelization of the actions, following the result of Theorem 2b).

As the created parallelization matrices are correct and complete, this holds also for the parallel splits and synchronizations constructed from the parallelization matrices.

■

Computational Complexity: We consider a graph with $|P|$ paths, where each path has n actions and each action has m preconditions and m effects. We assume the worst-case-scenario: This occurs when there are no direct dependencies at all. Evidently, in this case, no transitive dependencies exist as well. These conditions imply the worst-case-scenario because in each analysis of a direct dependency, the maximum amount of comparisons (of belief state variables) has to be executed. Additionally, they imply that whenever analyzing a potential transitive dependency, the maximum amount of possibilities has to be checked. In the following, we only consider a single path. To obtain the corresponding results for the complete planning graph, one just has to multiply by $|P|$. The multiplication by this constant factor has no impact when checking the computational complexity of our algorithm via O-notation.

Analysis of direct dependencies per path:

There are $\binom{n}{2} = \frac{n(n-1)}{2}$ pairs of actions that need to be checked.

For every pair (a_i, a_j) of actions, the algorithm compares the belief state variables of $effects(a_i)$ with the belief state variables of $effects(a_j)$, the belief state variables of $effects(a_i)$ with the belief state variables of $precond(a_j)$ and the belief state variables of $precond(a_i)$ with the belief state variables of $effects(a_j)$ and has to execute m^2 comparisons in each case, adding up to $3m^2$ comparisons in total.

Therefore, the total amount of executed comparisons per path is $3\left(\frac{1}{2}n^2 - \frac{1}{2}n\right)m^2$.

Analysis of transitive dependencies per path:

The analysis is independent of the number of variables m, as they are not considered at all.

If $n = 3$, just the pair (a_1, a_3) needs to be analyzed regarding transitive dependency.

If $n = 4$, the pairs (a_1, a_3), (a_2, a_4) and (a_1, a_4) need to be considered. Due to the two possibilities of $a_1 \dashleftarrow a_2 \dashleftarrow a_4$ and $a_1 \dashleftarrow a_3 \dashleftarrow a_4$, analysis of the pair (a_1, a_4) requires twice as much time as analysis of one of the other pairs.

Proceeding in this manner, one obtains the formula

$$\sum_{k=1}^{n-2}((n-1-k)*k) = \frac{1}{6}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n$$

for the computational effort for the transitive dependencies.

To sum it up, in the worst-case-scenario, the algorithm is in quadratic time in the number of variables and in cubic time in the number of actions (that is, $O(m^2)$ and $O(n^3)$).

C. Evaluation Results

Context	Name of the process	Number of actions and states in the planning graph	Number of paths in the planning graph	Number of constructed parallelizations (all/only nested ones)	Number of actions and states included by the parallelizations	Number of checks for direct and transitive dependencies	Required data types of the state variables of the preconditions and effects	Run time in sec.
Project management	Preparing and coordinating project profile	17/15	6	1/0	2	51/9	Boolean; String; Classes	0.002
Project management	Specifying project resources	25/18	36	3/0	7	60/948	Boolean; String; Classes	0.004
Project management	Allocating project resources	26/22	24	3/0	6	115/444	Boolean; String; Classes	0.004
Project management	Preparing the project report for management board	38/25	252	3/0	9	111/8,640	String; Classes	0.018
Insurance agency	Administrating customer and product database	43/38	52	3/0	6	266/772	Boolean; String; Classes	0.009

Table C.1. Application of our Approach in further Real-use Situations

Insurance agency	Handling insured events	54/44	60	2/0	4	206/2,912	Boolean; String; Classes	0.009
Loan management	Analyzing credit rating	40/31	197	6/2	19	123/10,598	Boolean; String; Classes	0.019
Loan management	Selling mortgage loans	57/43	216	7/0	21	116/7,128	Boolean; Double; Integer; String; Classes	0.021
Loan management	Settling mortgage loans	122/69	6,572	16/12	54	170/1,095,212	Boolean; Double; Integer; String; Classes	0.391
Private Banking	Contracting wealth management customer	278/189	1,244,416	19/4	48	2,760/ 1,407,962,480	Boolean; Integer; String; Classes	753.524
Human resources	Engaging new staff	83/75	76	12/0	31	219/7,524	Boolean; String; Classes	0.012

Table C.1. Application of our Approach in further Real-use Situations (continued)

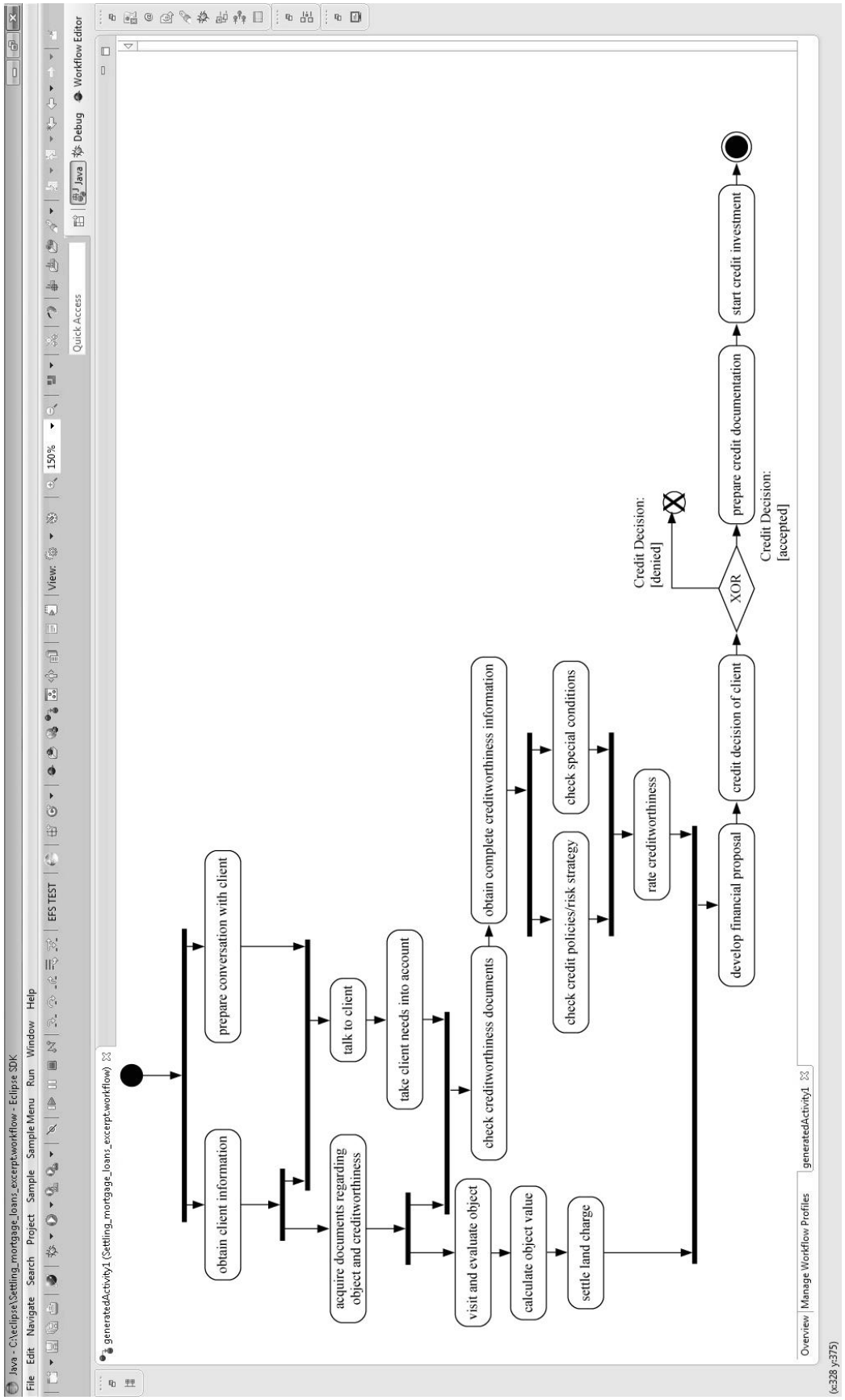

Figure C.1. Planned Model of a Part of the Settling Mortgage Loans Process (Screenshot Prototype)

D. Application to additional Real-use Situations

To further address the evaluation questions, the presented approach was applied in additional real-use situations from various application contexts and different companies as shown in Table C.1. The analyzed situations included up to 278 actions and 189 states in the planning graph and are of a medium to large size. This is also reflected in the number of paths of the different planning graphs which ranges up to over 1.2 million (due to the various orders the actions can appear in) in the case of the "Contracting wealth management customer" process. Our algorithm was able to cope with the required data types and could be applied in all situations without restrictions. The run time of the algorithm varied – depending on the size and complexity of the processes - from a few milliseconds up to around 12.5 minutes. The very moderate run times show that our approach is also viable for constructing parallelizations in large process models. When analyzing the available manually designed process models in the real-use situations, each parallelization contained in these process models corresponded to a feasible parallelization constructed by our algorithm in an automated manner. Furthermore, for almost all of the processes, our algorithm constructed additional feasible parallelizations not contained in the manually designed process models. All real-use situations incorporated parallelizations, often consisting of path segments with more than one action, and a significant number of the situations also contained nested parallelizations. This fact illustrates that larger, complex parallelizations are frequently occurring and we addressed a vital component of process models with our approach. To give an example, the process model depicted in Figure C.1 shows an excerpt of the "Settling mortgage loans" process. It contains a rather complex parallelization which is not easy to design manually but was constructed correctly and within one second by our prototype. In the initial belief state, the actions *obtain client information* and *prepare conversation with client* are applicable and can be parallelized. The latter action can also be parallelized to *acquire documents regarding object and creditworthiness*, which in turn can be parallelized to *talk to client* and *take client needs into account*. *Talk to client* and *take client needs into account* can also be parallelized to *visit and evaluate object*, *calculate object value* and *settle land charge*. As shown in the figure, the actions *check creditworthiness documents* and *obtain complete creditworthiness information*, the actions *check credit policies/risk strategy* and *check special conditions* in a nested parallelization and *rate creditworthiness* can also be parallelized to *visit and evaluate object*, *calculate object value* and *settle land charge*. The subsequent actions are executed in sequence. Overall, the parallelization contains path segments of length greater than one (e.g., *visit and evaluate object*, *calculate object value* and *settle land charge* is a path segment of length three) and incorporates different nested parallelizations, which are constructions not yet addressed by existing approaches.

E. Pseudocode of our Algorithm

```
Require: Graph G with m Paths p[1] … p[m] and p.length sequential actions
a[1][1],a[2][1]...,a[p.length][m]
function parallelize(G){
   //step 1); generate list in which each action in the graph occurs exactly once
   (will be used to determine rows in position matrix)
   Vector allActions := new Vector()
   //generate position matrix with rows=all actions of graph (no duplicates) and
   columns=path numbers. Cells=Position of every action in every path (for example,
   action "b" is in path p[3] at position 4)
   [][] positionMatrix:= new int [#actionsInGraph][m]
   for all p ∈ (1 ≤ p ≤ m) do
      for all i ∈ (1 ≤ i ≤ p.length) do
         if (a[i][p] ∉ allActions) then
            allActions.add(a[i][p])
         end if
         positionMatrix[allActions.getIndex(a[i][p])][p] = i
      end for
   end for
   //step 2); generate vector which contains for each path in G a
   parallelization matrix
   Vector ParaMatrices:= new Vector()
   for all p ∈ (1 ≤ p ≤ m) do
   //generate parallelization matrix with rows=columns=all actions of graphs. Cells
   will contain direct or transitive dependencies between actions.
     [][]ParaMatrix:= new String[allActions.length][allActions.length]
     ParaMatrices.insertElementAt(ParaMatrix, p)
   end for
   for all p ∈ (1 ≤ p ≤ m) do
   // step 3); examine direct dependencies
   for all i ∈ (2 ≤ i ≤ allActions.length) do
      //skip actions which do not occur in considered path p
      if (positionMatrix[i][p]=0) then
         continue
      end if
      for all j ∈ (i-1 ≥ j ≥ 1) do
         if (positionMatrix[j][p]=0) then
            continue
         end if
         //skip in case of previously identified direct dependencies
   if (ParaMatrices.elementAt(p).[i][j] ≠ ddep) then
    //mark all direct dependencies between compared actions in all suc-
   ceeding paths / parallelization matrices with "ddep"
         if(v(effects(a[i]))∩(v(precond(a[j]))∪v(effects(a[j])))≠∅
             ∨ v(precond(a[i]))∩v(effects(a[j]))≠∅) then
            for all a ∈ (p ≤ a ≤ m) do
               //skip actions which do not occur in considered path p
               if (positionMatrix[i][a]=0 ∨positionMatrix[j][a]=0) then
                  continue
               end if
               ParaMatrices.elementAt(a).[i][j] ← ddep
            end for
```

```
//if the two compared actions are connected and not directly dependent, they can be
marked with "parallelize" (="para") and be
parallelized in this path
          else
             if (|positionMatrix[i][p]-positionMatrix[j][p]| = 1) then
                ParaMatrices.elementAt(p).[i][j]←para
             end if
          end if
        end if
      end for
   end for
end for
//step 4); identify and mark all transitive dependencies in parallelization matrix
with "tdep" (path-wise)
for all p ∈ (1 ≤ p ≤ m) do
   //for all actions in a path compare actions
   for all i ∈ (3 ≤ i ≤ p.length) do
  //j needs to be decreasing to guarantee the correct proceeding order
      for all j ∈ (i-2 ≥ j ≥ 1) do
//the order of actions in the path may be different to the order in the parallelization
matrix for this path (essentially in all paths but the first). Therefore, the position
in the path p has to be "translated" to the position of the action in the parallel-
ization matrix (which is identical to the position in the vector "allActions").
        pos_i:= allActions.getindex(a[i][p])
        pos_j:= allActions.getindex(a[j][p])
//since we aim at filling only half of the dependency matrix, we have to assure that
we analyze only the "correct" triangle of the matrix (the other half of the matrix
consists of zeros and is of no importance). This is done by selecting the larger index
in the first dimension of the matrix and the smaller index in the second dimension of
the matrix.
       if(ParaMatrices.elementAt(p).[Max(pos_i,pos_j)][Min(pos_i,pos_j)])
         ≠ddep) then
           for all k ∈ (i > k > j) do
              pos_k:= allActions.getindex(a[k][p])
         if((ParaMatrices.elementAt(p).[Max(pos_i,pos_k)][Min(pos_i,pos_k)])=
      (ddep ∨ tdep)) ∧
(ParaMatrices.elementAt(p).[Max(pos_j,pos_k][Min(pos_j,pos_k)]) =
(ddep ∨  tdep))) then
   (ParaMatrices.elementAt(p).[Max(pos_i,pos_j)][Min(pos_i,pos_j)])←tdep
                break for
             end if
           end for
        end if
//if the two compared actions are neither directly nor transitively dependent, they
can be marked with "parallelize" and be parallelized (in the considered path)
   if(ParaMatrices.elementAt(p).[Max(pos_i,pos_j][Min(pos_i,pos_j]≠  (ddep ∨tdep))
then
           ParaMatrices.elementAt(p).[Max(pos_i,pos_j][Min(pos_i,pos_j]←   para
        end if
      end for
   end for
end for
```

References

Russell, N., W. M. P. van der Aalst and A. H. M. ter Hofstede (2016). *Workflow Patterns*. *The Definitive Guide:* MIT Press.

4.2 Paper 7: Adapting Process Models via an Automated Planning Approach

Current Status	Full Citation
under review (since 07/2019) for publication in *Journal of Decision Systems*	Heinrich, B., A. Schiller, D. Schön and M. Szubartowicz (2019). "Adapting Process Models via an Automated Planning Approach". *Working Paper*, University of Regensburg.

Summary

This paper treats RQ7 by proposing an automated planning approach for the adaptation of process models to needs for change in advance. To this end, in a first step, all possible changes to existing process models are identified and classified. In a second step, potential consequences resulting from these identified changes are addressed. For that purpose, automated planning is used, leading to correct and complete adapted process models. The approach is evaluated by means of mathematical proofs of correctness and completeness and an application to a real-world situation in an electrical engineering company. Further, its computational complexity is assessed in an algorithmic complexity analysis and by means of a simulation experiment, in which its runtime is benchmarked against planning process models from scratch.

The work relies on concepts (e.g., belief states, nondeterministic belief state-transition systems and process graphs) representing a planning domain from AI planning very similar to Paper 6. In particular, process graphs, which can for instance be constructed using AI planning methods, form the starting point for the approach. These concepts are furthered (e.g., by extending the notion of belief state with respect to its presence in the existing process graph), forming the basis for enhanced AI planning methods (e.g., the selective use of a planning algorithms) which are used in the approach. As shown in the evaluation, adapting process models using the presented approach provides considerable runtime advantages compared to planning from scratch, is advantageous in practice and thus decidedly supports business process agility and flexibility.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style.

Abstract:

Today's fast-paced business world poses many challenges to companies. Amongst them is the necessity to quickly react to needs for change due to shifts in their competitive environment. Hence, a high flexibility of business processes while maintaining their quality has become a crucial success factor. We address this issue by proposing an automated planning approach that is capable of adapting existing process models to upcoming needs for change. This means that the needs for change have not yet been implemented and the adapted process models have so far not yet been realized. Our work identifies and addresses possible changes to existing process models. Further, it provides adapted process models, which are complete and correct. More precisely, the process models resulting from the presented approach contain all feasible and no infeasible paths. To enable an automated adaptation, the approach is based on enhanced methods from automated planning. We evaluate our approach by means of mathematical proofs and an application in a real-world situation. Further, by means of a simulation experiment, its runtime is benchmarked against planning process models from scratch.

Keywords: process flexibility, processes changes, process models, business process management

1 Introduction

The ability to be agile and align existing capabilities to new needs quickly is one of the most important factors for companies' success and competitive advantage (McElheran, 2015). Companies are required to react to shifts in their competitive environment flexibly and within short time in order to stay operational and competitive (Döhring et al., 2014; Forstner et al., 2014; Katzmarzik et al., 2012; Reisert et al., 2018; Rosemann and vom Brocke, 2015). Examples of such shifts include dynamic customer behavior, market developments or new regulatory requirements and are referred to as needs for change. According to Le Clair (2013), in the last decade the inability to react to such needs for change has led to 70% of the Fortune 1000 companies to be removed from this list. The study proposes ten dimensions to characterize business agility, half of them being process-focused. This underlines that improving the flexibility of business processes while maintaining their quality has become a crucial success factor for companies (Reichert and Weber, 2012). Here, process flexibility is understood as the ability to configure or adapt a process and the according process model without completely replacing it (Afflerbach et al., 2014; Bider, 2005; Hallerbach et al., 2010; Regev et al., 2007). It is hardly surprising that the importance of process flexibility is widely recognized in the literature (cf., e.g., Cognini et al., 2018; Ellis et al., 1995; Hammer, 2015; Hull and Motahari Nezhad, 2016; La Rosa et al., 2017; Mejri et al., 2018; van der Aalst, 2013).

Process flexibility is also acknowledged as an important issue in practice. To give an example, in an extensive project with a European bank we analyzed over 600 core business processes as

well as 1,500 support processes. These processes spread across different departments and business areas of the bank. The majority of the processes and their corresponding process models, which initially had been modeled using the ARIS toolset, required a frequent redesign or adaptation due to needs for change caused by, for instance, new or enhanced distribution channels and changing products. Indeed, the bank has been conducting projects to adapt business process models much more frequently, causing the vast majority of the budget, compared to projects to design completely new business process models. Moreover, several IT and business executives of the bank stated that nowadays process redesign projects are more time-consuming than they were ten years ago due to a higher complexity. Interviews with staff members of companies in other industries supported these insights. This underlines the relevance of approaches for an adaptation of process models.

The increasing complexity of process models and process (re)designs also has another effect: Constructing and adapting process models manually turns out to be a more and more difficult task. According to Mendling et al. (2008), especially larger and more complex process models are likely to contain more errors when constructed manually. For instance, Roy et al. (2014) and Fahland et al. (2011) examined business process models in an industrial context and found that up to 92.9% of these models contained at least one (syntactical or semantic) error. Hence, in this paper, we follow other approaches making use of automation techniques (e.g., algorithms) when modeling processes (e.g., Marrella, 2018; Rosemann et al., 2010). The research strand "automated planning of process models" (Heinrich et al., 2015; Heinrich and Schön, 2015; Henneberger et al., 2008; Hoffmann et al., 2009, 2012) aims to construct process models in an automated manner and from scratch. Here, a process model is planned by means of algorithms based on states, actions, and control flow patterns. Our approach for an automated adaptation of process models enhances methods of automated planning and thus contributes to this research strand.

Adapting process models to needs for change comprises the modeling of an existing or desired process. Modeling an existing process means that the required changes have already been realized. In this case, the changed process can be modeled by adapting or reconstructing an existing process model to new records of event logs not already considered in the existing process model (cf. process enhancement; IEEE Task Force on Process Mining, 2012; Kalenkova et al., 2017). In contrast, modeling a desired process means that the needs for change have not yet been implemented and the desired process models have so far not yet been realized. In this paper, we focus on the latter perspective, thus aiming to adapt existing process models to needs for change *in advance* and to construct models of desired processes, leading to the following research question:

How can process models be adapted to needs for change in advance in an automated manner?

In literature, many existing approaches for the adaptation of process models aim to "repair" process models locally when considering changes (e.g., Alférez et al., 2014; Eisenbarth, 2013). However, both process models and process (re)designs are becoming increasingly large and

complex (cf. Hornung et al., 2007 and the discussion above) and local repairs or changes to just some components of process models are not sufficient. Instead, the challenging task of providing adapted process models, which are *correct* and *complete*, has to be addressed. Correct means that the adapted process models contain *only* feasible paths and no infeasible paths, while complete means that the adapted process models contain *all* feasible paths. Correct and complete process models are important to, for instance, increase "flexibility by definition", which is "the ability to incorporate alternative execution paths within a process definition at design time such that selection of the most appropriate execution path can be made at runtime for each process instance" (van der Aalst, 2013, p. 25). A correct and complete process model enables the flexibility to select the most appropriate feasible path for execution (e.g., based on economic criteria). Hence, we discuss the following research question:

How can process models be adapted such that the resulting process models are correct and complete?

The main contributions of this paper are thus as follows:

(C1) *Adaptation to needs for change in advance in an automated manner.* The approach adapts existing process models to needs for change in advance (i.e., no reconstruction of existing process models, e.g., to new records of event logs). To this end, it enhances methods especially from automated planning of process models.

(C2) *Construction of correct and complete process models.* The approach adapts process models in such a way that the resulting process models are correct and complete.

In the next section, we discuss related work to explicate our research gap. Thereafter, we introduce a running example and define the formal foundation, which forms the basis of our approach. After that, we present our approach to adapt existing process models to needs for change in advance via automated planning. Subsequently, we evaluate our approach by means of mathematical proofs of its key properties, demonstrate its efficacy by means of an application in a real-world situation and benchmark its performance in a simulation experiment. We conclude by summarizing our work, discussing its limitations and proposing future research.

2 Related Work

In the following, we will discuss existing approaches dealing with an adaptation of process models. To structure this discussion, we consider five phases of the BPM lifecycle as proposed by vom Brocke and Mendling (2018) and omit the process identification phase, as it is not subject of our research. We start with approaches in (1) the process discovery phase and continue by discussing existing approaches in (2) the process analysis phase. Thereafter, we briefly analyze approaches in (3) the process re-design phase, (4) the process implementation phase and close with (5) the process monitoring and controlling phase. Table 1 at the end of the section summarizes our discussion.

Ad (1): During the process discovery phase, detailed information about processes (e.g., in terms of process models) is derived from actually conducted processes in a company. The research field of process mining addresses the area of process discovery (cf., e.g., Augusto et al., 2018; IEEE Task Force on Process Mining, 2012; van der Aalst, 2015). Within this area, approaches use event logs from process instances to reconstruct process models (van Dongen et al., 2009). The issue that the information of event logs might change from time to time due to changes in the corresponding executed process, which needs to be considered when discovering the process model, has extensively been addressed in the literature (Bose et al., 2014; Chen et al., 2012; Mei-hong et al., 2012). The focus of this research, however, is different to ours because an *existing*, already changed process is *re*constructed. This means, the aim is a reconstruction of a (new) process model considering needs for change already realized in actual process instances. Therefore, these works do not aim to provide an approach for adapting process models to changes in advance and, as they rely on event logs, do not present concepts to support this task. In contrast, we aim to model a desired process which is not yet realized and thus to adapt to needs for change in advance (cf. (C1)).

Ad (2): During the process analysis phase, for instance, weaknesses in the discovered processes are determined. In this context, the research field of process (model) and workflow verification (e.g., Masellis et al., 2017) aims to check and improve syntactic and semantic correctness of process models. For instance, the automated repair of unsound workflow nets by means of annealing procedures (i.e., heuristic approaches which generate a set of alternative workflow nets containing fewer errors) is envisioned by Gambini et al. (2011). Further, the verification of workflows by means of Petri nets is focused on by Verbeek and van der Aalst (2005) and Wynn et al. (2009) in order to "detect the soundness property". However, within this research field, there is no work on the adaptation of process models to needs for change in advance (cf. (C1)).

Ad (3): Approaches in the process re-design phase aim to increase process flexibility in the way they model business processes, for instance by capturing customizable process models (La Rosa et al., 2017). To this end, manual as well as automated (i.e., by means of an algorithm) approaches have been proposed. Generalizations (van der Aalst et al., 2009; vom Brocke, 2009) or specific change patterns (Weber et al., 2008), which are both constructed manually, provide possibilities to increase process flexibility. Generalization approaches result in less specific process models, due to, for instance, the assignment of several specific actions to one abstract, general action. Hence, such generalized process models (e.g., reference models; cf. vom Brocke, 2009) may lack support for real-world scenarios that are not modeled explicitly, especially with respect to an (automated) process execution (cf., e.g., Khan et al., 2010; Weber, 2007). On the other hand, specific change patterns allow the replacement of parts of a process model – often supported by a modeling tool – by different, predesigned parts. The purpose of those approaches is different from ours, since they do not aim to provide an approach for the automated adaptation of process models, which are correct and complete (cf. (C1) and (C2)). A second research strand in the process re-design phase striving to increase process flexibility is the automated planning of process models (cf., e.g., Heinrich et al., 2018). In this strand, few

approaches exist that address the issue of adapting process models to needs for change in advance (cf. Eisenbarth et al., 2011; Eisenbarth, 2013; Lautenbacher et al., 2009). These works adapt parts of a process model due to (a few) changed actions. They identify so called single-entry-single-exit fragments surrounding an action to be changed. Based on such an identified fragment, the idea is to determine "quasi-" initial and goal states (for the considered fragment) and to initiate a regular process planning in order to replace the existing fragment by means of a newly planned fragment. Changed actions, however, can affect the whole process model (e.g., when a changed action results in several new feasible paths), so that the process models adapted by these approaches are usually not complete. Further, these approaches do not ensure that the whole process model is correct because of adapting only fragments. Additionally, the need for adapting a process model may not only arise from actions to be changed but also from changed initial and goal states, which is not covered by this research. To sum up, these works do not aim to provide adapted process models which are complete and correct and are "interested in adapting only parts of a model" (Eisenbarth et al., 2011), in contrast to (C2).

Ad (4): During the process implementation phase, the previously (in the process re-design phase) constructed process model is implemented in the according execution systems. For instance, (web) services are composed with the aim of aggregating existing functionality into new functionality. For this, graph structures consisting of services and states, which are similar to actions and states in process models, are constructed. Thus, within the research field of (web) service composition, issues similar to the adaptation of process models are discussed as "network configurations and QoS [Quality of Service] offerings may change, new service providers and business relationships may emerge and existing ones may be modified or terminated" (Chafle et al., 2017). Here, research focuses on replacing (web) services (or small combinations of services) by other, functionally equivalent (small combinations of) services (cf., e.g., Bucchiarone et al., 2011; Canfora et al., 2005). Within this research field, some authors use so called variability models, which are very similar to the change patterns mentioned above, to adapt service compositions (Alférez et al., 2014; La Rosa et al., 2017). However, in contrast to these approaches, our considered changes regarding process models are not limited to exchanging (a few) actions but rather we aim to adapt whole process models in such a way that the resulting process models are correct and complete (cf. (C2)).

Ad (5): In the process monitoring and controlling phase, several works exist that envision to use so called continuous planning for the recovery of failed process executions (cf., e.g., Linden et al., 2014; Marrella et al., 2011; Marrella et al., 2012; Marrella and Mecella, 2011; Tax et al., 2017; van Beest et al., 2014). These works aim for error handling procedures that are based on planning techniques in order to resolve process executions interrupted due to, for instance, external events. Other works support users by providing change operations to address ad-hoc deviations from pre-modeled task sequences within a workflow (Reichert and Dadam, 1997, 1998; Rinderle et al., 2004). However, they do not propose an approach to enable the adaptation of process models (cf. (C1)). In particular, as these works aim to address particular process instances, they do not strive to provide complete process models for the business process as a

whole (cf. (C2)). Another kind of approaches (cf., e.g., Garrido et al., 2010; Gerevini et al., 2012; Kambhampati, 1997; Marrella et al., 2017; Nunes et al., 2018; Scala et al., 2015; van der Krogt et al., 2002; van der Krogt and de Weerdt, 2005) that make use of planning algorithms deals with the issue of adapting a process model due to discrepancies which occurred during the conduction. Here, the task is to find a sequence of actions that will resolve the misalignment between the modeled and the actual reality (Marrella et al., 2017). Similarly, Kambhampati (1997) introduces the concept of refinement planning as "the process of starting with the set of all action sequences and gradually narrowing it down to reach the set of all solutions". Here, so called candidates (i.e., parts of a plan consistent with certain constraints) are combined to subsequently construct a feasible complete plan. Further, Gerevini and Serina (2010), for instance, propose a fast plan adaptation by identifying delimited parts of the plan that are inconsistent and then replanning the subgraph for these delimited parts. In the worst case, these parts comprise the whole plan, making planning from scratch necessary. However, they do not aim to adapt whole process models (cf. (C2)). Further, these approaches tend to address *momentary* changes that "occur on an individual or selective basis" (van der Aalst and Jablonski, 2000). However, we aim to address both momentary and *evolutionary (permanent)* changes that "are of a structural nature" and are typically "forced by legislature or changing market demands" (e.g., van der Aalst and Jablonski, 2000). Nebel and Koehler (1995) provide an empirical analysis about the efficiency of plan reuse versus (new) plan generation. They compare the worst-case complexity of planning from scratch with reusing and modifying plans (so called planning from second principles). The authors state that planning from second principles consists of two steps: The identification of an appropriate plan candidate from a plan library and its modification so that it solves a new problem instance. As they aim for a "minimal modification of a plan", they do not strive to construct complete process models (cf. (C2)). In this regard, there exist a few declarative process modeling approaches that address similar issues as well. Declarative process models are an alternative to the (imperative) process models addressed in this paper, specifying what should be done in a process, not how (Pesic et al., 2007; Pesic and van der Aalst, 2006; van der Aalst et al., 2009). They tend to address *momentary* changes, whereas we aim to address both momentary and *evolutionary* changes (van der Aalst and Jablonski, 2000). For declarative process models, it is further proposed to generate so called "optimized enactment plans" that could be understood as a planning problem (cf., e.g., Barba et al., 2013a; Jiménez-Ramírez et al., 2013). In this context, a replanning approach is envisioned by Barba et al. (2013b), in case the actually conducted process deviates from the generated optimized enactment plan. However, they aim at "optimizing performance goals like minimizing the overall completion time" in contrast to (C2) and do not adapt to needs for change in advance (cf. (C1)).

Finally, the research field of process mining comprises the areas of conformance checking and process enhancement (IEEE Task Force on Process Mining, 2012; Leemans et al., 2018; van der Aalst, 2015) that are also part of the process monitoring and controlling phase. Conformance checking is used to detect differences between the traces of a process execution (e.g.,

found in event logs) and a given process model (Garcia-Bañuelos et al., 2017; Leoni and Marrella, 2017; van der Aalst and Verbeek, 2014). In process enhancement (which deals with tasks such as "model extension" or "model repair"), the goal is to change or extend an already existing process model by taking information about the process instances from event logs into account (cf., e.g., Fahland and van der Aalst, 2012). The focus of this research, however, is different to ours because an *existing*, already instantiated and enacted process is analyzed with respect to deviations from an *existing* process model. This means, the aim is an adaptation of a process model to needs for change already realized in actual process instances. Therefore, these works do not aim to provide an approach for adapting process models to changes in advance as they rely on event logs. In contrast, we aim to model a desired process which is not yet realized and thus to adapt to needs for change in advance (cf. (C1)).

To sum up, to the best of our knowledge, there is no existing approach that adapts process models to needs for change in advance in an automated manner (cf. (C1)) and constructs correct and complete process models (cf. (C2)).

Lifecycle phase		Time of consideration	Adaptation to upcoming needs for change in advance in an automated manner (C1)		Construction of correct and complete process models (C2)
			Adaptation to upcoming needs for change in advance	Automated approach	
1) Process discovery		design time	✗ not considered; aiming to reconstruct process models that represent already realized needs for change in processes	✓ considered	✓ considered
2) Process analysis		design time	✗ not considered	✓ considered	○ considered in some selected works
3) Process re-design	Process flexibility	design time	✗ not considered	✗ manual approaches	○ not explicitly considered; it may be expected that correctness is considered implicitly
	Automated planning of process models	design time	○ consider upcoming needs for change but only for single actions/fragments	✓ considered	✗ not aiming to provide a complete and correct process model
4) Process implementation		design and execution time	✗ not considered	○ considered by some selected works	✗ not considered

Table.1. Overview of Related Work

5) Process monitoring and controlling	Workflow management	execution time	✗ not considered; aiming at error handling during the execution of processes	✓ considered	✗ not considered; not aiming to provide a complete process model
	Planning	execution time	✗ not considered; aiming to address discrepancies that occurred during process execution	✓ considered	✗ not aiming to provide a complete process model
	Declarative process modeling	design time	✗ not considered	○ considered by few works	✗ not considered; aiming at optimizing performance goals
	Conformance checking / Process enhancement	design time	✗ not considered; aiming to adapt process models that represent already realized needs for change	✓ considered	✓ considered

Table.1. Overview of Related Work (continued)

3 Running Example & Formal Foundation

In our research, we aim for a representation of process models independent of a particular process modeling language. More precisely, in contrast to relying on one single process modeling language such as Event-driven Process Chains (EPC), we use a formal foundation that provides a broader application scope for our approach. Our formal foundation includes so called process graphs, which are also referred to as planning graphs in the research field of automated planning of process models (e.g., Heinrich et al., 2015; Henneberger et al., 2008; Lin et al., 2012; Zheng and Yan, 2008). Process graphs utilize similar concepts as existing well-known process modeling languages such as EPCs, Business Process Model and Notation (BPMN) or Unified Modeling Language (UML) activity diagrams (e.g., van Gorp and Dijkman, 2013).

To illustrate the formal foundation and our approach, we use a simplified excerpt consisting of three actions from a real-world manufacturing process of a European electrical engineering company as a running example (the whole process is part of our evaluation in Section 5). The process is repeatedly influenced by changing requirements and new legal regulations and therefore needs to be adapted frequently. In a first step, the required material needs to be ordered (action "Order material") as there is no material in stock. Thereafter, a circuit board is prefabricated (action "Prefabricate circuit board"). Here, basically, the circuit board goes through the actions of developing, etching and stripping. In order to produce a complete product, the prefabricated circuit board subsequently needs to be assembled with other parts such as microchips and resistors (action "Assemble product"). Finally, the product is ready for sale and the process terminates. Figure 1 shows the process graph of our running example denoted in terms of the formal foundation presented in the following.

The process starts at an initial belief state (short: initial state). *Belief states* are denoted by tables (e.g., in Figure 1, the first belief state at the top, annotated with "Initial state"). They comprise multiple pieces of information, so called *belief state tuples* which are represented by the rows in the according tables. For instance, within our running example, the belief state tuple *(product, not manufactured)* in the upmost table of the process graph in Figure 1 (annotated with "Initial state") expresses that at the beginning of the process, the product is not yet manufactured. *Actions* which lead from one belief state to another are denoted by rounded rectangles (e.g., the action "Order material"). Actions contain *preconditions* (denoted by *pre(a)*) and *effects* (denoted by *eff(a)*). Preconditions (including inputs) denote everything an action needs to be applied, whereas *effects* (including outputs) denote everything an action provides, deallocates or alters after it was applied. The process ends at one to possibly many defined belief states meeting a goal state (i.e., the goal of the process is achieved). For example, in the belief state at the very bottom in Figure 1, a belief state meeting a goal state is reached because the product is manufactured which represents the defined goal state *(product, manufactured)*, denoted in italics.

The essential notions are presented formally in the following Definitions 1 to 5. Hereby, we follow common ways to represent a planning domain (Ghallab et al., 2004, 2016) within automated planning (Bertoli et al., 2001; Bertoli et al., 2006; Heinrich et al., 2015; Heinrich and Schön, 2015, 2016; Henneberger et al., 2008). We thus ensure compatibility with existing works.

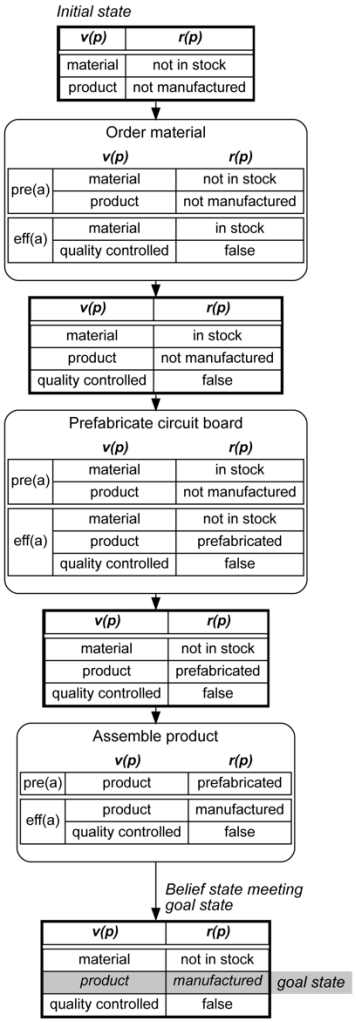

Figure 1. Process Graph of the Simplified Manufacturing Process

Definition 1. (*belief state tuple*). A *belief state tuple p* is a tuple consisting of a *belief state variable v(p)* and a subset *r(p)* of its *domain dom(p)*, which we will write as *p:=(v(p),r(p))*. The domain *dom(p)* specifies which values can generally be assigned to *v(p)* and can for instance represent a data type such as *integer* or a finite set. The set $r(p) \subseteq dom(p)$ is called the *restriction* of *v(p)* and contains the values that can be assigned to *v(p)* in this specific belief state tuple *p*. Let $BST=\{p_1,\ldots,p_n\}$ be a finite set of belief state tuples.

Definition 2. (*action*). An *action a* is a triple consisting of the action name and two sets, which we write as *a:=(name(a),pre(a),eff(a))*. The set $pre(a) \subseteq BST$ are the *preconditions* of *a* and the

set $eff(a) \subseteq BST$ are the *effects* of *a*. An action *a* is *applicable* in a belief state *bs* iff $\forall w \in pre(a)\ \exists u \in bs: v(w)=v(u) \wedge r(w) \cap r(u) \neq \emptyset$. In other words, *a* is applicable in *bs* iff all belief state variables in *pre(a)* also exist in *bs* and the respective restrictions of the belief state variables intersect.

Belief state tuples and actions are used in the definition of a nondeterministic belief state-transition system presented in the following. The graph in Figure 1 is based on such an underlying nondeterministic belief-state transition system. Here, the initial state contains the two belief state tuples *(material, {not in stock})* and *(product, {not manufactured})* with *material* and *product* being the belief state variables and *{not in stock}* and *{not manufactured}* being their restrictions. A nondeterministic belief state-transition system is defined in terms of its belief states, its actions and a transition function that describes how an action leads from one belief state to possibly many belief states (Bertoli et al., 2006; Ghallab et al., 2004, 2016).

Definition 3. (*nondeterministic belief state-transition system*). A *nondeterministic belief state-transition system* is a tuple $\sum=(BS,A,R)$, where

i. $BS \subseteq 2^{BST}$ is a finite set of *belief states*. A *belief state* $bs \in BS$ is a subset of *BST*, containing every belief state variable one time at the most.

ii. *A* is a finite set of actions. The set of actions that are applicable in *bs* are denoted by $app(bs):=\{a \in A \mid a \text{ is applicable in } bs\}$.

iii. $R: BS \times A \rightarrow 2^{BS}$ is the transition function. For each belief state $bs \in BS$ and each action $a \in A$ applicable in *bs* the set of next belief states is calculated as $R(bs,a)=bst_{old} \cup bst_{pre(a)} \cup eff(a)$. Here,
$bst_{old}=bs \backslash \{(v(t),r(t)) \in bs \mid \exists (v(s),r(s)) \in pre(a) \cup eff(a): v(t)=v(s)\}$ are the belief state tuples of *bs* that are determined by the transition function to remain unchanged (the notation "\" represents the set-theoretic difference). Furthermore,
$bst_{pre(a)}=\{(v(t),\ r(t) \cap r(s)) \mid (v(t),r(t)) \in bs \wedge (\exists (v(s),r(s)) \in pre(a): v(t)=v(s)) \wedge (\nexists (v(x),r(x)) \in eff(a): v(t)=v(x))\}$ are the belief state tuples of *bs* whose restriction is further limited by the preconditions of *a*. If *a* is not applicable in *bs*, $R(bs,a)=\emptyset$.

Based on this definition, a graph as presented in Figure 1 and defined in Definition 5 can be constructed from scratch by means of existing planning approaches (cf., e.g., Bertoli et al., 2006; Ghallab et al., 2004, 2016; Heinrich et al., 2009; Heinrich and Schön, 2015). The planning starts with an initial state, constructs the following belief state for each applicable action based on the transition function *R(bs,a)* and continues until a goal state is met (e.g., in Figure 1, the goal state *(product, manufactured)* written in italics is met by the belief state at the very bottom). The input data for the planning can, for instance, be obtained by extracting actions from existing process models, using interfaces of process modeling tools, fresh modeling of actions or conceptualization of (web) services (Bortlik et al., 2018; Heinrich and Mayer, 2018).

Definition 4. (*goal state*). A *goal state* is a subset of *BST*, containing every belief state variable one time at the most, which represents a termination criterion for the process. If a belief state

bs fulfills the termination criterion represented by a goal state *goal* (i.e., $\forall p \in goal: \exists p' \in bs, v(p)=v(p'), r(p') \subset r(p)$), we denote *bs* as *meeting goal*.

Definition 5. (*process graph*). A *process graph* is a bipartite, directed, finite graph $G=(N,E)$ with the set of nodes N and the set of edges E. The set of nodes N consists of two partitions: The set of action nodes A and the set of belief state nodes BS. Each node $bs \in BS$ represents one distinct belief state in the process graph. Each action node $a \in A$ represents an action in the process graph. The process graph starts with one initial state $bs_{init} \in BS$ and ends with one to possibly many belief states $bs_{goal,j} \in BS$ meeting a goal state. A (finite) sequence of states and actions $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ starting with the initial state is called a *path*. A path $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ is called a *feasible path* if the following three additional conditions apply:

i. bs_k meets a goal state

ii. $bs_{init}, \ldots, bs_{k-1}$ do not meet any goal state

iii. $a_1 \in app(bs_{init})$, $bs_2 = R(bs_{init}, a_1)$, ..., $a_{k-1} \in app(bs_{k-1})$, $bs_k = R(bs_{k-1}, a_{k-1})$

Within this paper, we present an approach to adapt process graphs as described in Definition 5. The result of this adaptation is again a process graph based on the definitions presented above. Hence, existing works for the automated construction of control flow patterns (van der Aalst et al., 2003) such as exclusive choice based on process graphs (e.g., Heinrich et al., 2015; Heinrich and Schön, 2016; Meyer and Weske, 2006) can be used as usual to construct process models containing control flow patterns. Thus, it is not necessary to address how to consider control flow patterns in this paper.

4 Design of our Approach

We propose two steps for the adaptation of process graphs. In the first step (i), we identify and classify possible changes to a given process graph. In the second step (ii), potential consequences resulting from these identified changes are addressed.

Ad (i): As given in Definition 5, a process graph consists of belief states (amongst them one initial state and one to possibly many belief states meeting a goal state) and actions. When constructing a process graph using an existing approach for the automated planning of process models, the initial state, the goal states and the actions are used as input. All other belief states of the process graph are constructed during planning and thus, it is not possible without creating inconsistencies that these belief states are *directly* adapted due to needs for change. Therefore, by using our formal foundation, every need for change to a process graph is reflected in a change to this input and consequently, only changes to this input need to be considered for the adaptation of process graphs.

The initial state, goal states and preconditions and effects of actions are all represented by sets of belief state tuples. When changing these sets we will consider changes to single belief state

tuples in the following as changes to multiple belief state tuples can be represented by a sequence of changes to single belief state tuples. In this way, we aim to establish so called *atomic changes*. These are changes that can represent every adaptation to a process graph when put into sequence.

To identify atomic changes, we align our research to the well-known CRUD operations. The CRUD operations have their origin in database systems (cf. Martin, 1983) and are the four elemental, low-level operations "create", "read", "update" and "delete" that cover all possible ways of accessing and altering data. We determine all atomic changes presented in Table 2 by combining these operations with the input discussed above. As "read" does not represent a change in our context, this operation is not taken into account in Table 2.

		CRUD operations		
		Create	**Update**	**Delete**
Input for planning	**Initial state**	---	- Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple	---
	Goal states	Add new goal state	- Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple	Remove existing goal state

	Actions	Add new action	- Update preconditions
			  - Add new belief state tuple
			  - Alter existing belief state tuple
			  - Remove existing belief state tuple
			- Update effects
			  - Add new belief state tuple
			  - Alter existing belief state tuple
			  - Remove existing belief state tuple
				Remove existing action

Table 2. Overview of Atomic Changes

Since exactly one initial state is used as input for planning, each change to the initial state can be represented by an update of it. For goal states and actions as well as for each belief state tuple, however, the operations "create", "update" and "delete" can be applied.

We note that updating a goal state or action could be treated as deleting the old, existing goal state or action and adding a new (the updated) one. However, reusing existing information about the process graph by considering the operation "update" enables a more efficient approach (cf. Section 5). For instance, when updating an action and retrieving the belief states in which the updated action is applicable, it may be beneficial to take into account in which belief states the action was applicable before it was updated.

To address any adaptation of a process graph, a sequence of the presented atomic changes can be used. To give an example, adding multiple single actions sequentially allows to address changes which include a larger number of added actions. A more detailed example is discussed in Section 4.4.

Ad (ii): We identify potential consequences for the process graph (e.g., actions becoming applicable in an updated initial state of the graph) resulting from each of the discussed atomic changes in the following sections. Thereby, we do not merely reduce each atomic change to a planning problem solvable by existing techniques for the automated planning of process models (e.g., Heinrich et al., 2015; Heinrich and Schön, 2015; Henneberger et al., 2008; Hoffmann et al., 2009, 2012; Lin et al., 2012; Zheng and Yan, 2008). Instead, we enhance these techniques to address each individual atomic change compared to "planning from scratch". To do this, we determine where parts of the existing process graph can be reused or where new belief states and actions have to be planned. In addition, we incorporate knowledge about the applicability of actions in the existing process graph to reduce the effort of verifying the applicability of these actions to changed belief states. Please note that a pseudo code of our approach is available in Appendix B.

4.1 Updating the Initial State

Following Definition 5, a process graph starts with exactly one initial state. Thus, every possible change regarding the initial state, seen as an ordered set of atomic changes, consists of the addition of a belief state tuple to the initial state, the removal of a belief state tuple that was present in the initial state, or the update of a belief state tuple's restriction (cf. Table 2). A belief state tuple p with empty restriction in a belief state (i.e., $r(p)= \emptyset$, no value of the belief state variable is feasible) is equivalent to a non-existing belief state tuple. Therefore, the addition of a belief state tuple p can be seen as an update of $(v(p),r(p))$ in which $r(p)= \emptyset$ is changed so that $r(p) \neq \emptyset$ and the removal of a belief state tuple p can be seen as an update of $(v(p),r(p))$ in which $r(p) \neq \emptyset$ is changed to $r(p)= \emptyset$. Thus, we subsequently only need to consider the single case of an updated belief state tuple to fully cover the three possible atomic changes regarding the initial state.

To be able to clearly address the initial state before and after the adaptation, we denote the initial state in the given (i.e., not adapted) process graph with bs_{init} and the initial state after the adaptation with bs_{init}'. As we outline the approach of adapting a process graph to an updated initial state in detail, it is necessary to distinguish between old, (completely) new and updated states in the process graph:

Definition 6. (*old, new, updated states*). Let BS be the set of belief states in the given process graph and BS' be the set of belief states in the adapted process graph. Each belief state $bs \in BS$ is called an *old state*.

We denote $bs' \in BS'$ as the *update* of $bs \in BS$ (or generally as *updated*), if all of the following criteria are fulfilled:

i. $bs' \notin BS$ (i.e., bs' is not old)

ii. there is a sequence of actions $a_1,a_2,\ldots,a_k$ in the given process graph so that a_1 is applicable in the initial state bs_{init} ($a_1 \in app(bs_{init})$, the set of actions applicable in bs_{init}, cf. Definition 2), $bs_1=R(bs_{init},a_1)$, $a_2 \in app(bs_1)$ and so forth until $bs=R(bs_{k-1},a_k)$

iii. this same sequence of actions remains applicable in the adapted process graph (considering the updated initial state) and applying this sequence yields bs'

We call belief states $bs \in BS'$ that are neither old nor updated states *new states*. In other words, if the belief state $bs \in BS'$ is an old state, it is a belief state that was already contained in the given process graph, without any change. If bs is an updated state, it is a belief state that was not contained in the given process graph, but is yielded by a sequence of actions already contained in the given process graph. A new state is a state that was not contained in the given process graph and that is yielded by sequences of actions not contained in the given process graph.

Now we will identify potential consequences resulting from updating the initial state bs_{init}. Updating a belief state tuple p of bs_{init} can impact whether an action a is applicable in bs_{init}' if a belief state tuple p' is contained in the preconditions of a such that $v(p')=v(p)$ (cf. Definition 2). Otherwise, the belief state tuple p is not relevant in order to determine whether a is applicable. Hence, the sets $app(bs_{init})$ and $app(bs_{init}')$ can only differ in actions containing a belief state tuple p' with $v(p)=v(p')$ in their preconditions. Thus, for the set of actions $\{a \in app(bs_{init}) | \nexists p' \in pre(a): v(p')=v(p)\}$, the applicability regarding bs_{init}' does not need to be checked as these actions are unaffected and thus remain applicable. Actions in the set $\{a \in A | \exists p' \in pre(a): v(p')=v(p)\}$, however, need to be checked for potential applicability in bs_{init}'. The actions not contained in $app(bs_{init}')$ are not planned at this point in the adapted process graph.

For each action a that is applicable in both bs_{init} and bs_{init}' (i.e., $a \in app(bs_{init}') \cap app(bs_{init})$) and hence "retained" its applicability we can use $bs=R(bs_{init},a)$ from the given graph, which helps us to determine $bs'=R(bs_{init}',a)$ as we only need to apply the transition function R (cf. Definition 3) with respect to p and transfer these effects to bs. If bs' was contained in the given process graph and thus is an old state, we can retain the whole subgraph starting with bs' as the actions that can be applied in this belief state are known from the given process graph and do not differ since both the belief state and the actions did not change. This is in accordance to existing techniques for the automated planning of process models where the traversal of a previously known state terminates planning. Otherwise (i.e., if bs' is not an old state) bs' is the update of bs (cf. Definition 6). In this case, the updated belief state can now either meet a goal state, which completes the path, or we need to continue by treating bs' as we currently handle bs_{init}'.

The set $app(bs_{init}')$, however, can also contain actions that were not applicable in bs_{init}. For each such action $a \in app(bs_{init}')$ with $a \notin app(bs_{init})$, the transition function R needs to be applied entirely (i.e., not only with respect to the updated belief state tuple p) in order to obtain the belief

state $bs=R(bs_{init}',a)$. If *bs* meets a goal state, the path is completed, else *bs* is either old, updated (from a hitherto feasible path) or new. In the first case, we retain the whole subgraph starting with *bs* from the given process graph. If, however, *bs* is a new state, we have to apply the transition function *R* entirely: We compute *app(bs)* and, for each $a \in app(bs)$, the belief state *R(bs,a)* following *bs*. Again, these belief states have to be checked in regard to being old, updated or new. Updated states are handled in the same way as bs_{init}'. We proceed iteratively in this manner with every upcoming state depending on its classification regarding Definition 6.

Altogether, the approach – in line with existing approaches for the automated planning of process models – starts with the initial state, aborting the traversal of a path as soon as an old state, a belief state which meets a goal state or a belief state *bs* with $app(bs)=\emptyset$ is reached. Especially when traversing updated states it poses an improvement to existing techniques for the automated planning of process models as information from the initial process graph is (re)used.

Within the example (cf. Figure 2; parts influenced by the adaptation are black, not influenced parts are grey), a new external supplier that meets the service level requirements is acquired as a business partner. This external supplier is able to provide prefabricated circuit boards. Hence, the fact that now an appropriate external supplier is available is denoted in terms of the belief state tuple *(external supplier, {available})*, which therefore is added in the initial state (bold). By means of this change, the action "Order prefabricated circuit board" (retrieved from the set of actions *A*, cf. Definition 3), which requires this particular belief state tuple, becomes applicable and thus is planned in the adapted initial state. After this action, a new belief state is created in which the action "Assemble product" is applicable, which in turn leads to the goal state. Thus, as result of the adaptation, a new feasible path (denoted by means of bold arrows and bold-bordered actions and belief states) is constructed.

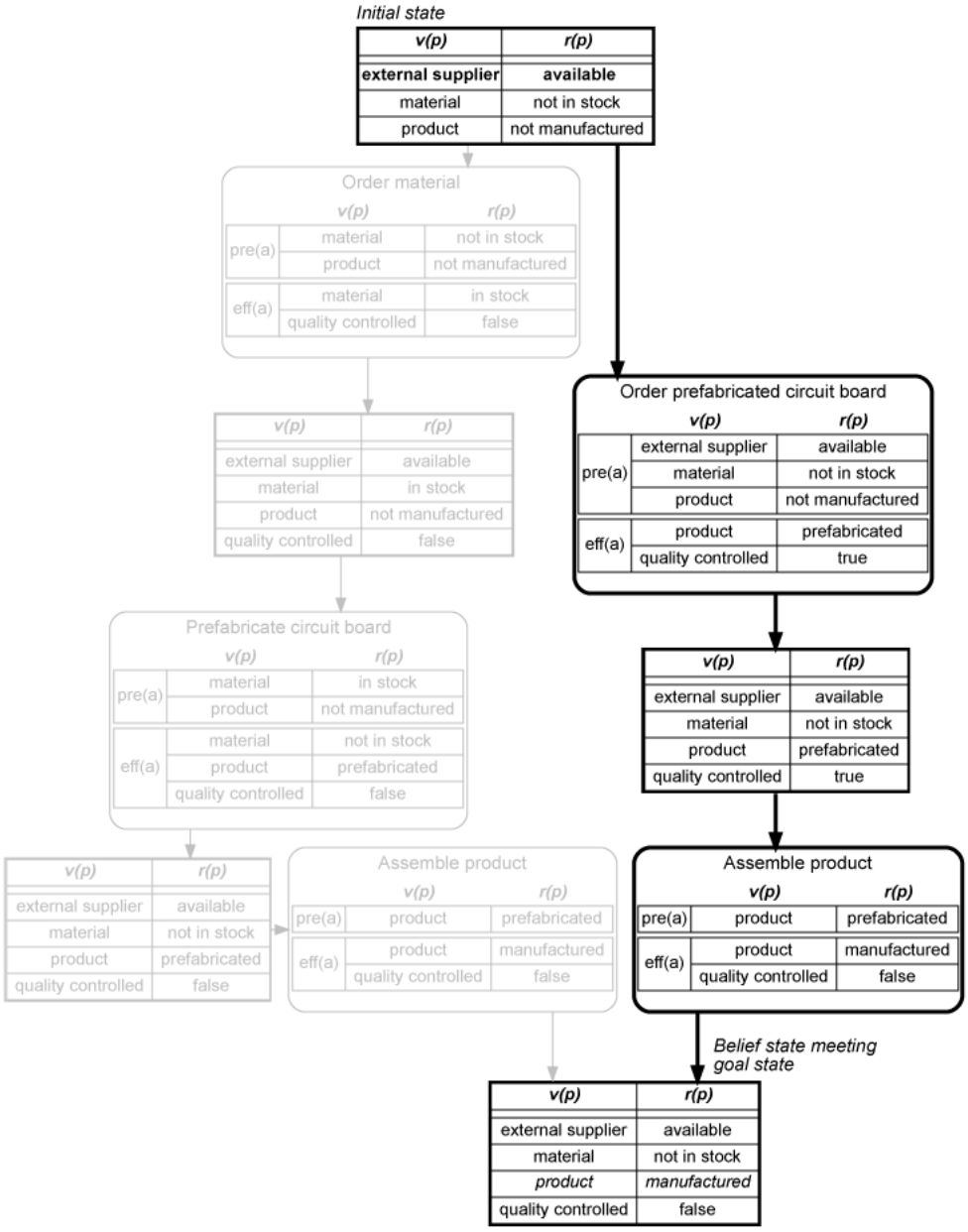

Figure 2. Process Graph after the Adaptation resulting from updating the Initial State

4.2 Changing (the Set of) Goal States

A process graph contains one to possibly many goal states (cf. Definition 5). In alignment with the CRUD functions, we consider the atomic changes “adding a goal state”, “removing a goal state” and “updating a goal state” (cf. Table 2).

4.2.1 Adding a Goal State

Denoting the set of all goal states in the given process graph with *GOALS*, the addition of a new goal state $goal \notin GOALS$ with $GOALS' = GOALS \cup \{goal\}$ could, on the one hand, result in new

feasible paths, which lead to this new goal state. Such new feasible paths have not been feasible in the given process graph and thus need to be newly constructed. On the other hand, as goal states serve as termination criteria, this new goal state could imply feasible paths in the given process graph being "shortened" so that for a given path $bs_{init},a_1,bs_1,a_2,\ldots,bs_k$ there exists $j<k$ with bs_j meeting *goal*.

To determine these consequences, we traverse the paths of the given process graph and their belief states (except for the belief states meeting a goal state from *GOALS* at the end of each such feasible path), starting with the initial state. For each belief state *bs*, we need to check whether *bs* meets the new goal state (first case) or whether actions applicable in *bs* lead to the new goal state subsequently (second case). If, in the first case, the currently considered belief state *bs* meets the new goal state, we abort the traversal of this path as it ends here. In the second case, if *bs* does not meet the new goal state, we have to take into account every possible new belief state that can follow right after *bs* and start planning from each of these new belief states in order to (possibly) retrieve new feasible paths that lead to *goal*. With this in mind, we first determine all actions $a \in app(bs)$ (retrieved from the set of actions *A*, cf. Definition 3) which were not planned in *bs* in the given process graph. For each of these actions we then determine the belief state $bs'=R(bs,a)$ and continue planning from *bs'*. If, during this planning, no belief state that meets *goal* is retrieved or no further action is applicable, the planning of the current path is aborted.

4.2.2 Removing a Goal State

Removing a goal state $goal \in GOALS$ (i.e., $GOALS'=GOALS\backslash\{goal\}$) implies that a termination criterion for the process is deleted. Therefore, each path in the given process graph that ends at a belief state meeting *goal* needs to be checked whether it can be extended by an existing planning technique so that it leads to one of the remaining goal states. If no goal state can be reached from its last belief state (which formerly had met the now removed goal state *goal*), it is not considered in the adapted process graph. No other paths are affected by this change.

We therefore take into account each belief state *bs* of the given process graph that meets *goal* and try to reach one of the remaining goal states by planning (i.e., applying the transition function *R* entirely and computing all applicable actions and the belief states resulting from them), starting with each such *bs*. Thus, we first check each belief state *bs* that meets *goal* for the criteria of the remaining goal states. If *bs* meets the criteria of a remaining goal state, it is ensured that the paths which had ended at *goal* remain feasible in the adapted process graph. Else, the next planning step is executed: We determine the applicable actions in *bs* and construct the according resulting belief states by means of the transition function. Note that as soon as there are no actions applicable in the examined belief state and thus the planning step fails, the paths which had ended at goal cannot be extended to a feasible path and are therefore not considered in the adapted process graph.

4.2.3 Updating a Goal State

We separate the case of updating a goal state *goal* into two subcases. Since goal states serve as termination criteria, we distinguish between a strengthening update (i.e., making the conditions for meeting *goal* more severe) and a weakening update (i.e., making the conditions less severe). The updated goal state will be denoted by *goal'* (and thus $GOALS'=(GOALS\setminus\{goal\})\cup\{goal'\}$). If a goal state is updated in any manner that is not included in the following two cases, we can represent this adaptation as a weakening update followed by a strengthening update.

Strengthening update. Strengthening the conditions of a goal state *goal* includes the addition of a belief state tuple to *goal* as well as changes to a belief state tuple $p\in goal$ limiting its restriction, formally replacing *p* by *p'* with $v(p)=v(p'), r(p)\neq\emptyset\neq r(p'), r(p)\neq r(p')$ and $r(p')\subset r(p)$ so that $goal'=(goal\setminus\{p\})\cup\{p'\}$. When strengthening the conditions of *goal*, the set of (world) states that meet *goal'* is a proper subset of the set of states that meet *goal*, as these criteria are more severe. Thus, we proceed in a similar way to the case of removing a goal state (cf. Section 4.2.2): We start planning for each belief state *bs* meeting *goal* and each action that can be applied in *bs*, trying to reach one of the goal states from *GOALS'*.

Looking at the running example, a new compliance directive has come into force, requiring the company to integrate quality management as a documented and controlled task in the manufacturing process. Due to the new directive, it is required that the quality assurance is documented as an inherent part of the process. Therefore, the belief state tuple *(quality controlled, {true})* is added to the goal state (bold and in italics). Thus, as seen in Figure 3, an action "External quality assurance" is now planned in the belief state meeting the old goal state in order to meet the new, adapted goal state including the new belief state tuple.

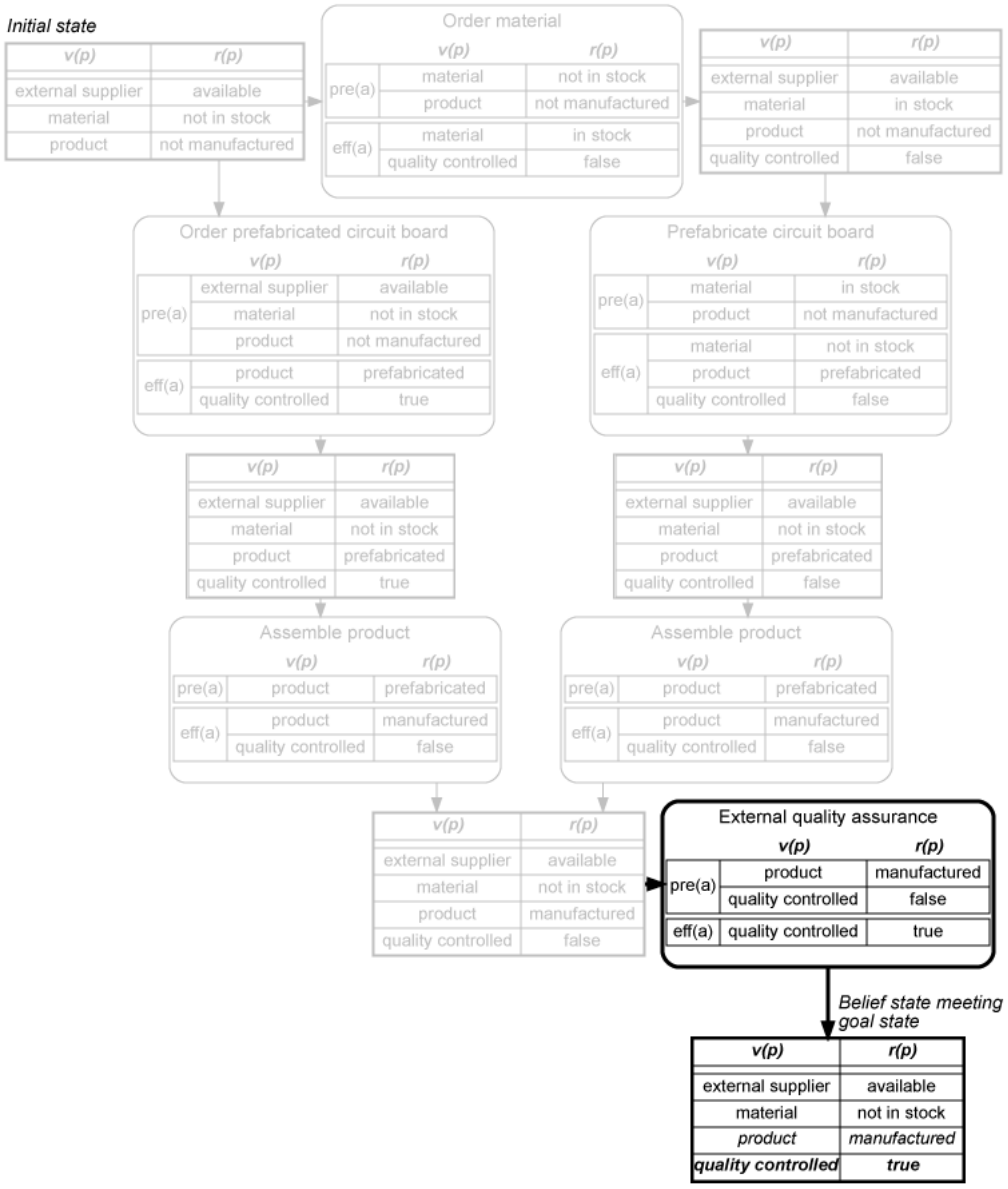

Figure 3. Process Graph after the Adaptation due to a strengthening Update of the Goal State

Weakening update. This case covers removing a belief state tuple from *goal* as well as changes that extend the restriction of a belief state tuple $p \in goal$ (i.e., replacing p by p' with $v(p)=v(p'), r(p) \neq \emptyset \neq r(p'), r(p) \neq r(p')$ and $r(p) \subset r(p')$). Belief states meeting the goal state *goal* canonically meet *goal'*. Additionally, there are possibly further belief states meeting *goal'* which do not meet *goal*. Therefore, we align the approach to the case of adding the goal state *goal'* (cf. Section 4.2.1): We traverse all belief states in the process graph, check whether a belief state meets *goal'*, and try to retrieve new feasible paths to *goal'* by checking whether actions applicable in the belief states lead to *goal'* subsequently. In this way, feasible paths in the existing process graph may be shortened and new feasible paths may be constructed.

4.3 Changing (the Set of) Actions

As described in Definition 2, actions are triples consisting of the action name, the preconditions of the action and the effects of the action. According to CRUD, the requirement of an adaptation can arise from the addition of an action to the set of actions *A*, the removal of an action from *A* or the update of the preconditions or effects of an action in *A* (cf. Table 2).

4.3.1 Adding an Action

Let *a* be a new action so that $A'=A \cup \{a\}$. As *a* might be applicable in the given process graph, we need to check whether there exists a belief state *bs* in the given process graph such that *a* is applicable in *bs*. In such belief states we start planning by applying the transition function *R(bs,a)*. Further, there may exist paths $(bs_{init}, a_1, \ldots, bs_k)$ with $a_1, \ldots, a_{k-1} \in A$ that have not been feasible paths in the given process graph and with *a* being applicable in bs_k. In such belief states we also start planning by applying the transition function $R(bs_k, a)$. Thereby, we possibly retrieve new feasible paths leading to a goal state.

Within the running example, the company decides to establish an own, internal quality assurance. This assurance, in difference to the external quality assurance contractor, is able to check the assembled product as well as (optionally) the internally prefabricated circuit board. As we see in Figure 4, an action “Internal quality assurance” (bold) is added to the process graph appropriately throughout the whole process.

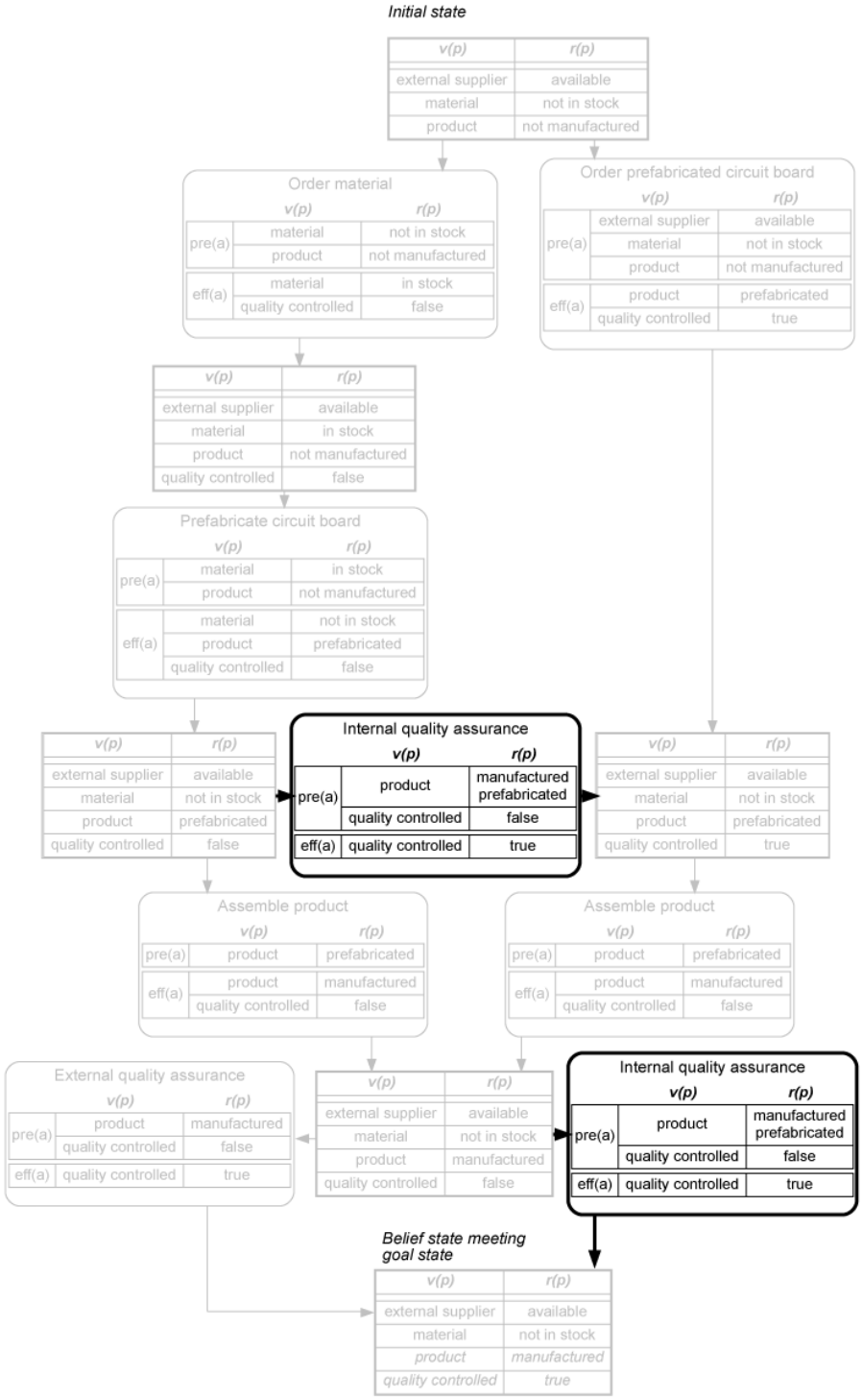

Figure 4. Process Graph after the Adaptation due to an added Action

4.3.2 Removing an Action

When removing an action *a* from *A* so that *A'=A\{a}*, there can be no new feasible paths leading to a goal state. Further, each path in the given process graph containing *a* is not feasible in the

adapted process graph and hence not retained. The paths not containing *a* are not affected at all and are retained.

4.3.3 Updating an Action

When updating an action *a* to *a'* we need to consider the case of updating the preconditions as well as the case of updating the effects of *a*. Updating the preconditions can be separated into the two subcases of strengthening and weakening updates since any update that is not covered by one of these two cases can be treated as performing an weakening update, followed by a strengthening update.

Strengthening update of the preconditions. When strengthening the preconditions of an action *a* (i.e., adding a belief state tuple *p* to *pre(a)* or updating *p* to *p'* so that $v(p)=v(p'), r(p)\neq\emptyset\neq r(p'), r(p)\neq r(p'), r(p')\subset r(p)$ and $pre(a')=(pre(a)\setminus\{p\})\cup\{p'\}$), only a subset of the belief states of the given process graph in which *a* was applicable also fulfills the requirements for the applicability of *a'*. Hence, we need to check for each belief state *bs* in which *a* was applicable whether *a'* is still applicable. If this is not the case, we do not consider the paths containing *a'* in the adapted process graph (cf. case of removing an action, Section 4.3.2). On the other hand, if *a'* is still applicable in *bs*, the belief state bs_1 that results from *R(bs,a)* may differ from the belief state bs_1' resulting from *R(bs,a')* (cf. Definition 2). In this case, it is possible that the sets $app(bs_1)$ and $app(bs_1')$ do not coincide. We then proceed analogously as we did when treating the case of updating the initial state (cf. Section 4.1) with bs_1' taking the role of the updated state to bs_1.

Weakening update of the preconditions. When weakening the preconditions of an action *a* (i.e., removing a belief state tuple from *pre(a)* or updating *p* to *p'* so that $v(p)=v(p'), r(p)\neq\emptyset\neq r(p'), r(p)\neq r(p'), r(p)\subset r(p')$ and $pre(a')=(pre(a)\setminus\{p\})\cup\{p'\}$) it is possible that *a'* becomes applicable in additional belief states in which *a* has not been applicable. We therefore check each belief state *bs* of the given process graph with $a\notin app(bs)$ in regard to $a'\in app(bs)$. If, indeed, $a'\in app(bs)$ holds, we apply a planning approach in accordance to the case of adding a new action (cf. Section 4.3.1). Further, there may exist paths $(bs_{init}, a_1, \ldots, bs_k)$ with $a_1, \ldots, a_{k-1}\in A$ that have not been feasible paths in the given process graph and with $a\notin app(bs_k)$, but $a'\in app(bs_k)$. In such belief states we also start planning by applying the transition function $R(bs_k,a')$. Thereby, we may retrieve new feasible paths leading to a goal state. Additionally, the same situation as in the preceding paragraph ($R(bs,a)\neq R(bs,a')$) can arise and is handled in the same manner as above (cf. Section 4.1).

Updating the effects. Finally, when updating the effects of an action *a* with respect to a single belief state tuple, we consider each belief state *bs* of the given process graph in which *a* is applicable. Due to the changed effects, once again, we may encounter the situation in which $R(bs,a)\neq R(bs,a')$ holds, which is handled as above (cf. Section 4.1).

4.4 Summary of the Approach

In the Sections 4.1-4.3 it was shown how to adapt a process graph to each of the atomic changes specified in Table 2. Table 3 summarizes the main enhancements with regard to existing methods from automated planning which do not reuse any results from previous planning runs.

		Main enhancements		
		Reuse of applicability information	**Reuse of existing subgraphs**	**Planning based on existing process graph**
Type of atomic change (cf. Table 2)	**Update initial state**	✓	✓	✕
	Add goal state	✕	✕	✓
	Update goal state	✕	✕	✓
	Remove goal state	✕	✕	✓
	Add action	✕	✕	✓
	Update action	✓	✓	✓
	Remove action	✕	✕	✓

Table 3. Enhancements over existing Planning Approaches

As all adaptations can be realized as a sequence of these atomic changes, this means that a full-featured approach for the adaptation of process models has been developed. We discuss this by means of our running example:

In order to enter new markets, a new manufacturing facility is built by the electrical engineering company. In this new facility the manufacturing process from above (cf. Fig. 4) is planned to be applied, however it needs to be adapted. To reach a broad market coverage, a second production line for the prefabrication of circuit boards consisting of two machines has to be added. Additionally, analyses show that a new packaging is needed for this market and hence, product packing is planned to be included into the manufacturing process. As the external quality assurance contractor does not operate in this market, it is planned to exclusively handle quality assurance at the facility. Furthermore, local regulatory requirements demand the quality assurance for prefabricated circuit boards to be mandatory.

From this description the corresponding atomic changes can be inferred directly. First, a second production line is incorporated into our process graph by adding the actions “Prefabricate circuit board on machine 1” and “Prefabricate circuit board on machine 2”. These actions have preconditions and effects similar to the action “Prefabricate circuit board” with the only difference being the belief state tuple *(product, {in prefabrication})* which is needed as these actions have

to be put in sequence. Second, product packing is enabled by adding the action "Packing product" and updating the goal state to contain the belief state tuple *(product, {packed})*. With these atomic changes, the ability as well as the necessity for a manufactured product to be packed is given. Third, to meet the business changes regarding quality assurance, the action "External quality assurance" is deleted. Additionally, the belief state tuple *(quality controlled, {true})* is added to the preconditions of the action "Assemble product" to comply with legal requirements. In this way prefabricated circuit boards cannot be processed without having their quality checked. The resulting process graph is shown in Figure 5.

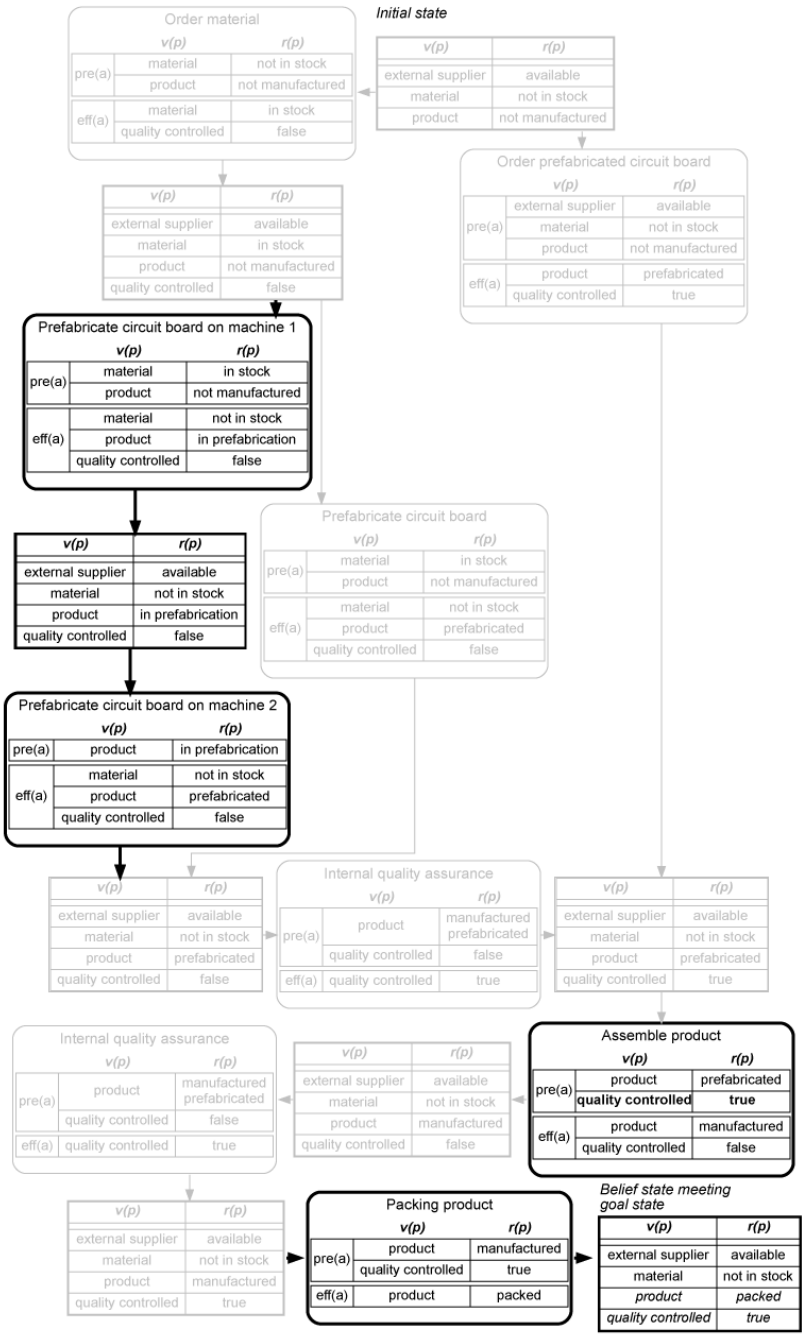

Figure 5. Process Graph after the Adaptation due to multiple Atomic Changes

5 Evaluation

We assessed our approach based on evaluation criteria stated in literature (Prat et al., 2015). In particular, the following Table 4 shows the four analyzed evaluation questions, the respective evaluation criterion, the way for analysis and the section or appendix in which a detailed description is given (please note that parts of the evaluation were moved to the appendix due to their length).

More precisely, the evaluation criterion regarding correctness and completeness (E1) was proved (cf. Appendix A for a detailed discussion), which shows that our approach can indeed adapt process models such that the resulting process models are correct and complete. This finding supports contribution (C2). To evaluate (E3), the approach was used in a real-world scenario for the adaptation of existing process models to needs for change in advance in an automated manner (cf. C1). Further, it was shown by means of a simulation experiment as well as an algorithmic complexity analysis (E4) that the presented approach provides advantages in performance and computational complexity compared to planning the process graphs from scratch.

	Evaluation question	**Evaluation criterion**	**Way for analysis**	**Section / Appendix**
(E1)	Does the proposed approach terminate and construct correct and complete adapted process models?	Correctness and Completeness	Mathematical proofs	A
(E2)	Can the approach be instantiated?	Technical feasibility	Prototypical software implementation, Pseudocode	5.1, B
(E3)	Can the approach be successfully applied in a real-world scenario?	Operational feasibility	Application to manufacturing process of an engineering company	5.2
(E4)	Does the approach provide performance advantages compared to planning from scratch?	Computational complexity	Simulation experiment based on 12 real-world processes, Algorithmic complexity analysis	5.3, C

Table 4. Overview of Evaluation

5.1 Evaluation of (E2) Technical Feasibility

We implemented our approach for the automated adaptation of process models in a software prototype. An existing Java implementation of a planning technique (cf. Bertoli et al., 2006;

Heinrich and Schön, 2015) for nondeterministic state transition systems able to construct process graphs (cf. Definition 5) served as a basis. The existing implementation allows to specify planning problems by means of XML files. We added the functionality to also specify changes to a process graph via XML files. In particular, using a set of XML files (one file specifying the initial regular planning problem and another file specifying changes), a process graph and one to possibly many subsequent changes can be specified. We further integrated the previously presented approach (cf. Sections 4.1 to 4.3) in the implemented planning technique. Persons other than the programmers validated the source code via structured walkthroughs. Moreover, the validity of this extended prototype was ensured by carrying out structured tests using the JUnit framework. The pseudocode of our implementation can be found in Section B in the appendix. This supports the technical feasibility (E2) of our approach.

5.2 Evaluation of (E3) Operational Feasibility

In order to evaluate whether the approach can adapt process models to needs for change in advance (contribution C1) and thus operational feasibility, we conducted a field experiment by applying our approach to a manufacturing process of a European electrical engineering company. To do so, we interviewed the manager of the manufacturing department about a process that was subject to several adaptations in recent history. Based on a first interview, the annotations of actions, initial state and goal states of the original process (that was in place before these adaptations) could be prepared. In a second meeting, we reviewed them together with the staff to validate that their specification was accurate. Thereafter, a detailed process graph, depicting the existing process (consisting of 27 actions, 20 belief states and 48 paths; cf. Table 5) could be planned by means of the aforementioned Java implementation. The running example used within this paper is a simplified excerpt of this graph. During further meetings, the manager provided us with information about the aforementioned needs for change to this process that took place in recent history. Please note that despite the changes to the processes had occurred in the past, our approach still adapted to a need for change in advance in this setting because only the need for change was used as input and the actually conducted processes and resulting changes to it just served as reference for comparison to our adaptation result. To assess the operational feasibility of our approach, we analyzed the following three questions necessary for a successful application of our approach in this real-world scenario:

(E3.1) Using our approach, is it possible to adapt the process graph to the needs for change stated by the manager?

(E3.2) Do the adapted process graphs represent the actually conducted processes?

(E3.3) Are the adapted process graphs assessed as correct and complete by the staff?

Ad (E3.1): Based upon the information given by the manager, we were able to determine atomic changes and specify them in terms of the aforementioned XML files. Subsequently, we adapted the process graph by means of our prototype in the order the changes occurred in reality (cf.

Table 5). The first adaptation resulted from the demand to consider the situation of prefabricated packaging being in stock (an update of the initial state). This adaptation resulted in 15 old and 5 updated belief states in the adapted process graph. Thereafter, based on the adapted process graph we addressed the second need for change and so on. Overall, with respect to Table 1, the changes "updating the initial state", "adding an action", "removing an action" and "updating a goal state" were addressed. All needs for change could be represented and addressed.

Ad (E3.2): In order to compare the adapted process graphs with the respective actual processes conducted after each change in the company, we scheduled a further meeting with the staff of the engineering company. After the first adaptation (cf. third row in Table 5), we presented the resulting process graph to the staff and discussed the differences with them. However, we observed that the staff had some problems comprehending this graph. Therefore, for the subsequent adaptations (cf. rows four to eight in Table 5), we visually simplified the adapted process graphs so that they became more understandable for the staff. In detail, we removed the belief states from the versions presented to the company and omitted the preconditions and effects of the actions represented in the graphs so that their layout was similar to UML activity diagrams. Still, the complete process model was presented. Then, we discussed these graphs with the manager and employees of the manufacturing department. Particularly, for each need for change (cf. Table 5), we elaborated the differences between the graph before adaptation and the adapted graph in detail. Thereby, we asked the staff whether the adapted graphs represent the processes as they had actually been conducted in the company as soon as the according change took place. The staff confirmed this for every case.

Ad (E3.3): The staff further assessed the paths in the adapted process graphs to be correct. We also asked whether the adapted graphs neglected any feasible paths. Here, the staff validated that the graphs contained all paths actually used in the company and that no feasible paths not represented in the adapted process graphs were known. Please note that while the number of paths to check was high in some cases, most paths just contained the same actions in different order, making a manual verification possible.

Description of the needs for change	Type of atomic change (cf. Table 2)	Number of …					
		feasible paths	actions	old belief states	new belief states	updated belief states	belief states (in total)
		… in the process graph after the adaptation					
Original process graph (starting point)	-	48	27	-	-	-	20
Considering the situation that prefabricated packaging is in stock	Updating the initial state	56	28	15	0	5	20
Considering the situation that there is an external supplier for circuit boards	Updating the initial state	76	45	7	12	13	32
Preproduction of spare parts can now be done by the company itself	Adding an action	80	55	32	0	9	41
A quality assurance department is set up internally	Adding an action	460	76	41	0	9	50
A production machine for circuit boards is sold	Removing an action	268	70	47	0	0	47
Testing the functionality of the product at the installation site is required	Updating a goal state	2,412	136	47	42	0	89

Table 5. Adaptations performed in the Case of the Engineering Company

To conclude, the (E3) operational feasibility of our approach could be supported in this real-world scenario. Our approach was able to adapt process models to needs for change in advance in an automated manner (cf. C1). However, we observed that the resulting process graphs are not yet easy to comprehend. This issue may be solved by using approaches to construct control

flow patterns based on our adapted process graph (Heinrich et al., 2015; Heinrich and Schön, 2016; Meyer and Weske, 2006). For assessing (E3), however, it was sufficient to describe the process graphs to the staff and discuss the changes of these graphs in detail.

5.3 Evaluation of (E4) Performance

We additionally conducted an analysis to evaluate computational complexity as well as a simulation experiment to analyze the difference in performance of our approach compared to planning from scratch[1].

Overall, the complexity analysis (cf. Section C in the appendix) shows that our approach provides considerable advantages in computational complexity compared to planning process graphs from scratch.

Regarding the simulation experiment, we focus on atomic changes to provide transparent results. However, please note that all possible adaptations can be realized as a sequence of these atomic changes. For our analysis, we measured absolute runtimes as well as each ratio, which means, the absolute runtime for adaptation divided by the corresponding absolute runtime for planning from scratch for all possible types of atomic change (cf. Table 2). To do so, we used adaptation cases based upon 12 existing real-world process graphs of different companies from the application contexts *Project Management*, *Insurance Management*, *Loan Management* and *Private Banking* (cf. Table 6). The process graphs consist of 17 to 8,267 actions and 15 to 2,693 belief states and contain numeric domains as well as discrete domains in their belief state tuples. There are two process graphs that stand out regarding the number of feasible paths (*Selling an insurance contract* and *Contracting wealth management customer*). In these two cases, large parts could be conducted in parallel to other large parts in the same process. Process graphs do not contain control flow patterns and paths represent sequences of subsequent belief states and actions. Hence, these two process graphs represent large processes of complete value chains (including back office) in which many actions can be executed in multiple different orders. This results in process graphs comprising a vast number of feasible paths, each of which represents a different possible order of actions. Yet, the number of distinct state variables remains rather small because these central business state variables can have many different values and auxiliary state variables, which were not counted here. We deliberately analyzed these two process graphs to show the feasibility of our approach regarding large process graphs.

[1] We executed the prototype on a Dell Latitude Notebook with an Intel Core i7-2640M, 2.80 GHz, 8GB RAM running on Windows 8.1 (Version 6.3.9600) 64 bit and Java 1.8.0 (build 1.8.0.-b132) 64 bit.

Application context	Process description	Number of actions in the given process graph	Number of belief states in the given process graph	Number of feasible paths in the given process graph	Number of distinct state variables
Project Management	Preparing and coordinating project profile	17	15	6	4
Project Management	Specifying project resources	25	18	36	7
Project Management	Allocating project resources	26	22	24	6
Project Management	Preparing the project report for the board	38	25	252	5
Insurance Management	Administrating customer and product database	43	38	52	7
Insurance Management	Handling insured events	54	44	60	6
Insurance Management	Selling an insurance contract	8,267	2,693	97,501,324,491	9
Loan Management	Analyzing credit rating	40	31	197	6
Loan Management	Selling mortgage loans	57	43	216	8
Loan Management	Settling mortgage loans	122	69	6.572	11
Private Banking	Contracting wealth management customer	278	189	1,244,416	27
Private Banking	Order management	82	74	32	19

Table 6. Key Properties of used Real-world Processes

In a first step, we defined random adaptation cases for each type of atomic change based on the belief state variables of each of the 12 process graphs. Each of these generated adaptation cases was then (automatically) validated with regard to its validity. For instance, a goal state could only be removed if there was at least one goal state remaining after the adaptation. Invalid adaptation cases remained unconsidered. We randomly generated 1,500 valid adaptation cases for each type of atomic change over all 12 process graphs. For each case, the specification of the process graph, the specification of the adaptation and the specification for planning the adapted process graph from scratch were automatically prepared in terms of XML files that could be imported in our prototypical implementation.

We applied our approach to these adaptation cases and automatically verified that adapting the given process graphs and planning the adapted process graphs from scratch resulted in exactly the same process model in each case. Then, we compared the runtime required for adapting the given process graphs with the runtime required for planning the graphs from scratch. Both runtimes do not take into account the time required to generate the XML files representing the adaptation cases. The results of this runtime comparison can be seen in Figure 6.

We observed that the required runtime for adapting the existing process graphs by the prototype is lower than for planning the graphs from scratch for each type of atomic change. The left part of Figure 6 shows the mean of the required absolute runtime for adapting the process graphs (first line in each cell) as well as the mean percentage ratio (absolute runtime for adaptation divided by absolute runtime for planning from scratch; second line in each cell), which varies between 3.68% and 10.52%, depending on the type of atomic change. To give an example of a process, independent of the type of atomic change our approach takes on average 0.35 seconds for adapting the graph of the process *Selling an insurance contract*. In contrast, planning from scratch would on average take about 10 seconds, which leads to an average time saving of 96.4%. The right part of Figure 6 shows box plots of these percentage ratios. The left and right ends of the boxes are the first and third quartiles, and the bands inside the boxes are the medians. The whiskers (i.e., the horizontal lines outside of the boxes) include all values within 1.5*interquartile range.

These results of the simulation experiment show that while for just a few adaptation cases, adapting instead of planning from scratch provided only negligible or even non-existent runtime advantages, for most adaptation cases, the runtime advantage was considerable. Further, the results support that using our approach provides considerable performance advantages, even if a sequence of changes has to be addressed. To analyze the significance of our findings, we further conducted a one-tailed paired t-test. This kind of test was chosen because we aimed to analyze a paired sample dataset (runtimes for adapting versus planning from scratch). There was a significant difference in the runtimes according to the results of the t-test: *degrees of freedom*=139,190; *t-value*=88.685, *p-value*=2.2e-16. The *p*-value was consistently less than or equal to 2.2e-16 across all types of atomic changes. These significant results support the thesis

that our approach is faster than planning the adapted process graphs from scratch. This is especially advantageous when working with graphical process modeling tools. In such tools, modelers can – *in real time* – conduct a sequence of changes in order to, step by step, adapt a given process model to needs for change. The previously mentioned example of adapting *Selling an insurance contract* underlines this argument. Here, for instance, using the presented approach would on average result in a 0.35 second waiting time after each entered change instead of 10 seconds when planning from scratch. Thus, our approach enables modelers to work much more comfortably compared to using existing approaches.

	Add	Update	Remove
Initial state	---	7.73ms 9.05%	---
Goal states	3.52ms 3.80%	6.07ms 7.32%	0.58ms 3.68%
Actions	11.67ms 10.52%	18.68ms 7.80%	2.87ms 5.71%

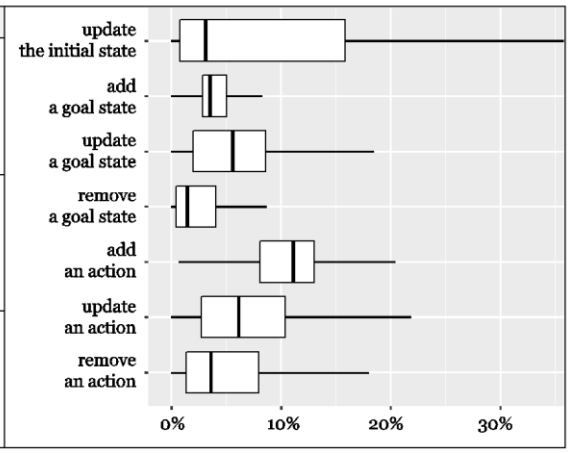

Figure 6. Evaluation Results by means of a Prototypical Implementation

6 Conclusion, Limitations and Future Work

In this paper, we presented a novel approach for the automated adaptation of process models that – in contrast to existing works – constructs correct and complete process models (cf. contribution (C2)). This approach can be used to adapt existing process models to needs for change in advance in an automated manner (cf. contribution (C1)). We mathematically verified our approach, showed its technical feasibility by means of a prototypical software implementation and its operational feasibility by applying the approach to a real-world process in a field experiment. Additionally, we conducted a complexity analysis which shows that our approach provides considerable advantages in regard to computational complexity compared to planning the process graph from scratch. Moreover, we analyzed the performance of our approach in a simulation experiment.

Our research possesses some limitations, which should be addressed in future work. First, our approach adapts process graphs without cycles (cf. Definition 5). However, the (sub) paths within a cycle could be analyzed once and separately using the approach presented in Section 4. In this way, process graphs containing arbitrary cycles could be adapted as well. Second, we evaluated the operational feasibility of our approach by applying it to a single real-world scenario in an experimental setting. Thus, the presented approach should be further evaluated in a

broader context. In particular, a larger number of field experiments in different industry sectors should be conducted to verify the operational feasibility.

Further, the runtime of adapting a process model to a (very) large number of changes may be slower than planning a process model from scratch. Although we have already provided some insights on this topic by means of the simulation experiment presented above, additional work needs to be done. For instance, an estimation regarding the expected runtime of adapting a given process model compared to the expected runtime of planning it from scratch, based on the needs for change, would be useful. This could provide fruitful insights for deciding whether to use our approach or to plan the process model from scratch, given a large number of needs for change. Further, we aim to construct complete adapted graphs (cf. contribution (C2)) and thus do not focus on a heuristic approach in this paper. However, the runtime for adapting process graphs may additionally be reduced further by means of a heuristic approach in case a process model with all feasible paths is not necessary.

Moreover, research work has to be done in order to transfer our approach to related fields. While a larger number of approaches in (web) service composition utilizes concepts of automated planning (including notions of "states" and "actions" with "preconditions" and "effects", cf. Fan et al., 2018; Montarnal et al., 2018; Rao and Su, 2005) recent research has shown how to apply similar concepts to the field of process mining as well (e.g., cf. Mannhardt et al., 2018; Song et al., 2016). To extensively evaluate the feasibility of our approach in these fields it needs to be implemented in the according toolsets and applied to different real-world scenarios. We hope that our work will open doors for further research in this exciting area.

7 References

Afflerbach, P., G. Kastner, F. Krause and M. Röglinger (2014). "The Business Value of Process Flexibility" *Business & Information Systems Engineering (BISE)* 6 (4), 203–214.

Alférez, G. H., V. Pelechano, R. Mazo, C. Salinesi and D. Diaz (2014). "Dynamic adaptation of service compositions with variability models" *Journal of Systems and Software* 91, 24–47.

Augusto, A., R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella and A. Soo (2018). "Automated discovery of process models from event logs: Review and benchmark" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 31 (4).

Barba, I., C. Del Valle, B. Weber and A. Jiménez-Ramírez (2013a). "Automatic generation of optimized business process models from constraint-based specifications" *International Journal of Cooperative Information Systems* 22 (02), 1350009.

Barba, I., B. Weber, C. Del Valle and A. Jiménez-Ramírez (2013b). "User recommendations for the optimized execution of business processes" *Data & Knowledge Engineering (DKE)* 86, 61–84.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2001). "Planning in nondeterministic domains under partial observability via symbolic model checking" *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)* 1, 473–478.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2006). "Strong planning under partial observability" *Artificial Intelligence* 170 (4–5), 337–384.

Bider, I. (2005). "Masking flexibility behind rigidity: Notes on how much flexibility people are willing to cope with". In: *Proceedings of the International Conference on Advanced Information Systems (CAiSE 2005)*, pp. 7–18.

Bortlik, M., B. Heinrich and M. Mayer (2018). "Multi User Context-Aware Service Selection for Mobile Environments" *Business & Information Systems Engineering (BISE)* 60 (5), 415–430.

Bose, R.P. J. C., W. M. P. van der Aalst, I. Zliobaite and M. Pechenizkiy (2014). "Dealing with concept drifts in process mining" *IEEE Transactions on Neural Networks and Learning Systems* 25 (1), 154–171.

Bucchiarone, A., M. Pistore, H. Raik and R. Kazhamiakin (2011). "Adaptation of service-based business processes by context-aware replanning". In: *IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2011)*, pp. 1–8.

Canfora, G., M. Di Penta, R. Esposito and M. L. Villani (2005). "An Approach for QoS-aware Service Composition based on Genetic Algorithms". In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO 2005)*, pp. 1069–1075.

Chafle, G., K. Dasgupta, A. Kumar, S. Mittal and B. Srivastava (2017). "Adaptation in Web Service Composition and Execution". In: *Proceedings of the IEEE International Conference on Services Computing (SCC 2017)*, pp. 549–557.

Chen, L., X. Li and Q. Yang (2012). "Continuous process improvement based on adaptive workflow mining technique" *Journal of Computational Information Systems (JCIS)* 8 (7), 2891–2898.

Cognini, R., F. Corradini, S. Gnesi, A. Polini and B. Re (2018). "Business process flexibility - a systematic literature review with a software systems perspective" *Information Systems Frontiers* 20 (2).

Döhring, M., H. A. Reijers and S. Smirnov (2014). "Configuration vs. adaptation for business process variant maintenance: An empirical study" *Information Systems* 39, 108–133.

Eisenbarth, T. (2013). "Semantic Process Models: Transformation, Adaptation, Resource Consideration". Dissertation. University of Augsburg.

Eisenbarth, T., F. Lautenbacher and B. Bauer (2011). "Adaptation of Process Models – A Semantic-based Approach" *Journal of Research and Practice in Information Technology* 43 (1), 5–23.

Ellis, C., K. Keddara and G. Rozenberg (1995). "Dynamic change within workflow systems". In: *Proceedings of the Conference on Organizational Computing Systems (COCS 1995)*, pp. 10–21.

Fahland, D., C. Favre, J. Koehler, N. Lohmann, H. Völzer and K. Wolf (2011). "Analysis on demand. Instantaneous soundness checking of industrial business process models" *Data & Knowledge Engineering (DKE)* 70 (5), 448–466.

Fahland, D. and W. M. P. van der Aalst (2012). "Repairing Process Models to Reflect Reality". In *Business Process Management*, pp. 229–245: Springer.

Fan, S.-L., Y.-B. Yang and X.-X. Wang (2018). "Efficient Web Service Composition via Knapsack-Variant Algorithm". In: *Proceedings of the IEEE International Conference on Services Computing (SCC 2018)*, pp. 51–66.

Forstner, E., N. Kamprath and M. Röglinger (2014). "Capability development with process maturity models – Decision framework and economic analysis" *Journal of Decision Systems (JDS)* 23 (2), 127–150.

Gambini, M., M. La Rosa, S. Migliorini and A. H. M. ter Hofstede (2011). "Automated error correction of business process models". In *Business Process Management*, pp. 148–165: Springer.

Garcia-Bañuelos, L., N. R. van Beest, M. Dumas, M. La Rosa and W. Mertens (2017). "Complete and interpretable conformance checking of business processes" *IEEE Transactions on Software Engineering* 44 (3), 262–290.

Garrido, A., C. Guzman and E. Onaindia (2010). "Anytime plan-adaptation for continuous planning". In: *Proceedings of the 28th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG 2010)*, pp. 62–69.

Gerevini, A. E., A. Saetti and I. Serina (2012). "Case-based Planning for Problems with Real-valued Fluents: Kernel Functions for Effective Plan Retrieval" *Frontiers in Artificial Intelligence and Applications* 242, 348–353.

Gerevini, A. E. and I. Serina (2010). "Fast Plan Adaptation through Planning Graphs: Local and Systematic Search Techniques". In: *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems*, pp. 112–121.

Ghallab, M., D. S. Nau and P. Traverso (2004). *Automated Planning: Theory & Practice:* Morgan Kaufmann.

Ghallab, M., D. S. Nau and P. Traverso (2016). *Automated Planning and Acting:* Cambridge University Press.

Hallerbach, A., T. Bauer and M. Reichert (2010). "Capturing variability in business process models: the Provop approach" *Journal of Software Maintenance and Evolution: Research and Practice* 22 (6-7), 519–546.

Hammer, M. (2015). "What is Business Process Management?". In *Handbook on Business Process Management 1*, pp. 3–16: Springer Berlin Heidelberg.

Heinrich, B., M. Bolsinger and M.-A. Bewernik (2009). "Automated planning of process models: the construction of exclusive choices". In: *Proceedings of the 30th International Conference on Information Systems (ICIS 2009)*.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models: Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

Heinrich, B. and M. Mayer (2018). "Service selection in mobile environments: considering multiple users and context-awareness" *Journal of Decision Systems (JDS)* 27 (2), 92–122.

Heinrich, B., A. Schiller and D. Schön (2018). "The cooperation of multiple actors within process models: an automated planning approach" *Journal of Decision Systems (JDS)* 27 (4), 238–274.

Heinrich, B. and D. Schön (2015). "Automated Planning of Context-aware Process Models". In: *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*.

Heinrich, B. and D. Schön (2016). "Automated Planning of Process Models: The Construction of Simple Merges". In: *Proceedings of the 24rd European Conference on Information Systems (ECIS 2016)*.

Henneberger, M., B. Heinrich, F. Lautenbacher and B. Bauer (2008). "Semantic-Based Planning of Process Models". In *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2008)*, pp. 1677–1689.

Hoffmann, J., I. Weber and F. M. Kraft (2009). "Planning@SAP: An application in business process management". In: *Proceedings of the 2nd International Scheduling and Planning Applications woRKshop (SPARK 2009)*.

Hoffmann, J., I. Weber and F. M. Kraft (2012). "SAP Speaks PDDL: Exploiting a Software-Engineering Model for Planning in Business Process Management" *Journal of Artificial Intelligence Research* 44 (1), 587–632.

Hornung, T., A. Koschmider and A. Oberweis (2007). "A Rule-based Autocompletion Of Business Process Models". In: *Proceedings of the 19th Conference on Advanced Information Systems Engineering (CAiSE 2007)*.

Hull, R. and H. R. Motahari Nezhad (2016). "Rethinking BPM in a Cognitive World. Transforming How We Learn and Perform Business Processes". In: *Business Process Management*: Springer International Publishing, pp. 3–19.

IEEE Task Force on Process Mining (2012). "Process Mining Manifesto". In *Business Process Management Workshops*, pp. 169–194: Springer Berlin Heidelberg.

Jiménez-Ramírez, A., I. Barba, C. Del Valle and B. Weber (2013). "Generating Multi-objective Optimized Business Process Enactment Plans". In *Advanced Information Systems Engineering*, pp. 99–115: Springer Berlin Heidelberg.

Kalenkova, A. A., W. M. van der Aalst, I. A. Lomazova and V. A. Rubin (2017). "Process mining using BPMN: relating event logs and process models" *Software and Systems Modeling* 16 (4), 1019–1048.

Kambhampati, S. (1997). "Refinement Planning as a Unifying Framework for Plan Synthesis" *AI Magazine* 18 (2), 67–98.

Katzmarzik, A., M. Henneberger and H. U. Buhl (2012). "Interdependencies between automation and sourcing of business processes" *Journal of Decision Systems (JDS)* 21 (4), 331–352.

Khan, F. H., S. Bashir, M. Y. Javed, A. Khan and M. S. H. Khiyal (2010). "QoS Based Dynamic Web Services Composition & Execution" *International Journal of Computer Science and Information Security (IJCSIS)* 7 (2), 147–152.

La Rosa, M., W. M. van der Aalst, M. Dumas and F. P. Milani (2017). "Business process variability modeling: A survey" *ACM Computing Surveys (CSUR)* 50 (1), 2.

Lautenbacher, F., T. Eisenbarth and B. Bauer (2009). "Process model adaptation using semantic technologies". In: *13th Enterprise Distributed Object Computing Conference Workshops (EDOCW 2009)*, pp. 301–309.

Le Clair, C. (2013). *Make Business Agility A Key Corporate Attribute – It Could Be What Saves You*. URL: http://blogs.forrester.com/craig_le_clair/13-09-09-make_business_agility_a_key_corporate_attribute_it_could_be_what_saves_you (visited on 07/03/2019).

Leemans, S. J. J., D. Fahland and W. M. van der Aalst (2018). "Scalable process discovery and conformance checking" *Software and Systems Modeling* 17 (2), 599–631.

Leoni, M. d. and A. Marrella (2017). "Aligning Real Process Executions and Prescriptive Process Models through Automated Planning" *Expert Systems with Applications* 82, 162–183.

Lin, S.-Y., G.-T. Lin, K.-M. Chao and C.-C. Lo (2012). "A Cost-Effective Planning Graph Approach for Large-Scale Web Service Composition" *Mathematical Problems in Engineering* 2012 (1), 1–21.

Linden, I., M. Derbali, G. Schwanen, J.-M. Jacquet, R. Ramdoyal and C. Ponsard (2014). "Supporting Business Process Exception Management by Dynamically Building Processes Using the BEM Framework". In *Decision Support Systems III - Impact of Decision Support Systems for Global Environments*, pp. 67–78: Springer International Publishing.

Mannhardt, F., M. de Leoni, H. A. Reijers, W. M. van der Aalst and P. J. Toussaint (2018). "Guided Process Discovery-A pattern-based approach" *Information Systems* 76, 1–18.

Marrella, A. (2018). "Automated Planning for Business Process Management" *Journal on Data Semantics*, 1–20.

Marrella, A. and M. Mecella (2011). "Continuous Planning for Solving Business Process Adaptivity". In *Enterprise, Business-Process and Information Systems Modeling*, pp. 118–132: Springer Berlin Heidelberg.

Marrella, A., M. Mecella and A. Russo (2011). "Featuring Automatic Adaptivity through Workflow Enactment and Planning". In: *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing*.

Marrella, A., M. Mecella and S. Sardina (2017). "Intelligent Process Adaptation in the SmartPM System" *ACM Transactions on Intelligent Systems and Technology (TIST)* 8 (2), 1–43.

Marrella, A., A. Russo and M. Mecella (2012). "Planlets: Automatically Recovering Dynamic Processes in YAWL". In *On the Move to Meaningful Internet Systems: OTM 2012*, pp. 268–286: Springer Berlin Heidelberg.

Martin, J. (1983). *Managing the data-base environment:* Prentice-Hall.

Masellis, R. de, C. Di Francescomarino, C. Ghidini, M. Montali and S. Tessaris (2017). "Add data into business process verification: Bridging the gap between theory and practice". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

McElheran, K. (2015). "Do Market Leaders Lead in Business Process Innovation? The Case(s) of E-business Adoption" *Management Science* 61 (6), 1197–1216.

Mei-hong, S., J. Shou-shan, G. Yong-gang, C. Liang and C. Kai-duan (2012). "A Method of Adaptive Process Mining Based on Time-Varying Sliding Window and Relation of Adjacent Event Dependency". In: *2nd International Conference on Intelligent System Design and Engineering Application (ISDEA 2012)*, pp. 24–31.

Mejri, A., S. Ayachi-Ghannouchi and R. Martinho (2018). "A quantitative approach for measuring the degree of flexibility of business process models" *Business Process Management Journal (BPMJ)* 24 (4), 1023–1049.

Mendling, J., H. M. W. Verbeek, B. F. van Dongen, W. M. P. van der Aalst and G. Neumann (2008). "Detection and prediction of errors in EPCs of the SAP reference model" *Data & Knowledge Engineering (DKE)* 64 (1), 312–329.

Meyer, H. and M. Weske (2006). "Automated service composition using heuristic search". In *Business Process Management*, pp. 81–96: Springer Berlin Heidelberg.

Montarnal, A., W. Mu, F. Benaben, J. Lamothe, M. Lauras and N. Salatge (2018). "Automated deduction of cross-organizational collaborative business processes" *Information Sciences* 453, 30–49.

Nebel, B. and J. Koehler (1995). "Plan reuse versus plan generation. A theoretical and empirical analysis" *Artificial Intelligence* 76 (1-2), 427–454.

Nunes, V. T., F. M. Santoro, C. M. L. Werner and C. G. Ralha (2018). "Real-Time Process Adaptation. A Context-Aware Replanning Approach" *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48 (1), 99–118.

Pesic, M., M. H. Schonenberg, N. Sidorova and van der Aalst,W. M. P. (2007). "Constraint-Based Workflow Models: Change Made Easy". In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pp. 77–94: Springer Berlin Heidelberg.

Pesic, M. and W. M. van der Aalst (2006). "A declarative approach for flexible business processes management". In: *Business Process Management Workshops*, pp. 169–180.

Prat, N., I. Comyn-Wattiau and J. Akoka (2015). "A Taxonomy of Evaluation Methods for Information Systems Artifacts" *Journal of Management Information Systems (JMIS)* 32 (3), 229–267.

Rao, J. and X. Su (2005). "A Survey of Automated Web Service Composition Methods". In *Semantic Web Services and Web Process Composition*, pp. 43–54: Springer Berlin Heidelberg.

Regev, G., I. Bider and A. Wegmann (2007). "Defining business process flexibility with the help of invariants" *Software Process: Improvement and Practice* 12 (1), 65–79.

Reichert, M. and P. Dadam (1997). "A framework for dynamic changes in workflow management systems". In: *Proceedings of the 8th International Workshop on Database and Expert Systems Applications*, pp. 42–48.

Reichert, M. and P. Dadam (1998). "Adeptflex--Supporting Dynamic Changes of Workflows Without Losing Control" *Journal of Intelligent Information Systems* 10 (2), 93–129.

Reichert, M. U. and B. Weber (2012). *Enabling flexibility in process-aware information systems: challenges, methods, technologies:* Springer.

Reisert, C., S. Zelt and J. Wacker (2018). "How to move from paper to impact in business process management: The journey of SAP". In *Business Process Management Cases*, pp. 21–36: Springer.

Rinderle, S., M. Reichert and P. Dadam (2004). "Correctness criteria for dynamic changes in workflow systems – a survey" *Data & Knowledge Engineering (DKE)* 50 (1), 9–34.

Rosemann, M., J. C. Recker and C. Flender (2010). "Designing context-aware business processes". In *Systems Analysis and Design: People, Processes, and Projects*, pp. 53–74: ME Sharpe, Inc.

Rosemann, M. and J. vom Brocke (2015). "The Six Core Elements of Business Process Management". In *Handbook on Business Process Management 1. Introduction, Methods, and Information Systems*. 2nd Edition, pp. 107–122: Springer.

Roy, S., A. S. M. Sajeev, S. Bihary and A. Ranjan (2014). "An Empirical Study of Error Patterns in Industrial Business Process Models" *IEEE Transactions on Services Computing* 7 (2), 140–153.

Scala, E., R. Micalizio and P. Torasso (2015). "Robust plan execution via reconfiguration and replanning" *AI Communications* 28 (3), 479–509.

Song, W., H.-A. Jacobsen, C. Ye and X. Ma (2016). "Process discovery from dependence-complete event logs" *IEEE Transactions on Services Computing* 9 (5), 714–727.

Tax, N., I. Verenich, M. La Rosa and M. Dumas (2017). "Predictive business process monitoring with LSTM neural networks". In: *International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, pp. 477–492.

van Beest, N.R.T.P., E. Kaldeli, P. Bulanov, J. C. Wortmann and A. Lazovik (2014). "Automated runtime repair of business processes" *Information Systems* 39, 45–79.

van der Aalst, W. M. P. (2013). "Business process management: A comprehensive survey" *ISRN Software Engineering* (507984).

van der Aalst, W. M. P. (2015). "Extracting Event Data from Databases to Unleash Process Mining". In *BPM - Driving Innovation in a Digital World*, pp. 105–128: Springer International Publishing.

van der Aalst, W. M. P. and S. Jablonski (2000). "Dealing with workflow change: identification of issues and solutions" *International Journal of Computer Systems Science & Engineering (CSSE)* 15 (5), 267–276.

van der Aalst, W. M. P., M. Pesic and H. Schonenberg (2009). "Declarative workflows: Balancing between flexibility and support" *Computer Science - Research and Development* 23 (2), 99–113.

van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns" *Distributed and Parallel Databases* 14 (1), 5–51.

van der Aalst, W. M. P. and H. M.W. Verbeek (2014). "Process discovery and conformance checking using passages" *Fundamenta Informaticae* 131 (1), 103–138.

van der Krogt, R., A. Bos and C. Witteveen (2002). "Replanning in a Resource-Based Framework". In *Multi-Agent Systems and Applications II*, pp. 148–158: Springer Berlin Heidelberg.

van der Krogt, R. and M. de Weerdt (2005). "Plan Repair as an Extension of Planning". In: *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS 2005)*, pp. 161–170.

van Dongen, B. F., A. K. Alves de Medeiros and L. Wen (2009). "Process Mining: Overview and Outlook of Petri Net Discovery Algorithms". In *Transactions on Petri Nets and Other Models of Concurrency II*, pp. 225–242: Springer Berlin Heidelberg.

van Gorp, P. and R. Dijkman (2013). "A visual token-based formalization of BPMN 2.0 based on in-place transformations" *Information and Software Technology* 55 (2), 365–394.

Verbeek, H. M. W. and W. M. P. van der Aalst (2005). "Analyzing BPEL processes using Petri nets". In: *Proceedings of the 2nd International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management*, pp. 59–78.

vom Brocke, J. (2009). "Design Principles for Reference Modelling". In *Innovations in Information Systems Modeling*, pp. 269–296: IGI Global.

vom Brocke, J. and J. Mendling (eds.) (2018). *Business Process Management Cases. Digital Innovation and Business Transformation in Practice:* Springer International Publishing.

Weber, B., M. Reichert and S. Rinderle-Ma (2008). "Change patterns and change support features-enhancing flexibility in process-aware information systems" *Data & Knowledge Engineering (DKE)* 66 (3), 438–466.

Weber, I. (2007). "Requirements for Implementing Business Process Models through Composition of Semantic Web Services". In *Enterprise Interoperability II*, pp. 3–14: Springer London.

Wynn, M. T., H. M. W. Verbeek, W. M. P. van der Aalst, A. H. M. ter Hofstede and D. Edmond (2009). "Business process verification-finally a reality!" *Business Process Management Journal (BPMJ)* 15 (1), 74–92.

Zheng, X. and Y. Yan (2008). "An Efficient Syntactic Web Service Composition Algorithm Based on the Planning Graph Model". In: *IEEE International Conference on Web Services (ICWS 2008)*, pp. 691–699.

Appendix

A. Evaluation of (E1) Correctness and Completeness

In the following, we focus on proving that the presented approach for the adaptation of process graphs fulfills the properties correctness, completeness and termination. In our approach, we state to "conduct planning steps". Here, we make use of existing techniques for the automated planning of process models (e.g., Bertoli et al., 2006; Heinrich et al., 2015) and refer to their works for providing proofs for the fact that planning by conducting planning steps fulfills the properties correctness, completeness and termination.

THEOREM 1. *The process graphs constructed by the approach are correct: Only feasible paths are contained in an adapted process graph.*

Proof. We distinguish the possible changes as described in the approach.

Updating the initial state. We show the feasibility of each path by using Definition 5 and, in particular, conditions i. to iii. To satisfy condition iii. of Definition 5, we examine all state transitions within the paths of the adapted process graph. In accordance with our approach, these transitions can be divided into transitions from old, updated and new states.

For each new and updated state the applicability of each following action in the adapted process model is verified and the following states are constructed by (partially) applying the transition function where needed, which leads to feasibility condition iii. being fulfilled by construction. Additionally, old states stem from the given process graph. Since only the initial state was updated, no further changes occur once an old state has been reached and thus the according subgraphs, which remain correct, are used for the adapted process graph. Hence, the feasibility condition iii. is fulfilled in these cases as well.

A path is completed as soon as we are certain to reach a goal state through an old, updated or new state. When no goal state is reached and no further actions can be applied, the current path is not considered in the adapted process graph. Thus, conditions i. and ii. of Definition 5 are fulfilled and the feasibility of every path in the adapted process graph is proven.

Adding a goal state. At first, we consider all paths feasible in the given process graph. For such a path $(bs_{init}, a_1, bs_2, \ldots, bs_k)$, two cases may occur: In the first case, $bs_{init}, bs_{2,\ldots,} bs_{k-1}$ do not meet the new goal state. Then, the path remains feasible as the conditions for a feasible path in Definition 5 are still fulfilled. In the second case, (at least) one of the belief states $bs_{init}, bs_{2,\ldots,} bs_{k-1}$ meets the new goal state *goal*. In this case, the existing path is shortened until only one such belief state is contained and is the last belief state in the path. This shortened path is feasible as well: i. and ii. in Definition 5 are fulfilled by construction and iii. remains fulfilled. In addition to these kinds of paths, the adapted process graph may also contain new paths leading to *goal*. As we construct these paths by conducting planning steps (i.e., iteratively computing *app(bs)* and applying the transition function for every belief state *bs*), the paths are always feasible.

Removing a goal state. We again consider all paths feasible in the given process graph. Let $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ be such a path. If bs_k meets an element of $GOALS' = GOALS \setminus \{goal\}$, the path obviously remains feasible as the conditions in Definition 5 are still fulfilled. On the other hand, if the path had ended at *goal* it is extended until a belief state meeting a goal state from *GOALS'* is reached. This is done by conducting planning steps until the extended path is feasible. If such an extension to a feasible path is not possible, the path, which formerly was feasible, is not considered in the adapted process graph. No further new paths are constructed. Hence, overall, only feasible paths are contained in the adapted process graph.

Updating a goal state. Strengthening update. Again, we consider all paths feasible in the given process graph. Let $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ be such a path. If bs_k meets a goal state from $GOALS \setminus \{goal\}$, the path obviously remains feasible as the conditions in Definition 5 are still fulfilled. If the path had ended at *goal*, it is extended until a belief state meeting a goal state from *GOALS'* is reached. This is done by conducting planning steps until the extended path is feasible. If such an extension to a feasible path is not possible, the path, which formerly was feasible, is not considered in the adapted process graph. No further new paths are constructed. Thus, similar to the case of removing a goal, only feasible paths are contained in the adapted process graph.

Updating a goal state. Weakening update. Yet again, we first consider all paths feasible in the given process graph. For such a path $(bs_{init}, a_1, bs_2, \ldots, bs_k)$, two cases may occur: In the first case, $bs_{init}, bs_{2,\ldots,} bs_{k-1}$ do not meet *goal'*. Then, the path remains feasible as the conditions for a feasible path in Definition 5 are still fulfilled. In the second case, (at least) one of the belief states $bs_{init}, bs_{2,\ldots,} bs_{k-1}$ meets *goal'*. In this case, the existing path is shortened until only one such belief state is contained and is the last belief state in the path. This shortened path is feasible as well: i. and ii. in Definition 5 are fulfilled by construction and iii. remains fulfilled. In addition to these kinds of paths, the adapted process graph may also contain new paths leading to *goal'*. As we construct these paths by conducting planning steps, the paths are always feasible.

Adding an action. At first we note that all paths of the given process graph remain feasible when adding *a* to the set of actions since the conditions i. to iii. of Definition 5 are still fulfilled. In addition to that we try to reach goal states through new paths that include the action *a*. As we do this by conducting planning steps, the constructed paths are always ensured to be feasible.

Removing an action. When removing an action *a* from the set of actions, the adapted process graph contains the feasible paths of the given process graph which do not contain *a*. As every action in such a path still fulfills the applicability criterion and the last belief state is the only belief state that meets a goal state in such a path, these paths remain feasible (cf. Definition 5).

Updating an action. Strengthening update of the preconditions. The adapted process graph consists of paths retained from the given process graph and newly constructed paths. We start by considering all paths feasible in the given process graph. If such a path $(bs_{init}, a_1, bs_2, \ldots, bs_k)$

does not contain a at all, it obviously remains feasible as the path does not change at all and the conditions i.-iii. in Definition 5 are still fulfilled. Otherwise, there exists $i<k$ with $a = a_i$ and the applicability of a' in bs_i is checked. If $a' \in app(bs_i)$ and $R(bs_i,a) \neq R(bs_i,a')$, one tries to construct a new path and proceeds as when treating the case of the initial state, for which correctness was already proven. Hence, all newly constructed paths are feasible. On the other hand, if $a' \in app(bs_i)$ and $R(bs_i,a) = R(bs_i,a')$ for all i, the conditions i.-iii. in Definition 5 remain fulfilled as all belief states in the path remain identical. Finally, if $a' \notin app(bs_i)$, the path is not considered in the adapted process graph. Hence, only feasible paths are retained from the given process graph.

Updating an action. Weakening update of the preconditions. In case of a weakening update of the preconditions of an action a resulting in the action a', all feasible paths of the given process graph which do not contain a are retained. Since these paths remain unchanged, they stay feasible. Furthermore, additional paths in the adapted process graph are retrieved by conducting planning steps from belief states bs with $a \notin app(bs)$; they are feasible by construction. Additionally, for every belief state bs of the given process graph with $a \in app(bs)$ and $a' \in app(bs)$, we follow the approach for an update of the initial state. This approach, as seen above, leads to feasible paths.

Updating the effects. When updating the effects of an action a, one retains all paths $(bs_{init},a_1,bs_2,\ldots,bs_k)$ which do not contain a at all and obviously remain feasible as they do not change at all. Otherwise, for $a = a_i$ ($i<k$), if $R(bs_i,a) \neq R(bs_i,a')$, one tries to construct a new path and proceeds as when treating the case of the initial state, for which correctness was already proven.

Thus, Theorem 1 is shown for each case and hence proven. *q.e.d.*

THEOREM 2. *The process graphs constructed by the approach are complete: All feasible paths are contained in an adapted process graph.*

Proof. We distinguish the possible changes as described in the approach.

Updating the initial state. According to Definition 5, each feasible path has to start with the initial state and each following action has to be applicable in its preceding state and has to lead to a goal state (in the sense that the conduction of this action and potentially subsequent actions results in a goal state). As in the previous theorem, we therefore examine all state transitions from old, updated and new states in order to show that the adapted process graph contains every path of this nature. Let bs be a belief state.

In case of bs being an old state, we retain the subgraph from the given process model which starts with bs. As the given process graph is complete, this subgraph contains all actions from $app(bs)$ that lead to a goal state.

Let bs be an updated or new state. If bs itself does not meet a goal state, each action $a \in app(bs)$ is checked in regard to whether it leads to a goal state: For each such action a all possible

subsequent actions and states are retrieved by applying the transition function until a goal state is reached, an old state is reached or no further action can be applied. In the first two cases, *a* indeed leads to (at least) one goal state and consequently the sequence *bs,a* is part of a feasible path. In the last case, on the contrary, every path containing the sequence *bs,a* is not considered in the adapted process graph as *a* does not lead to a goal state.

To sum up, beginning with bs_{init} for each old, updated or new state we either examine all applicable actions and thereafter retain those leading to a goal state or we retain a subgraph of the given process graph which contains all applicable actions that lead to a goal state. Thus, the adapted process graph contains all feasible paths.

Adding a goal state. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the addition of *goal* to the set of goal belief states *GOALS* such that *GOALS'=GOALS∪{goal}*. We need to show that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is indeed contained in the adapted process graph. As $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is feasible, according to Definition 5 i., bs_k meets one of the goal states from *GOALS'=GOALS∪{goal}*. In the first case, let bs_k meet one of the goal states from *GOALS*. In this case, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was feasible in the given process graph and – since bs_{init},bs_2,bs_{k-1} do not meet a goal state from *GOALS'* because of Definition 5 ii. – is retained from the given process graph and thus contained in the adapted process graph. In the second case, bs_k meets *goal*. It is then possible that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was part of a feasible path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ with $m>k$ in the given process graph. Such a path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ is, according to the approach, shortened to $(bs_{init},a_1,bs_2,\ldots,bs_k)$ and then retained. Hence, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is contained in the adapted process graph. If $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was not part of a feasible path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ in the given process graph, (at least) one of the actions $a_1,a_2,\ldots$ was not planned in the given process graph in bs_{init}, resp. $bs_1,\ldots$. For such actions, planning steps are conducted, which lead to the inclusion of $(bs_{init},a_1,bs_2,\ldots,bs_k)$ into the adapted process graph. Thus, overall, it is guaranteed that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is contained in the adapted process graph.

Removing a goal state. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the removal of *goal* from the set of goal belief states *GOALS* such that *GOALS'=GOALS\{goal}*. If $bs_{init},bs_2,\ldots,bs_{k-1}$ do not meet *goal*, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was a feasible path in the given process graph (cf. conditions i.-iii. in Definition 5), which is retained and hence contained in the adapted process graph. Otherwise, let bs_m ($m<k$) be the first belief state that meets *goal*. Then, $(bs_{init},a_1,bs_2,\ldots,bs_m)$ was a feasible path in the given process graph which, by according to our approach, is extended to $(bs_{init},a_1,bs_2,\ldots,bs_k)$ – and possibly other additional feasible paths – in the adapted process graph.

Updating a goal state. Strengthening update. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the update of a goal state *goal* to *goal'*. If $bs_{init},bs_2,\ldots,bs_{k-1}$ do not meet *goal*, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was a feasible path in the given process graph (cf. conditions i.-iii. in Definition 5), which is retained and hence contained in the adapted process graph. Otherwise, let bs_m be the first such belief state. Then, $(bs_{init},a_1,bs_2,\ldots,bs_m)$ was a feasible path in the given

process graph which, according to our approach, is extended to $(bs_{init},a_1,bs_2,\ldots,bs_k)$ – and possibly other additional feasible paths – in the adapted process graph.

Updating a goal state. Weakening update. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the update of a goal state *goal* to *goal'*. We need to show that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is indeed contained in the adapted process graph. As $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is feasible, according to Definition 5 i., bs_k meets one of the goal states from $GOALS'=(GOALS\backslash\{goal\})\cup\{goal'\}$. In the first case, let bs_k meet one of the goal states from $GOALS\backslash\{goal\}$. In this case, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was feasible in the given process graph and – since bs_{init},bs_2,bs_{k-1} do not meet a goal state from *GOALS'* because of Definition 5 ii. – is retained from the given process graph and thus contained in the adapted process graph. In the second case, bs_k meets *goal'*. It is then possible that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was part of a feasible path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ with $m\geq k$ in the given process graph. Such a path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ is shortened to $(bs_{init},a_1,bs_2,\ldots,bs_k)$ and then retained. Hence, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is contained in the adapted process graph. If $(bs_{init},a_1,bs_2,\ldots,bs_k)$ was not part of a feasible path $(bs_{init},a_1,bs_2,\ldots,bs_k,\ldots,bs_m)$ in the given process graph, (at least) one of the actions $a_1,a_2,\ldots$ was not planned in the given process graph in bs_{init}, resp. $bs_2,\ldots$. For such actions, planning steps are conducted, which lead to the inclusion of $(bs_{init},a_1,bs_2,\ldots,bs_k)$ into the adapted process graph. Thus, overall, it is guaranteed that $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is contained in the adapted process graph.

Adding an action. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the addition of a to the set of actions. If a is not among the actions $a_1,a_2,\ldots$ of the aforementioned path, this feasible path is contained in the given process graph. As our approach retains all paths from the given process graph, this path is contained in the adapted process graph as well. Now let a be contained in the actions $a_1,a_2,\ldots$ of the selected feasible path. Then there is a belief state bs preceding (the first occurrence of) a in this path with $a\in app(bs)$. As elaborated in the design of our approach, we examine the path $(bs_{init},a_1,bs_2,\ldots,bs)$ as it ends with a state in which a is applicable. From here, we conduct planning steps to determine and retain all feasible paths that extend $(bs_{init},a_1,bs_2,\ldots,bs)$ and hence $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is contained in the adapted process graph.

Removing an action. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after the removal of a from the set of actions A. As, in regard to Definition 5, actions must be part of the set of actions to be contained in a path, this path does not contain a. It is contained in the given process graph and, as such a path is not changed at all, retained in the adapted process graph according to our approach.

Updating an action. Strengthening update of the preconditions. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after a strengthening update of the preconditions of an action a resulting in the action *a'*. If this path does not contain *a'*, it remains unchanged, is retained from the given process graph and hence contained in the adapted process graph. If, however, the path contains *a'* at some place a_i (let i be the smallest index such that $a'=a_i$), a part of the considered path,

more precisely $(bs_{init},a_1,bs_2,\ldots,bs_i)$, is contained in the given process graph. From here, we proceed as in the case of updating the initial state and retrieve all feasible paths from bs_i. As this was proven to be complete, the feasible path $(bs_{init},a_1,bs_2,\ldots,bs_k)$ is retrieved as well.

Updating an action. Weakening update of the preconditions. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after a weakening update of the preconditions of an action a resulting in the action a'. If this path does not contain a', it remains unchanged, is retained from the given process graph and hence contained in the adapted process graph. If, however, the path contains a' at some place a_i (let i be the smallest index such that $a'=a_i$), it is either retrieved by conducting planning steps from a belief state bs_j with $j \leq i$ of the given process graph or by following the approach for the initial state (which, as seen above, is complete).

Updating the effects. Let $(bs_{init},a_1,bs_2,\ldots,bs_k)$ be any feasible path after updating the effects of an action a resulting in the action a'. Again, if this path does not contain the updated action, the path is contained in the given process graph and, according to the approach, retained. Otherwise, there exists an index $i<k$ with $a'=a_i$ and $a' \neq a_j$ for all $j<i$. Here, we treat $R(bs_i,a')$ as the update of $R(bs_i,a)$ and follow the approach for an update of the initial state which retrieves all feasible paths (seen in the proof above) starting with $(bs_{init},a_1,bs_2,\ldots, bs_i,a',R(bs_i,a'))$, $(bs_{init},a_1,bs_2,\ldots,bs_k)$ being amongst them.

Thus, Theorem 2 is shown for each case and hence proven. *q.e.d.*

THEOREM 3. *The approach terminates.*

Proof. To show that the algorithm terminates, we distinguish the possible changes as described in the approach.

Updating the initial state. At first, we will show that the traversal of each belief state terminates. The traversal of an old state immediately terminates as we retain the corresponding subgraph. Traversing an updated state bs, we first note that the computation of the set $app(bs)$ comes to an end and that the set $app(bs)$ is finite. This stems from the fact that checking the applicability of an action a (i.e., $\forall w \in pre(a)\ \exists u \in bs : v(w)=v(u) \wedge r(w) \cap r(u) \neq \emptyset$) is a comparison of the restrictions of a finite number of belief state tuples and hence terminates and the fact that we are provided with a finite set of actions. As a next step, the finite set of all following belief states is retrieved by executing the state transition function on each action in $app(bs)$ which terminates as an operation on finite sets of belief state tuples. To check which of these belief states are old, new or updated, we need to compare each one of them with each belief state of the given process graph, which is a finite graph. Thus, handling an updated state terminates. Dealing with a new belief state bs terminates in a similar way as we again have to compute $app(bs)$, the belief states following bs and classify them as old, new or updated. To sum up, each traversal step terminates. Hence, it suffices to show that the number of traversal steps is finite which follows from the fact that the number of distinct belief states one can reach from an initial state by combining and conducting actions from a finite set of actions is finite.

Adding a goal state. The case of adding a goal state *goal* to the set of goal states *GOALS* is handled by a depth-first search through the belief states of the given process graph. Since the number of these belief states is finite (cf. Definition 5), it suffices to show that the traversal of each belief state terminates. Thus, we examine the traversal of a belief state *bs*. As a first step, we check whether *bs* meets *goal* (i.e., $\forall p \in goal$: $\exists p' \in bs$, $v(p)=v(p')$, $r(p') \subset r(p)$), which is a comparison of the restrictions of a finite number of belief state tuples and hence terminates. If *bs* indeed meets *goal*, the traversal of *bs* ends. Otherwise, we try to reach a belief state meeting *goal* by conducting planning steps, which terminates as well.

Removing a goal state. For the finite number of paths of the given process graph which end at a belief state meeting *goal*, it is checked whether they can be extended so that they lead to one of the remaining goal states. This is done by conducting planning steps for a finite number of belief states, which terminates.

Updating a goal state. Strengthening update. For the finite number of paths of the given process graph which end at a belief state meeting *goal*, it is checked whether they can be extended so that they lead to one of goal states from $GOALS'=(GOALS \setminus \{goal\}) \cup \{goal'\}$. This is done by conducting planning steps for a finite number of belief states, which terminates.

Updating a goal state. Weakening update. The case of updating a goal state *goal* to a goal state *goal'* which weakens the conditions of *goal* is handled by a depth-first search through the belief states of the given process graph. Since the number of these belief states is finite (cf. Definition 5), it suffices to show that the traversal of each belief state terminates. Thus, we examine the traversal of a belief state *bs*. As a first step, we check whether *bs* meets *goal'* (i.e., $\forall p \in goal'$: $\exists p' \in bs$, $v(p)=v(p')$, $r(p') \subset r(p)$) which is a comparison of the restrictions of a finite number of belief state tuples and hence terminates. If *bs* indeed meets *goal'*, the traversal of *bs* ends. Otherwise we try to reach a belief state meeting *goal'* by conducting planning steps, which terminates as well.

Adding an action. Adding an action *a* to the set of actions is handled by a depth-first search through the belief states of the given process graph. From each such belief state, planning steps are conducted in order to find belief states in which *a* is applicable and hence new feasible paths can possibly be constructed. As the conduction of planning steps terminates and the given process graph has a finite number of belief states (cf. Definition 5), the depth-first search terminates as well.

Removing an action. When removing an action *a* from *A* so that $A'=A \setminus \{a\}$, the finite set of all feasible paths of the given process graph is traversed. It is checked whether such a feasible path contains the action *a* in order to determine whether this path is retained. These checks terminate as each feasible path contains only a finite number of actions and thus our approach terminates.

Updating an action. Strengthening update of the preconditions. Again, all feasible paths of the given process graph are traversed in order to check whether in the belief states in which *a* was applicable, the updated action *a'* is applicable as well. If this is not the case, the path is not

considered in the adapted process graph and the traversal of this path ends. On the other hand, if *a'* is applicable in a belief state *bs* with $a \in app(bs)$, it is checked whether $R(bs,a)$ and $R(bs,a')$ coincide, which requires a finite amount of set comparisons and hence terminates. Finally, if these belief states do not coincide, we follow the approach for updating the initial state, which, as seen above, terminates. As the given process graph contains a finite number of feasible paths, this means that our approach addressing the strengthening update of the preconditions of an action terminates.

Updating an action. Weakening update of the preconditions. This case is handled by a depth-first search through the belief states of the given process graph. From each belief state, planning steps are conducted in order to find belief states *bs* with $a \notin app(bs)$ and $a' \in app(bs)$ and possibly construct new feasible paths containing *a'*, which terminates. Additionally, there may be belief states *bs* in which both *a* and *a'* are applicable. In this case, it is checked whether $R(bs,a)$ and $R(bs,a')$ coincide, which requires a finite amount of set comparisons and hence terminates. If these belief states do not coincide, we apply our approach for updating the initial state, which, as seen above, terminates. As the given process graph has a finite number of belief states (cf. Definition 5), the depth-first search terminates as well.

Updating the effects. When updating the effects of an action *a* resulting in the action *a'*, we traverse each belief state of the given process graph in which *a* is applicable. As by Definition 5 the given process graph contains a finite number of belief states, the set of such belief states is finite as well. For each such belief state *bs*, we treat $R(bs,a')$ as the update of $R(bs,a)$ and apply our approach for updating the initial state, which terminates as seen above.

Thus, Theorem 3 is shown for each case and hence proven. *q.e.d.*

B. Pseudocode of the presented Approach

```
def updateinit(updated_inital_state, original_inital_state):
        handleUpdatedState(updated_inital_state, original_inital_state)

def handleUpdatedState(updated_state, original_state):
        if checkForGoal(updated_state):
                return
        old_actions = originalModel.getFollowingActions(updated_state)
        new_actions = actionLibrary - old_actions
        for action in old_actions:
                if (action.preconditions.containsVariable(updated_belief_state_tuple) and
                        isApplicable(action, updated_belief_state_tuple)) or
                        not action.preconditions.containsVariable(updated_belief_state_tuple):
                                old_following_state =
originalModel.getFollowingState(original_state, action)
                                new_following_state =
old_following_state.update(updated_belief_state_tuple)
                                adaptedModel.addTransition(updated_state, action, new_fol-
lowing_state)
                                if originalModel.contains(new_following_state):
                                        adaptedModel.addAll(originalModel.getSubgraphFrom-
State(new_following_state))
                                else:
                                        handleUpdatedState(new_following_state)
        for action in new_actions:
                if action.preconditions.containsVariable(updated_belief_state_tuple) and
                        isApplicable(action, updated_belief_state_tuple):
                        following_state = apply(action, updated_state)
                        adaptedModel.addTransition(updated_state, action, following_state)
                        if originalModel.contains(following_state):
                                adaptedModel.addAll(originalModel.getSubgraphFrom-
State(following_state))
                        elif isUpdatedState(following_state):
                                handleUpdatedState(following_state)
                        else:
                                handleNewState(following_state)
```

```
def handleNewState(new_state):
    if checkForGoal(new_state):
        return
    following_states = planStateTransitions(getApplicableActions(new_state), new_state)
    for following_state in following_states:
        if originalModel.contains(following_state):
            adaptedModel.addAll(originalModel.getSubgraphFromState(following_state))
        elif isUpdatedState(following_state):
            handleUpdatedState(following_state)
        else:
            handleNewState(following_state)

def addGoal(new_goal_state):
    adaptedModel.addGoal(new_goal_state)
    for state in originalModel.states:
        if checkForParticularGoal(state, new_goal_state):
            adaptedModel.removeTransitions(originalModel.getSubgraphFromState(state))
    for state in unplannedStates(originalModel):
        if checkForParticularGoal(state, new_goal_state):
            adaptedModel.addTransitions(originalModel.getTransitionsFromTo(inital_state, state))

def removeGoal(old_goal_state):
    adaptedModel.removeGoal(old_goal_state)
    for state in pathEndingStates:
        if checkForParticularGoal(state, old_goal_state):
            planSubGraphFromState(adaptedModel, state)
    adaptedModel.removeTransitionsNotLeadingToGoalStates()

def updateGoal(updated_goal_state, original_goal_state):
    if isStrengtheningUpdate(updated_goal_state, original_goal_state):
        adaptedModel.addGoal(new_goal_state)
        removeGoal(original_goal_state)
    elif isWeakeningUpdate(updated_goal_state, original_goal_state):
        adaptedModel.removeGoal(original_goal_state)
        addGoal(updated_goal_state)
    else:
        addGoal(updated_goal_state)
        removeGoal(original_goal_state)
```

```
def addAction(new_action):
        actionLibrary.add(new_action)
        for state in originalModel.states:
                if isApplicable(new_action, state):
                        following_state = apply(new_action, state)
                        adaptedModel.addTransition(state, new_action, following_state)
                        planSubGraphFromState(adaptedModel, following_state)
        for state in unplannedStates(originalModel):
                if isApplicable(new_action, state):
                        adaptedModel.addTransitions(originalModel.getTransitionsFromTo(in-
ital_state, state))
                        following_state = apply(new_action, state)
                        adaptedModel.addTransition(state, new_action, following_state)
                        planSubGraphFromState(adaptedModel, following_state)

def removeAction(old_action):
        actionLibrary.remove(old_action)
        for transition in adaptedModel.stateTransitions:
                if old_action in transition:
                        adaptedModel.removeTransition(transition)
        adaptedModel.removeTransitionsNotLeadingToGoalStates()

def updateAction(updated_action, original_action):
        actionLibrary.remove(original_action)
        actionLibrary.add(updated_action)
        if updated_action.preconditions != original_action.preconditions:
                if isStrengtheningUpdate(updated_action.preconditions,original_action.precon-
ditions):
                        strengtheningUpdatePreconditions(adaptedModel, updated_action,
original_action)
                elif isWeakeningUpdate(updated_action.preconditions,original_action.precon-
ditions):
                        weakeningUpdatePreconditions(adaptedModel, updated_action, origi-
nal_action)
                else:
                        weakeningUpdatePreconditions(adaptedModel, updated_action, origi-
nal_action)
                        strengtheningUpdatePreconditions(adaptedModel, updated_action,
original_action)

        if updated_action.effects != original_action.effects:
                for transition in adaptedModel.stateTransitions:
                        if original_action in transition:
                                replaceTransitionAndUpdate(transition, updated_action)
```

```
def strengtheningUpdatePreconditions(adaptedModel, updated_action, original_action):
    for transition in adaptedModel.stateTransitions:
        if original_action in transition:
            if isApplicable(updated_action, transition.fromState()):
                replaceTransitionAndUpdate(transition, updated_action)
            else:
                adaptedModel.removeTransition(transition)
                adaptedModel.removeTransitionsNotLeadingToGoalStates()

def weakeningUpdatePreconditions(adaptedModel, updated_action, original_action):
    for state in originalModel.states:
        if isApplicable(original_action, state):
            transition = adaptedModel.findTransition(state, original_action)
            replaceTransitionAndUpdate(transition, updated_action)
        elif isApplicable(updated_action, state):
            following_state = apply(updated_action, state)
            adaptedModel.addTransition(state, updated_action, following_state)
            planSubGraphFromState(adaptedModel, following_state)

    for state in unplannedStates(originalModel):
        if isApplicable(updated_action, state):
            adaptedModel.addTransitions(originalModel.getTransitionsFromTo(in-
ital_state, state))
            following_state = apply(updated_action, state)
            adaptedModel.addTransition(state, updated_action, following_state)
            planSubGraphFromState(adaptedModel, following_state)

def replaceTransitionAndUpdate(transition, updated_action):
    adaptedModel.removeTransition(transition)
    old_following_state = transition.fromState()
    updated_belief_state_tuple = applyForUpdatedBeliefState(updated_action, old_fol-
lowing_state)
    new_following_state = old_following_state.update(updated_belief_state_tuple)
    adaptedModel.addTransition(transition.fromState(), updated_action, new_follow-
ing_state)
    handleUpdatedState(new_following_state)
```

C. Evaluation of Computational Complexity

In the following, we outline the differences in complexity between the presented adaptation of a process graph and planning the adapted process graph from scratch. To this end, we use the notation found in Table 7. If necessary, further notation is provided for each adaptation case.

k	Number of belief states that are planned or otherwise known during planning
k_{old}	Number of belief states in the original process graph
k_{new}	Number of belief states in the adapted process graph
$k_{unplanned}$	Number of belief states which are reachable from the initial state, but not planned in the process graph (i.e., they do not lead to a goal state)
n	Number of all actions
m	Number of all belief state variables
g	Number of goal states

Table 7. Notation

Updating the initial state. Let $n_{old}=|app(bs_{old})|$ be the number of actions applicable in an old belief state. Evaluating the applicability in such a belief state can be done just for the updated belief state tuple. The same holds for the application of the transition function. Hence, these two steps require no more than $2+n_{old}$ comparisons using the presented approach versus $m*(2+n_{old})$ comparisons when planning from scratch. Having determined the following belief states to each applicable action, the effort of checking whether these states are already planned or meeting a goal state condition is the same for both approaches with $(k+g)*m$ comparisons. Please note that for every belief state, which is contained in the original process model the entire subgraph is adopted by our adaptation approach which takes (virtually) no effort. In contrast, when planning from scratch in a worst case scenario every combination of actions is feasible and thus $n!$ planning steps are required with each planning step consisting of $(k+g+3)*m*n$ comparisons.

Adding a goal state. As shown in Section 4.2.1, adding a goal state is addressed in two possible ways. Firstly, paths are shortened by checking each belief state of the existing process graph for the goal condition of the added goal state. Once this check yields true, removing all following edges and nodes is of insignificant computational cost which leads to a total of $k_{old}*m$ comparisons needed versus at least k_{new} planning steps with $(k+g+3)*m$ comparisons when planning from scratch. Secondly, new feasible paths are planned which lead to the added goal state by conducting $k_{unplanned}$ planning steps (versus $k_{old}+k_{unplanned}$ planning steps when planning from scratch). The reduction of complexity is even more substantial if all reachable belief states have been stored in the course of computing the original process graph. In this scenario, the presented approach does not need to execute planning steps. Instead, all reachable states have to be checked regarding the added goal state condition which in total requires $(k_{goal}+k_{unplanned})*m$ comparisons.

Removing a goal state. Let k_{goal} be the number of belief states, which meet the goal condition of the removed goal state. The task at hand is to search for new paths to the remaining goal states beginning from the belief states meeting the removed goal state condition. Thus, the presented approach reuses and modifies the original process graph where necessary by conducting planning from the aforementioned belief states. This leads to $k_{goal}*g*m$ comparisons and $k_{unplanned}$ planning steps when adapting the process graph compared to $k_{old}+k_{unplanned}$ planning steps when planning from scratch. Again, if all reachable belief states are accessible, the complexity can be reduced to $(k_{goal}+k_{unplanned})*g*m$ comparisons to check all reachable states regarding the remaining goal state conditions.

Updating a goal state. In the worst possible case, the update of a goal state is addressed by the two steps above (i.e., adding the updated goal state and thereafter removing the obsolete goal state). With this in mind, the computational effort of these two steps can be added and compared to planning the updated process graph from scratch leading to $k_{old}*m+k_{goal}*g*m+k_{unplanned}*(k+g+3)*m*n$ comparisons versus at least k_{new} planning steps with $(k+g+3)*m*n$ comparisons.

Adding an action. Again, the presented approach fully makes use of the original process graph and tries to plan new paths by applying the added action where possible. Contrarily, planning the original process graph from scratch amounts to at least k_{old} planning steps, each containing $(k+g+3)*m*(n-1)$ comparisons. In both approaches the applicability of the added action is checked for each state of the original process graph which accounts for $m*k_{old}$ comparisons. If applicable, the state transition ($2*m$ comparisons) as well as further planning steps ($(k+g+3)*m*n$ comparisons each) are computed. As the added action can be applicable in reachable belief states, which are not contained in the original process graph, the computation of these belief states can be skipped when adapting the process graph. Here, only the applicability of the added action is determined, resulting in $m*k_{unplanned}$ comparisons opposed to $k_{unplanned}$ planning steps with $(k+g+3)*m*n$ comparisons.

Removing an action. The presented approach identifies and deletes all paths that contain the removed action, which has no significant computational complexity. However, as seen above, planning the adapted graph from scratch requires at least k_{new} planning steps.

Updating an action. Updating the effects of an action does not affect its applicability. Instead, each following belief state has to be updated regarding the updated belief state tuple, which requires 2 comparisons. Afterwards, each updated state is handled in the same way as an updated initial state. Hence, we refer to the discussion above. When conducting a weakening update of the preconditions, the presented approach proceeds is similar to adding an action. Additionally, belief states, which follow the updated action in the original process graph, might be updated and treated as above. Analogously, a strengthening update of the preconditions leads to the removal of each path containing the updated action if it is not applicable. Otherwise, the following belief state is updated and handled accordingly.

Overall, the complexity analysis shows that our approach provides considerable advantages regarding computational complexity compared to planning process graphs from scratch.

References

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2006). "Strong planning under partial observability" *Artificial Intelligence* 170 (4–5), 337–384.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models: Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

4.3 Paper 8: The Cooperation of Multiple Actors within Process Models: An Automated Planning Approach

Current Status	Full Citation
accepted and published (04/2019) in Volume 27, Issue 4 of *Journal of Decision Systems*	Heinrich, B., A. Schiller, and D. Schön (2018). "The Cooperation of Multiple Actors within Process Models: An Automated Planning Approach". *Journal of Decision Systems (JDS)* 27 (4), 238-274.

Summary

This paper addresses RQ8 by presenting a conceptual foundation for multi-actor process models, based on which an automated planning approach (comprising an algorithm) for constructing such models is proposed. Extending existing single-actor planning approaches, the conceptual foundation supports the needs of multi-actor processes, for instance enabling actor-specific initial states and goal states. Moreover, the approach includes the cooperation of actors in the control flow of process models by constructing explicit actions which determine when to jointly conduct actions. The constructed multi-actor process models are proven to be correct and complete. Further, the feasibility and effectiveness of both conceptual foundation and planning approach are demonstrated by an application to several real-world scenarios from different contexts and the assessment of a practitioner in an experimental setting from healthcare.

Similar to Paper 6 and Paper 7, the work builds on a conceptual basis from AI planning, comprising, for instance, belief states, actions, applicability and planning graphs as part of the underlying single-actor planning domain. The paper iteratively extends these concepts to facilitate the consideration of actor-specific information, the conduction of actions by multiple actors and actor-cooperation. Similarly, the proposed algorithm for planning multi-actor process models enhances existing single-actor planning methods. The paper promotes business process agility by enabling the rapid construction of multi-actor process models which are shown to adequately represent multi-actor processes conducted in practice. Further, incorporating the cooperation of multiple actors in the control flow of process models helps individual actors to achieve their individual goals from a decision support perspective.

Please note that the paper has been adjusted to the remainder of the dissertation with respect to overall formatting and citation style. Moreover, terms only common in British English have been converted to corresponding American English terms.

The paper as published by Taylor&Francis is available at:
https://doi.org/10.1080/12460125.2019.1600894

Abstract:

In many business processes, multiple actors such as different employees, departments or companies are involved. These actors need to work together and form appropriate partnerships in parts of these processes to achieve their individual goals. Hence, from a business process management perspective, the actors need to cooperate. We present a conceptual foundation for multi-actor process models, which enables the consideration of individual starting points and goals as well as partnerships. Further, we incorporate the cooperation of actors in the control flow of process models by constructing explicit actions determining where in the process to form and disband partnerships. We pursue an automated planning approach due to the complexity of the required cooperation. The constructed multi-actor process models are proven to be correct and complete. We demonstrate the feasibility of our approach by an application in several real-world scenarios and its effectiveness through the assessment of a practitioner.

Keywords: business process management, multi-actor processes, process modeling

1 Introduction

Ever-increasing competition in today's business world requires companies to reduce costs and increase their efficiency. Hence, companies need to consider economic effects of, for instance, strengthening their own capabilities in a particular business area (Forstner et al., 2014) to stay competitive. With companies focusing on particular capabilities, oftentimes multiple actors such as different departments, companies or individuals are involved in the conduction of a business process (cf., e.g., Davenport and Short, 1990; The Workflow Management Coalition Specification, 1999). These conducting actors *cooperate* by forming so called *partnerships* – which means, sets of selected actors (cf. Grefen et al., 2000; Huang et al., 2013; Leymann et al., 2002) jointly conduct parts of the process (Pulgar and Bastarrica, 2017). In a similar vein, Serve et al. (2002) state that 'business processes are linked and managed across multiple companies' as it could be beneficial for a company to source out parts of its business processes (Katzmarzik et al., 2012). In such inter-organizational processes, each conducting actor (e.g., suppliers, partnering companies or customers) usually starts at an individual starting point, follows its own individual goals (cf., e.g., Becker et al., 2013; Chiu et al., 2003; Skjoett-Larsen et al., 2003) but needs to cooperate with other conducting actors (cf. Lambert et al., 1996; Stadtler et al., 2015). Similarly, multiple actors (e.g., departments or employees of a company) cooperate within intra-organizational processes (cf., e.g., Ghrab et al., 2017). In either case, the partnerships formed to jointly conduct actions during the process are usually not required throughout the entire process, but just for certain parts of it (Grefen et al., 2000). To give an example, customers can possibly conduct parts of a process on their own by using self-service technologies (Klier et al., 2016). A process comprising partnerships of actors that conduct parts of it jointly and in which actors start at an individual starting point and follow individual goals is referred to as a *multi-actor process* in the following.

Besides this discussion based on scientific literature, the relevance of multi-actor processes can also be reflected from a practical perspective. For instance, in cooperations with two European financial services providers (a bank and an insurance company) we supported an analysis of over 600 (core) processes from – amongst others – the divisions credit lending and securities trading (in case of the bank) and the general project management department (in case of the insurer). The aim of the cooperations was to increase transparency (e.g., definition of responsibilities) and efficiency regarding economic indicators and capacities of these processes. Therefore, detailed data for the processes themselves as well as the involved departments and actors was ed. In this context, we examined – amongst other characteristics – which of these processes are multi-actor processes, which means, whether several actors in terms of employees and departments of the financial services providers as well as external service providers and customers have to work together and thus form partnerships in parts of the processes. Our analyses showed the following results: Partnerships of at least two actors conduct parts of the process (i.e., actions) jointly in more than 90% of all considered insurer processes resp. more than 70% in case of the bank processes. Partnerships of three or more actors are comprised in more than 60% of the insurer processes resp. in more than 50% of the bank processes. Thereby, these actors do not necessarily represent individual employees but also departments that usually comprise more than one individual. Hence, the aforementioned sizes of the partnerships serve as lower bounds and, in a particular process execution, usually more individuals are involved. The examined partnerships were used for several reasons by both companies: For example, in many processes, they allowed a high utilization of resources and an efficient workload of employees. Furthermore, in some cases, they were required to ensure legal and regulatory compliance (e.g., to realize a dual or triple control principle). The security order management process of the bank may serve as an example for a multi-actor process: A number of brokers and order processing specialists, the internal risk assessor as well as external contractors are just some of the indispensable actors in this process to conduct actions jointly. This illustrates the motivation and importance of partnerships and jointly conducted actions in practice. Besides, we refer to the Section Evaluation, where an evaluation of the approach provided in this paper by means of several of these processes is discussed and concrete key properties for the processes (e.g., the number of involved partnerships) are presented.

After discussing the relevance and importance of multi-actor processes in research and practice, we will focus on how multiple actors in process models are currently addressed within the research field of BPM (business process management; e.g., Chinosi and Trombetta, 2012) as process models are an established way to represent processes. Within different well-known process modeling languages, concepts for representing multiple actors exist. For example, the languages BPMN and UML comprise so called swimlanes that allow to associate actions to one specific actor that needs to conduct these actions (Object Management Group, 2013, 2015). These swimlanes merely serve as an annotation. However, the association of actions to conducting actors by swimlanes is not sufficient for reliably forming appropriate partnerships (cf., e.g., Kossak et al., 2016; Natschläger and Geist, 2013; Pulgar and Bastarrica, 2017; Recker et

al., 2006; Wohed et al., 2006). In particular, the lack of expressiveness is considered a major weakness of swimlanes (Kossak et al., 2016; Natschläger and Geist, 2013). In this context, lack of expressiveness means that it is hardly possible to express that an action needs to be performed by a particular number of selected actors jointly, which means to express the required size and composition of a partnership. Pulgar and Bastarrica (2017) highlight this issue and state that 'there is no natural way to represent collaborative activities performed by different roles' (i.e., actors in our terms). Hence, in order to represent multi-actor processes with possibly individual starting points and goals for each actor (cf. Aspect (A1)) comprising actions performed by partnerships (each with a required size and composition; cf. Aspect (A2)) by means of established process modeling languages, a new approach needs to be developed. Further, from a decision support perspective, it is promising to support individual actors explicitly when and with whom to cooperate (Peleteiro et al., 2014). We therefore aim at incorporating the cooperation of multiple actors in the control flow perspective of process models (van der Aalst et al., 2003; van der Aalst and van Hee, 2002) by planning explicit actions determining when to form and when to disband appropriate partnerships (Aspect (A3)). Constructing a multi-actor process model based on this conceptual foundation – instead of using annotations such as swimlanes mentioned above – is envisioned in order to increase the expressiveness of multi-actor process models. We therefore want to take the Aspects (A1) to (A3) into account and state our first research question of *how a conceptual foundation to represent multi-actor process models can be specified.*

Besides addressing Aspects (A1) to (A3), as process (re)design projects and process models are becoming increasingly large and complex (Hornung et al., 2007), constructing process models manually develops into a more and more difficult and error-prone task. More precisely, according to Mendling et al. (2008), larger (they refer to 40 actions and more) and more complex process models particularly tend to contain more errors when constructed manually. Empirical studies of Roy et al. (2014) and Fahland et al. (2011), for instance, show that up to 92.9 % of process models are erroneous in industrial contexts. Besides semantic errors (e.g., missing actions), in particular, syntactical errors such as hanging nodes and ambiguous gateways are contained in these process models. Even though these errors do not render the process models completely worthless, they make it very difficult to use the models for potential process improvements or to apply several approaches for the automated verification (Weber et al., 2008) and execution (Khan et al., 2010; Weber, 2007), for instance. Further, compared to constructing single-actor process models, constructing multi-actor process models manually is even more complex and error-prone as it poses additional challenges (Aspects (A1) to (A3)). For example, individual starting points and goals, actors cooperating in partnerships and the size and composition of these partnerships need to be taken into account. We thus strive to address the construction of multi-actor process models *in an automated manner* and state our second research question of *how feasible multi-actor process models can be constructed by means of an automated planning approach*.

The second research question is in accordance with the emergence of several approaches to support modelers and business analysts by means of automation (e.g., algorithms) in the last years. For instance, process mining (e.g., IEEE Task Force on Process Mining, 2012; van der Aalst et al., 2004; van der Aalst, 2015) assists business analysts especially in the *process analysis* phase of the BPM Lifecycle (cf. Wetzstein et al., 2007). Automated service selection and composition (e.g., Ding et al., 2015; Paik et al., 2014; Wang et al., 2014) increase the degree of automation within the phases *process implementation* and *process execution*. Our second research question falls within the research strand automated process planning, which envisions the construction of process models in an automated manner by means of algorithms (Heinrich et al., 2015; Heinrich and Schön, 2015; Henneberger et al., 2008; Hoffmann et al., 2012) to support modelers in the phase of *process modeling*.

To sum up, we present an approach for the *automated planning of multi-actor process models* (second research question) based on a *conceptual foundation* (first research question) that incorporates the cooperation of conducting actors. The main contributions are as follows:

❶ *Conceptual foundation for multi-actor process models (Aspects (A1) to (A3))*. We propose a conceptual foundation that enables the consideration of individual starting points and goals of conducting actors as well as partnerships that need to conduct actions jointly. These partnerships are of a particular size and consist of specific actors. The conceptual foundation further includes the cooperation of conducting actors within the control flow of process models. Cooperation is expressed by explicit actions denoting where and with whom to form and disband partnerships.

❷ *Automated planning of multi-actor process models*. We propose an automated planning approach, the first to support a construction of feasible, correct and complete multi-actor process models.

In the remainder of this paper we follow the research approach as presented by Bertrand and Fransoo (2002) as well as Mitroff et al. (1974) and its phases conceptualization, modeling, model solving, and implementation: After this introduction of the problem context (conceptualization), we discuss related work in the next section. Thereafter we introduce our planning domain and the running example we use to illustrate our approach. Subsequently, we present a conceptual foundation for multi-actor process models (i.e., a 'model of the object reality'; cf. Meredith et al., 1989; modeling) and discuss how the construction of multi-actor process models can be supported by means of the proposed automated planning approach (model solving by means of an algorithm). In the penultimate section, we evaluate our approach in terms of its termination as well as the completeness and correctness of the constructed process models (i.e., 'proof of the solution'; cf. Bertrand and Fransoo, 2002). We further demonstrate the feasibility of our approach by means of a software prototype (implementation) as well as an application to different real-world scenarios as proposed by Meredith et al. (1989). Moreover, we evaluate its effectiveness by an application in an experimental real-world scenario together with a practitioner. Thereby we aim at evaluating in how far our approach is able to construct multi-actor

process models that reflect processes as actually conducted in reality according to the assessment of a practitioner. Finally, we summarize our considerations, discuss limitations and provide an outlook on future steps.

2 Related Work

In this section, we give an overview of how different fields of research address multi-actor processes and the construction of multi-actor process models. We (1) introduce approaches dealing with multi-actor processes and workflows within the general field of BPM before (2) distinguishing existing approaches within the focused research field of automated planning from ours. Finally, we briefly analyze the related areas process mining (3) and multi-agent-systems / autonomous systems (4).

Ad (1): Multi-actor processes are heavily discussed within the research field of BPM (cf., e.g., Chen and Hsu, 2001; Fleischmann et al., 2013; Jennings et al., 2000; Kannengiesser, 2017).

To begin with, the swimlanes in modeling languages such as BPMN and UML allow the annotation of multi-actor processes (Object Management Group, 2013, 2015), but do not specify a conceptual foundation for multi-actor process models (cf. contribution ❶). Shapiro et al. (2012) discuss different possibilities to represent actions that need to be performed by partnerships by means of swimlanes. However, each of these possibilities has major shortcomings. For instance, one proposition is to duplicate actions in the swimlane for each actor that jointly conducts the action. This results in ‘messy and difficult to understand’ process models (Pulgar and Bastarrica, 2017). In contrast, a cooperation of actors is also discussed in the research field of workflow management, where cross-organizational and collaborative workflows (cf. Boukhedouma et al., 2017; Liu et al., 2015; Liu and Zhang, 2016; van der Aalst, 1999) are examined. Here, ‘some tasks can only be executed by certain business partners and a case always resides at exactly one location’ (van der Aalst, 1999). However, these works do not present a conceptual foundation for multi-actor process models (cf. contribution ❶). Further, they do not aim to plan multi-actor process models in an automated manner but mostly rely on already existing process models (cf., e.g., Boukhedouma et al., 2017), and thus do not address contribution ❷.

Other works in the field of BPM consider so called agents (corresponding to ‘actors’ in our sense) as (decentrally) acting entities, that need to interact with each other during the execution of a process in order to reach their individual goals (Jennings et al., 2000; Kannengiesser, 2017). In particular, these agents apply (bilateral) communication and collaboration during process execution to align their individual tasks with others. However, such a consideration during the execution of a process is not beneficial in all cases. For instance, processes that are conducted within a department of an organization or even inter-organizational processes usually are controlled, modeled and managed across the involved actors in order to ‘reduce operating cost, improve customer service and expand into markets’ (Serve et al., 2002). We therefore aim at

constructing multi-actor process models (at design time; cf. contribution ❷) instead of a communication-based approach that takes place during the execution of a process.

Moreover, works in the research field of resource management deal with the task of automated team selection and allocation (cf., e.g., Cabanillas et al., 2015; Havur et al., 2015, 2016). For example, Cabanillas et al. discuss a language for the description of teams (corresponding to 'partnerships' in our sense) in process models, which partly focuses on Aspect (A2), but does not cope with Aspects (A1) and (A3) (cf. ❶). Particularly, they extend the 'organizational metamodel' (Russell et al., 2005) by means of team-related concepts such as 'TeamRoles' (i.e., the role a person has in a team) or sizes of teams. However, these approaches do not aim to construct process models (cf. ❷) and instead rely on a given process model.

Ad (2): Within the field of automated planning, several approaches dealing with the problem of planning in so called multi-agent environments exist. A survey conducted by de Weerdt and Clement (2009) classifies these approaches into two basic categories: Planning by multiple instances (i.e., different planners) and planning for multiple actors. Approaches considered in the first category strive to distribute the problem of planning among several instances (cf., e.g., Torreño et al., 2012), which is not in the scope of this paper. The approaches of the second category address the problem of planning for multiple actors in different ways. Dimopoulos and Moraitis (2006) aim to coordinate two actors with individual plans (i.e., processes), where one actor has to provide his/her plan proposals to a second actor based on which the second actor has to construct non-conflicting (to the provided plan proposal) plans to achieve the goals of both actors. Nissim et al. (2010) address the problem of planning in multi-actor environments by means of single-actor planning approaches, conducted by each actor in a distributed manner. In a second step, the corresponding single-actor plans are matched by means of 'seeking sequences of public actions [i.e., single-actor plans] that satisfy a certain CSP [constraint satisfaction problem]' (Nissim et al., 2010). Other approaches (cf., e.g., Crosby et al., 2013; Ephrati and Rosenschein, 1994) deal with the problem by decomposing a general multi-actor process into smaller, single-actor processes in a first step and conducting a single-actor planning for each of those subparts in a second step. This is also envisioned within the research field of (web) service composition (cf. Falou et al., 2009). However, none of these approaches aims to provide a conceptual foundation for multi-actor process models (cf. contribution ❶), which makes them substantially different from ours. Further planning approaches considering multiple actors exist, but the planning domain they use is fundamentally different from the planning domain needed for the automated planning of process models, for instance, by not considering actor-specific goal states (Torreño et al., 2014a). To give another example, Chouhan and Niyogi (2017) address the issue of different actors having to perform different actions simultaneously to achieve a common goal and hence conduct a planning approach to "synchronize" these actors. In contrast, on the one hand, we aim at the cooperation of different actors performing one or more actions (or parts of processes) jointly instead of actors performing their actions separately but simultaneously. On the other hand, the planning domain they use does not aim to cope with actor-specific goal states as well, which is needed for the automated planning of

multi-actor process models. Moreover, some of these approaches are – additionally to their different aims – based on heuristic techniques and do not provide a complete solution, which means, the constructed graphs do not contain all feasible paths (Štolba and Komenda, 2013; Torreño et al., 2014b). Hence, these approaches do not fit the needs of automated planning of process models.

Ad (3): The research field of process mining addresses, amongst others, the issue of reconstructing process models from event logs in an automated manner. Here, as well, approaches to reconstruct models of processes with multiple actors (cf., e.g., Ou-Yang and Winarjo, 2011; Rozinat et al., 2009) or process models with a consideration of 'resources' executing the tasks (Schönig et al., 2015) exist. However, process mining follows an *as-is* perspective (cf. Rosemann and vom Brocke, 2015) by reconstructing process models that denote already implemented and executed processes. In contrast, automated planning follows a *to-be* perspective as it strives to construct new process models (cf. contribution ❷). Furthermore, existing approaches do not aim at providing a conceptual foundation for multi-actor process models (cf. contribution ❶).

Ad (4): The research fields of multi-agent-systems (Shoham and Tennenholtz, 1995; Wooldridge, 2009; Zhang, 2017) and autonomous systems (Dobson et al., 2006) aim to address the cooperation of multiple agents (in our terms, 'actors') in processes. They do so during the execution of a process based on communication between the actors or between actors and a central coordination mechanism, which is a related but different task (cf. contributions ❶ and ❷).

To sum up: There are several valuable contributions regarding the consideration of multiple actors in process models in the literature (cf. Table 1). However, none of these works aim to incorporate the cooperation of multiple actors based on a conceptual foundation (cf. contribution ❶) in process models in an automated manner (cf. contribution ❷).

Research field	Analyzed related work	Phase of consideration	Conceptual foundation for multi-actor process models (cf. ❶)			Automated construction of multi-actor process models (cf. ❷)
			Consideration of individual starting points and goals (cf. Aspect (A1))	Consideration of required size and composition of partnerships (cf. Aspect (A2))	Incorporation of formation and disbandment of partnerships in control flow (cf. Aspect (A3))	
BPM (general)	Chen and Hsu (2001); Fleischmann et al. (2013); Jennings et al. (2000); Kannengiesser (2017)	Design time and execution time	Not considered	Not considered	Not considered	Not considered
Process modeling languages	Object Management Group (2013, 2015); Shapiro et al. (2012)	Design time	Not considered	Not considered	Not considered	Not considered
Workflow management	Boukhedouma et al. (2017); Liu et al. (2015); Liu and Zhang (2016); van der Aalst (1999)	Execution time	Not considered	Not considered	Not considered	Not considered; relying on already existing process models
Resource management	Cabanillas et al. (2015); Havur et al. (2015, 2016)	Execution time	Not considered	Partly considered	Not considered	Not considered; relying on already existing process models

Table 1. Overview of Related Work

Automated planning & Web service composition	Chouhan and Niyogi (2017); Crosby et al. (1994); Dimopoulos and Moraitis (2006); Falou et al. (2009); Nissim et al. (2010); Torreño et al. (2014a);	Design time	Not considered	Not considered	Not considered	Partly considered (e.g., heuristic approaches; different planning domains)
Process mining	Ou-Yang and Winarjo (2011); Rozinat et al. (2009	Design time	Not considered	Not considered	Not considered	Considered but as-is perspective (i.e., reconstructing models of already executed processes)
Multi-agent-systems	Dobson et al. (2006); Shoham & Tennenholtz (1995); Wooldridge (2009); Zhang (2017)	Execution time	Considered	Not considered	Not considered	Not considered

Table 1. Overview of Related Work (continued)

3 Planning Domain

Within this section, we introduce the planning domain, which we will extend to cope with multi-actor planning in the remainder of the paper. The automated construction of process models can be understood as a planning problem (e.g., Heinrich et al., 2009). More precisely, we have to abstract from an individual process execution and its world states in order to construct entire process models, valid for various process executions, resulting in a nondeterministic planning problem with belief states (Ghallab et al., 2004). Here, a belief state represents possibly infinite sets of world states. Hence, we use a general set-theoretic planning domain (cf. Ghallab et al., 2016; Ghallab et al., 2004) independent of a concrete representation language (e.g., process modeling language) for our approach. This ensures a maximum of compatibility with existing approaches in the literature (e.g., Bertoli et al., 2001; Bertoli et al., 2006; Heinrich et al., 2015; Heinrich and Schön, 2016; Meyer and Weske, 2006; Sycara et al., 2003) and enables a widespread use of our approach to construct multi-actor process models. Considering this planning domain, a bipartite *planning graph,* which consists of two types of nodes – representing *belief states* and *actions* – and edges is used.

To illustrate our approach and the planning domain, we will use an excerpt of a real-world human resources process at a university with several participating actors that need to cooperate. In this process, one of the two research project managers (*Bob* and *Danielle*), in a first step, checks the application documents sent by an applicant. If the application documents meet the requirements, the action *job interview* is conducted. Here, the personnel officer (*Eric*), the two research project managers and one additional (but not mandatory) chair member (*Silvia*) interview the applicant jointly. Further, if the applicant was convincing and the salary requirements of *Eric* and the applicant fit, s/he is engaged. In a next step, the results of the interview are filed. We will use this action *file interview results* (denoted by a rounded rectangle; belief states denoted by tables with a bold border) to illustrate the core concepts of the planning domain (cf. Figure 1).

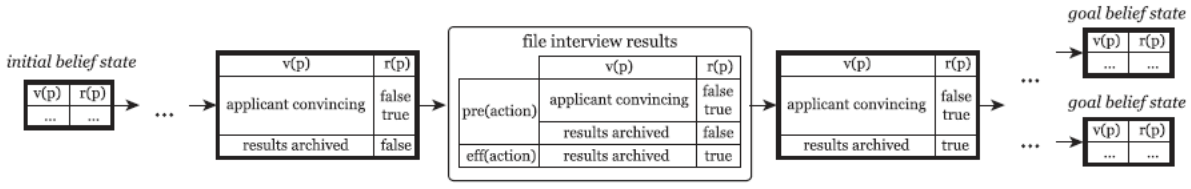

Figure 1. Illustrating the Action file interview results *in the Running Example*

A belief state *bs* can be seen as a set of information about the variables currently available in a process state (so called *belief state variables*). A belief state is a set of *belief state tuples* (denoted by rows in the tables in Figure 1), each of which denotes one particular characteristic. For instance, the belief state tuple *(results archived, {false})* in the belief state before the action *file interview results* expresses that at this state in the process, the results have not yet been archived.

Definition 1 *(belief state tuple).* A *belief state tuple p* is a tuple consisting of a *belief state variable v(p)* and a subset *r(p)* of its predefined *domain dom(p)*, which we will write as *p:=(v(p),r(p))*. The domain, *dom(p)*, specifies which values can generally be assigned to *v(p)*. The set $r(p) \subseteq dom(p)$ is called the *restriction* of *v(p)* and contains the values that can be assigned to *v(p)* in this specific belief state tuple *p*.

Definition 2 *(belief state).* A belief state *bs* is a finite set of belief state tuples, containing every belief state variable one time at the most. In the following, *BS* is a finite set of belief states.

To represent *actions* (denoted by rounded rectangles) conducted by an actor during a process, a second type of node is defined:

Definition 3 *(action).* Let *BST* be a finite set of belief state tuples. An action *action* is a triple consisting of the action name and two sets, which we write as *action:=(name(action), pre(action), eff(action))*. The set $pre(action) \subseteq BST$ are the preconditions of the action *action*, which describe the circumstances under which *action* can be applied and the set $eff(action) \subseteq BST$ are the effects of the action *action*, denoting the consequences that result from applying *action*. In the following, *ACTIONS* is a finite set of actions.

Definition 4 (*applicability*). An action *action* is *applicable* in a belief state *bs* iff $\forall p \in pre(action)\ \exists q \in bs: v(p)=v(q) \wedge r(p) \cap r(q) \neq \emptyset$. In other words, *action* is applicable in *bs* iff all belief state variables in *pre(action)* also exist in *bs* and the respective restrictions of the belief state variables intersect.

The preconditions and effects of actions are denoted by the table underneath the action name (cf. Figure 1). The action *file interview results* is applicable if the belief state variable *applicant convincing* is either *false* or *true* and the belief state variable *results archived* is *false* in the previous belief state. Its effects set the belief state variable *results archived* to *true*. Based on Definitions 1-4, a planning graph can be generated by means of different existing algorithms that progress from an initial belief state to goal belief states (see, e.g., Bertoli et al., 2001; Bertoli et al., 2006; Heinrich et al., 2009). To each action *action* applicable in a belief state *bs* a state transition function *R(bs, action)* associates the next belief state. We define our planning graph as follows:

Definition 5 (*planning graph*). A planning graph is a bipartite, directed, finite graph *G=(NODES, EDGES)*, with the set of nodes *NODES* and the set of edges *EDGES*. The set of nodes *NODES* consists of two partitions: The set of action nodes *ACTIONS* and the set of belief state nodes *BS*. Each node $bs \in BS$ represents one distinct belief state in the planning graph. The planning graph starts with one initial belief state $Init \in BS$ and ends with one to possibly many goal belief states $Goal_j \in BS$.

As many real-world processes use large data types (e.g., many of the processes of the financial services providers mentioned in the introduction), a possibly infinite set of different process instances may exist. The above presented planning domain supports this subject, in contrast to

STRIPS (Fikes and Nilsson, 1971) and other planners based on a classical planning framework (e.g., Hoffmann et al., 2012; for a detailed discussion of this aspect, cf. Heinrich et al., 2015). However, as our approach extends existing single-actor planning approaches based on the introduced, common planning domain, existing works for the automated construction of control flow patterns within single-actor process models (e.g., Heinrich et al., 2015; Heinrich and Schön, 2016; Meyer and Weske, 2006) can be further used for this purpose. Thus, it is not necessary to address how to incorporate control flow patterns (van der Aalst et al., 2003) such as exclusive choice into multi-actor process models in this paper. Following this, we do not consider control flow patterns in our running example as well.

4 Approach to Construct Multi-Actor Process Models

We divide the overall goal of an automated planning of multi-actor process models (cf. contribution ❷) into sub goals in accordance with the previously discussed Aspects (A1) to (A3). At first, we extend the introduced planning domain to cope with actor-specific information. Thereafter, as actions may need to be conducted by partnerships (i.e., sets of selected actors, each with a particular size), we include cardinalities (i.e., the size of these partnerships) in the planning domain. Subsequently, we outline how to enable the cooperation of multiple actors (cf. contribution ❶). We will discuss each of these three sub goals:

Consider actor-specific information within the planning domain. To consider actor-specific information in our planning domain, we adapt Definitions 1-5. In particular, we extend the definition of belief state tuples to denote actor-specific variables in terms of a set-theoretic representation.

Consider cardinalities. Actions in a multi-actor process may need to be conducted by a certain number of actors. Therefore, we extend the definition of actions to represent a condition regarding the cardinality of a partnership, which is required to conduct a certain action.

Plan partnerships of actors. As actions in a multi-actor process may be required to be conducted jointly by multiple actors represented within different belief states, we propose the join of multiple belief states into one belief state, representing a partnership. Similarly, actions may be required to be conducted by a subset of the actors represented within a single belief state. Thus, we describe how to split belief states into multiple belief states with those subsets of actors. In this way, the envisioned concept for enabling the cooperation of actors by explicit actions is addressed.

After explicating these sub goals (1) to (3) in more detail in the next subsections, we present our algorithm for the automated planning of multi-actor process models in a final subsection.

4.1 Consider Actor-specific Information within the Planning Domain

As a first step, we describe how to represent actor-specific information in terms of the planning domain. Within the planning domain given by Definitions 1 to 5, there is no differentiation between *non-actor-specific* belief state variables and *actor-specific* belief state variables. Thus, there is no way to describe belief state variables related to a certain actor. This is insufficient for planning multi-actor processes: Not only information unrelated to a specific actor such as for example the availability of general resources or general process conditions is required in order to fully characterize the current process situation by means of belief states. Rather, actor-specific information such as the present status or capabilities of an actor needs to be included as well. Hence, we adapt the planning domain described in the previous section and distinguish between actor-specific and non-actor-specific belief state variables. To be more precise, we extend the previous definition of belief state tuples by a so called *actor specification a(p)* with $a(p) \subseteq ACTORS \cup \{non\text{-}actor\} \cup \{arbitrary\}$. Here, *ACTORS* represents the set of actors participating in the conduction of the process, *{non-actor}* serves as an identifier for non-actor-specific variables and *{arbitrary}* denotes not mandatory actor-specific belief state tuples which will be discussed later in this subsection. Formally, the extended definition of belief state tuples is as follows:

Definition 1' *(belief state tuple).* Let *ACTORS* be a finite set of actors participating in the conduction of a process. A belief state tuple *p* is a tuple of a *belief state variable v(p),* its *restriction r(p),* a subset of its predefined domain *dom(p),* and the *actor specification a(p),* which is written as *(v(p),r(p),a(p))*. The actor specification *a(p)* is *{non-actor}* for non-actor-specific variables and *{arbitrary}* or a subset of *ACTORS* with $\emptyset \neq a(p) \subseteq ACTORS$ for actor-specific belief state variables.

Following this, we adapt the previous definition of belief states as well.

Definition 2' *(belief state).* A belief state *bs* is a finite set of belief state tuples such that for all $p=(v(p),r(p),a(p)) \in bs: (a(p) \subseteq ACTORS \wedge \nexists\, q \neq p \in bs: v(p)=v(q),\ r(p)=r(q) \wedge \nexists\, q \neq p \in bs: v(p)=v(q),\ a(p) \cap a(q) \neq \emptyset) \vee (a(p)=\{non\text{-}actor\} \wedge \nexists\, q \neq p \in bs: v(p)=v(q))$. *BS* is a finite set of belief states.

Definition 2' takes into account that the restrictions (*r(p)*) of the same belief state variable (*v(p)*) may differ for multiple actors (*a(p)*) in a belief state: While Definition 2 states that a belief state contains 'every belief state variable one time at the most', this limitation has been adjusted appropriately in Definition 2'. For instance, in the context of our running example, if *Eric* has already conducted the job interview with the applicant whereas *Bob* and *Danielle* have not (actor-specific) and the contract is not closed yet (non-actor-specific), the according belief state to represent this situation is as follows: *bs = {(applicant interviewed, {true}, {Eric}), (applicant interviewed, {false}, {Bob, Danielle}), (contract closed, {false}, {non-actor})}.*

According to Definition 3, an action consists of its name, its preconditions – which comprise everything an action requires to be applied, including input parameters – and its effects, which denote how the application of an action affects the state of the world, including output parameters. Both preconditions and effects consist of belief state tuples. In light of Definition 1', these preconditions and effects can now contain not only non-actor-specific variables, but also incorporate actor-specific variables. To be more precise, by defining actor-specific preconditions (i.e., belief state tuples with *a(p)⊆ACTORS* within the preconditions) it is possible to limit the applicability of an action to certain actors or to describe actor-specific conditions. To give an example, the restriction of the belief state variable *applicant interviewed* must be *false* for *Eric* and the two research project managers (*Bob* and *Danielle*) in a belief state in order to be able to apply the action *job interview*. To give another example, in one of the processes of the aforementioned insurance company, a project completion report is prepared. This report has to be approved by the client and the internal project manager of the insurance company jointly. However, the project manager previously prepares the report. Hence, in order to apply the action *approve project completion report*, the belief state variable *project report* has to be *prepared* for the project manager and *not approved* for the client. Similarly, actor-specific effects (i.e., belief state tuples with *a(p)⊆ACTORS* within the effects) allow to express actor-specific post-conditions and, if needed, to limit the effects of an action to belief state variables referring to particular actors. Within our running example, this enables to denote that *Eric* and the applicant discuss the *salary requirements* during the job interview and *Eric* decides whether these requirements fit (cf. belief state tuple *(salary requirements fit, {false, true}, {Eric})*).

When preconditions and effects of actions are equivalent for different actors or partnerships (i.e., their belief state variables and restrictions are the same), it is preferable to not model explicit 'personalized' actions (which would differ only in the actor specification and possibly the name) for each actor or partnership. Instead, the amount of required specified actions can be reduced by means of generalization. This reduces the manual effort for the modeler and is more intuitive. For instance, within our example, the action *job interview* basically is the same task whether *Silvia*, the (not mandatory) chair member, participates or not. Hence, it should be modeled as one generic action (cf. Figure 2) instead of several actions which in each case explicitly include all selected actors (i.e., one action in which *Silvia* participates and one action in which she does not).

v(p)	r(p)	a(p)
applicant interviewed	false	Bob Danielle Eric Silvia
contract closed	false	*non-actor*

job interview

	v(p)	r(p)	a(p)
pre(action)	applicant interviewed	false	Bob Danielle Eric
	applicant interviewed	false	*arbitrary*
eff(action)	applicant convincing	false true	*arbitrary*
	applicant interviewed	true	*arbitrary*
	salary requirements fit	false true	Eric

v(p)	r(p)	a(p)
applicant convincing	false true	Bob Danielle Eric Silvia
applicant interviewed	true	Bob Danielle Eric Silvia
salary requirements fit	false true	Eric
contract closed	false	*non-actor*

Figure 2. Illustrating the Action job interview *in the Running Example*

The actor specification of belief state tuples within the preconditions and effects of actions enables us to cope with this challenge: For this purpose, actor-specific belief state tuples with *a(p)={arbitrary}* can be used within the preconditions and effects of actions. This actor specification represents preconditions and effects that concern all actors conducting the action for which no other explicit precondition or effect (by means of a belief state tuple *q* with *v(p)=v(q), a(q)⊆ACTORS*) is specified. To give an example, the aforementioned action *approve project completion report* comprises the effect *{(project report, {approved, not approved}, {arbitrary})}* as the belief state variable *project report* is either *approved* or *not approved* for both actors, the project manager as well as the client.

The definition of applicability (cf. Definition 4) is also adapted in order to take actor-specific variables in belief state tuples into account. For non-actor-specific variables (*a(p)={non-actor}*) in the preconditions of an action the applicability check remains as specified in Definition 4. For belief state variables with *a(p)⊆ACTORS* (e.g., *(applicant interviewed, {false}, {Bob, Danielle, Eric})*), it additionally needs to be checked whether all actors defined in the actor specification are represented in the belief state, the according actor-specific belief state variable exists and the restrictions (here: *{false}*) intersect. For belief state tuples with *a(p)={arbitrary}* (e.g., *(applicant interviewed, {false}, {arbitrary})*), the restriction (here: *{false}*) needs to be checked for all actors in the belief state that are not affected by an according actor-specific precondition (here: *Silvia*). Formally, the extended definition is as follows:

Definition 4' (*applicability*). Let $A(bs) := \bigcup_{(v(p),r(p),a(p))\in bs | a(p)\subseteq ACTORS} a(p)$ be the set of all actors in a belief state bs. An action *action* is *applicable* in *bs* iff the following criteria are fulfilled:

- for all *(v(q),r(q),a(q)) ∈ pre(action)* with *a(q)={non-actor}* there is a *(v(p),r(p),a(p)) ∈ bs* such that *v(p)=v(q)* and *r(p) ∩ r(q)≠ ∅*.
- for all *(v(q),r(q),a(q))∈pre(action)* with *a(q)⊆ACTORS* it holds: For all actors *a∈a(q)* there is a *(v(p),r(p),a(p)) ∈ bs* such that *v(p)=v(q), a∈a(p)* and *r(p) ∩ r(q) ≠ ∅*.
- for all *(v(q),r(q),a(q)) ∈ pre(action)* with *a(q)={arbitrary}* it holds: For all actors *a∈A(bs) \ {a'∈ACTORS|∃(v(x),r(x),a(x)) ∈ pre(action)* such that *v(x)=v(q), a(x)⊆ ACTORS, a' ∈ a(x)}* there is a *(v(p),r(p), a(p)) ∈ bs* such that *v(p)=v(q), a ∈ a(p)* and *r(p) ∩ r(q) ≠ ∅*.

To obtain a planning graph containing information about actors in the belief states, actor-specific effects (i.e., belief state tuples with *a(p)⊆ACTORS* or *a(p)={arbitrary}* in the effects of actions) are applied when performing the state transition. Thus, belief state tuples *p* with *a(p)⊆ACTORS* (e.g., *(salary requirements fit, {false, true}, {Eric})*) within the effects of an action are included in the belief state after the action. Further, for each belief state tuple *p* with *a(p)={arbitrary}* (e.g., *(applicant interviewed, {true}, {arbitrary})*), the belief state tuple

(v(p),r(p),A(bs)\{a'∈ACTORS|∃ (v(x),r(x),a(x))∈eff(action) such that *v(x)=v(p), a(x)⊆ACTORS, a'∈a(x)})* is included in the belief state after the action. This guarantees that the respective effect is applied for each participating actor for which no contrary actor-specific belief state tuple is contained in the effects. By applying these actor-specific effects to the belief state on the left of Figure 2, the belief state tuple *(applicant interviewed, {true}, {Bob, Danielle, Eric, Silvia})* is included in the belief state after the action *job interview* as can be seen on the right of Figure 2.

Each actor involved in a multi-actor process may start at an individual starting point and tends to reach individual goals (Aspect (A1); cf., e.g., Becker et al., 2013; Chiu et al., 2003; Skjoett-Larsen et al., 2003). Therefore, to construct feasible multi-actor process models, we take actor-specific initial states and goal states into account. For instance, in our running example, the goal of the research project managers and the chair member is to hire a new employee who has great professional expertise and integrates well into the team, whereas the personnel officer *Eric* requires the salary expectations of a new employee to fit into the budget. To give another example, in the project completion report process of the insurance company, the individual goal state of the project team is reached as soon as the final project meeting took place but the project manager has to conduct several further actions such as the preparation of the final report. To consider actor-specific initial states (which include actor-specific belief state tuples of only one actor) and goal states, the definition of a planning graph needs to be adapted:

Definition 5' (*planning graph*). A planning graph is a bipartite, directed, finite graph *G=(NODES, EDGES)*, with the set of nodes *NODES* and the set of edges *EDGES*. The set of nodes *NODES* consists of two partitions: The set of action nodes *ACTIONS* and the set of belief state nodes *BS*. Each node *bs∈BS* represents one distinct belief state in the planning graph. The planning graph starts with one to possibly many initial belief states $Init_i \in BS$ (one for each participating actor) and ends with one to possibly many goal belief states $Goal_j \in BS$, in which the goals of at least one actor are fulfilled.

4.2 Consider Cardinalities

We have just outlined how to consider actor-specific information in the planning domain. However, an important characteristic of multi-actor processes has not yet been addressed: Actions may potentially need to be conducted by a certain number of actors (Aspect (A2)). For instance, by means of the previous definition of the action *job interview*, it is only determined that the actor-specific variable *applicant interviewed* needs to have the restriction *false* for *Bob*, *Danielle*, *Eric* and all further actors conducting the action. However, it is not clear whether the action is supposed to be conducted by *Eric* and the two research project managers without an additional chair member or with a certain number of additional chair members. Thus, we extend the common definition of an action (cf. Definition 3) by including the cardinality of the partnership (i.e., set of actors) that has to conduct the action. The cardinality can be defined as a subset of the natural numbers. This definition reduces the amount of specification effort: It

enables to specify actions that can be conducted by partnerships of different sizes (or even by a single actor) in one single action, instead of having to specify each of these possibilities (i.e., for each feasible subset of actors) separately. We adapt Definition 3 as follows:

Definition 3' *(action).* Let *BST* be a finite set of belief state tuples. An action *action* is a quadruple *(name(action), cardinality(action), pre(action), eff(action))* consisting of the action name *name(action)*, the set *cardinality(action)* $\subset \mathbb{N}$ denoting the possible sizes of partnerships required to conduct the action, the set *pre(action)*$\subseteq$*BST* of preconditions of *action* and the set *eff(action)*$\subseteq$*BST* of effects of *action*. It must hold $\min(cardinality(action)) \geq |\bigcup_{(v(p),r(p),a(p))\in pre(action)|a(p)\subseteq ACTORS} a(p)|$.
In the following, *ACTIONS* is a finite set of actions.

In our example, the action *job interview* has to be conducted by at least *Eric* jointly with two research project managers *Bob* and *Danielle*, hence the cardinality of a partnership required to conduct the action *job interview* has to be at least 3. As a (not mandatory) additional chair member may or may not conduct the interview jointly with *Eric* and the two research project managers, the action should be applicable if the cardinality of the partnership is either 3 or 4. Thus, *cardinality(action) = {3,4}* is included in the definition of the action *job interview*.

The cardinality now needs to be considered in the applicability definition (cf. Definition 4') in order to ensure that actions are only applied in a belief state if their cardinality is met.

Definition 4'' (*applicability*). Let $A(bs) := \bigcup_{(v(p),r(p),a(p))\in bs|a(p)\subseteq ACTORS} a(p)$ be the set of all actors in a belief state bs. An action *action* is *applicable* in *bs* iff the following criteria are fulfilled:

- $|A(bs)| \in cardinality(action)$
- for all $(v(q), r(q), a(q)) \in pre(action)$ with $a(q) = \{non\text{-}actor\}$ there is a $(v(p), r(p), a(p)) \in bs$ such that $v(p)=v(q)$ and $r(p) \cap r(q) \neq \emptyset$.
- for all $(v(q), r(q), a(q)) \in pre(action)$ with $a(q) \subseteq ACTORS$ it holds: For all actors $a \in a(q)$ there is a $(v(p), r(p), a(p)) \in bs$ such that $v(p)=v(q)$, $a \in a(q)$ and $r(p) \cap r(q) \neq \emptyset$.
- for all $(v(q), r(q), a(q)) \in pre(action)$ with $a(q) = \{arbitrary\}$ it holds: For all actors $a \in bs \setminus \{a' \in ACTORS | \exists\ (v(x), r(x), a(x)) \in pre(action)$ such that $v(x)=v(q)$, $a(x) \subseteq ACTORS$, $a' \in a(x)\}$ there is a $(v(p), r(p), a(p)) \in bs$ such that $v(p)=v(q)$, $a \in a(p)$ and $r(p) \cap r(q) \neq \emptyset$.

By Definition 4'' – compared to the common Definition 4 – the requirements for the planning of multi-actor process models are addressed by enabling to consider actor-specific belief state variables as well as the number of actors that has to conduct an action jointly.

4.3 Plan Partnerships of Actors

To complete the conceptual foundation for multi-actor process models and thus to fully address

contribution ❶, we describe how to form and disband partnerships in the context of planning multi-actor process models and thus enable the cooperation of multiple actors by explicit actions.

As described in Definition 5' of the planning graph, for each actor $a_i \in ACTORS$ an individual initial state $Init_i \in BS$ with $A(Init_i)=\{a_i\}$ may be specified so that actions are planned from each of these individual starting points. However, in a multi-actor process, it is likely that an action (e.g., the action *job interview* from our running example) can or even needs to be applied jointly by multiple actors. Formally, this may happen due to actor-specific preconditions or cardinality restrictions. Hence, all actors conducting the process (e.g., *Bob, Danielle, Eric* and *Silvia*) need to be taken into account with regard to forming and disbanding partnerships, particularly in order to enable an application of actions that require specific actors and/or a specific number of actors. Partnerships can be seen as a set of actors represented by means of one, joint belief state. Thus, joining multiple, for example single-actor belief states (belief states with $|A(bs)|=1$), into one multi-actor belief state (a belief state with $|A(bs)|>1$) is required. Within our running example, the individual initial states of *Bob, Danielle, Eric* and *Silvia* need to be joined in order to construct a joint belief state, so that the action *job interview* is applicable in this joint belief state. Additionally, in a multi-actor process, the situation can arise that only a subset of actors in an existing partnership can conduct an action jointly (e.g., due to an upper bound in the cardinality). We thus need to be able to disband a partnership and to split a belief state *bs* into a set of 'sub' belief states, each containing a subset of actors in *bs*. To enable the automated construction of a complete process model in which all appropriate partnerships are considered and hence to enable supporting individual actors when and with whom to cooperate, we address these issues by automatically identifying possibilities for forming and disbanding partnerships. We will construct respective *join actions* and *split actions* by means of an algorithm (cf. next subsection) and in compliance with the planning domain: These join actions and split actions are defined in terms of name, cardinality, preconditions and effects just like regular actions (cf. Definition 3'), and the joint/split states result from the application of the join/split actions and their effects on the preceding states. Thus, by planning these explicit actions we incorporate the formation and disbandment of appropriate partnerships in the control flow of the constructed process models (cf. Aspect (A3)) while ensuring compatibility with existing single-actor planning approaches.

The preconditions of a join/split action *action* are determined based on the according preceding belief state *bs* so that *action* is applicable in *bs*. Their cardinality is set to $|A(bs)|$. The effects of a join action are constructed so that all according belief state tuples for actors in the other (to be joined) belief states are added (i.e., created) by means of the effects. For instance, the effects of the join action expressing that *Eric* cooperates with *Bob, Danielle* and *Silvia* are defined as *{(applicant interviewed, {false}, {Bob, Danielle, Silvia})}* so that the joint state bs_{joined} *= {(applicant interviewed, {false}, {Eric, Bob, Danielle, Silvia})}* results from its application in the belief state bs_{Eric} *= {(applicant interviewed, {false}, {Eric})}*. The effects of a split action are

specified contrarily, removing actor-specific belief state tuples of actors that are no longer part of the partnership after the disbandment.

Further, we aim at constructing join/split actions only when appropriate (i.e., feasible and necessary). Thus, we need to determine which belief states are appropriate for being *joined* and which belief states need to be *split*.

In a first step, we need to ensure that forming a partnership (i.e., joining a set of preceding belief states $\{bs_1, \ldots, bs_n\}$) is feasible and does not lead to logical contradictions. When forming a partnership, the status of each actor participating in the partnership must not be represented by more than one of the belief states. Hence, before constructing a join action, we require that

i. $A(bs_i) \cap A(bs_j) = \emptyset$ for all $i, j \in \{1, \ldots, n\}$ such that $i \neq j$.

Further, it is required that – before forming a partnership – multiple belief states representing different actors are not contradictory with respect to non-actor specific belief state variables. For instance, within our running example, joining two belief states with the non-actor-specific belief state variable *contract closed* being *{true}* in one of the belief states and being *{false}* in the other belief state would lead to a contradiction:

ii. For each $bs_i, i \in \{1, \ldots, n\}$: for each $(v(p), r(p), a(p)) \in bs_i$ with *a(p)={non-actor}*: for each $bs_j, j \neq i$: there is a $(v(p_j), r(p_j), a(p_j)) \in bs_j$ with $v(p_j) = v(p)$ such that $r(p_1) \cap \ldots \cap r(p_n) \cap r(p) \neq \emptyset$.

We further want to avoid the construction of unnecessary join actions. We thus require that the formed partnership is able to conduct at least one action. We ensure this with the following criterion iii. that has to be met by the joint belief state *bs* before constructing the according join action:

iii. In *bs*, at least one action is applicable (cf. Definition 4'').

In a second step, when disbanding a partnership, we need to ensure that the resulting process model does not contain logical contradictions and thus a belief state *bs* with the partnership *A(bs)* can be *split* into the belief states $bs_1, \ldots, bs_n$ with the partnerships $A(bs_1), \ldots, A(bs_n)$ by split actions only if the following criteria i. and ii. are fulfilled. These criteria are the counterparts to the previously defined criteria for forming a partnership. First, after disbanding a partnership, each actor may be contained in exactly one state after disbanding the partnership (cf. i.). This again results from the fact that the current status of an actor is always represented by one single belief state. Further, we need to ensure that each and every actor contained in the to-be-disbanded partnership is contained in a belief state after splitting the belief state (cf. ii.):

i. $A(bs_i) \cap A(bs_j) = \emptyset$ for all $i, j \in \{1, \ldots, n\}$ such that $i \neq j$.

ii. $\cup_i A(bs_i) = A(bs)$

Additionally, we again ensure that at least one action is applicable in each belief state after splitting to avoid the construction of unnecessary split actions (cf. iii.). This, together with ii., is required as otherwise, actors would possibly not be able to reach their individual goal state(s):

iii. In each belief state bs_i, at least one action is applicable (cf. Definition 4'').

For each set of belief states that meets the criteria for being joined, respective each single belief state that meets the criteria for being split, we construct the according join actions resp. split actions by means of an algorithm, which is presented in the following subsection.

4.4 Algorithm

Existing single-actor planning approaches (e.g., Bertoli et al., 2001; Bertoli et al., 2006; Heinrich et al., 2012; Heinrich and Schön, 2015; Henneberger et al., 2008; Hoffmann et al., 2009) construct planning graphs by means of a forward search that iteratively ***1)*** retrieves which actions are applicable in a belief state and ***2)*** generates the next belief state for each of these actions. We adopt these approaches, consisting of the major phases *identification of applicable actions* and *retrieval of next belief state*, but extend them for planning multi-actor process models (cf. contribution ❷). Our algorithm is presented in form of a pseudocode (see Appendix A) and outlined in a textual description.

The algorithm works iteratively, starting with the initial belief states. For a belief state, it ***1a)*** checks which actions are applicable (cf. Definition 4''; line 4 of Listing 2 in Appendix A) in the considered belief state. Further, actions that ***1b)*** can be conducted by a subset of the actors represented in the belief state (line 6 of Listing 2) and actions that ***1c)*** can possibly be conducted by a partnership that needs to be formed (line 9 of Listing 2) are identified.

For each action identified as applicable (cf. step *1a)*), ***2a)*** the belief state resulting from the application of the action is constructed and planned (line 5 of Listing 2). If an action can be conducted by a subset of the actors (cf. step *1b)*; line 6 of Listing 2 and SUB disband; cf. Listing 5), ***2b)*** according split actions and the subsequent belief states are constructed automatically (line 11 in Listing 5), based on the belief state and the information which (smaller partnership of) actor(s) could conduct the action. If a partnership can possibly be formed to conduct the action (cf. step *1c)*; line 9 in Listing 2 and SUB join; cf. Listing 4), ***2c)*** the action together with the currently considered preceding belief state is saved as *potentially suitable for cooperation* (line 2 in Listing 4).

Further, in step ***2d)*** such an identified possibility for cooperation is matched with other combinations of belief states with the considered action already identified in preceding iterations (line 3 in Listing 4). If thereby an action is identified as applicable by a partnership of actors represented in different belief states (and thus all criteria i. to iii. are fulfilled), ***2e)*** the algorithm subsequently performs an automated planning of join actions (line 7 in Listing 4). These join actions create a joint belief state by means of a regular state transition. They thus enable a joint conduction of the action (in the joint belief state) by actors that formerly were represented in

their own individual belief states or cooperated in smaller partnerships. After ***2e)***, the next iteration step starts.

To sum up the proposed approach and to illustrate the resulting planning graph, Figure 3 shows an excerpt of our running example. Each conducting actor starts with an individual initial state (cf. Definition 5'), denoted by means of a square, tagged with IS and the according name of the actor, at the leftmost area in Figure 3. The actors need to form a partnership in order to conduct the action *job interview* jointly. This is achieved by join actions labelled with 'cooperate' (actions are denoted by rounded rectangles). The partnership is represented by the joint belief state (tabular representation of belief state tuples) in the left area of the detailed excerpt framed by the dashed line. Then, the action *job interview* – our focus in Figure 3 – is planned for *Bob*, *Danielle* and *Eric* (here, the not mandatory chair member *Silvia* participates as well). It leads to the following belief state at the right of the detailed excerpt. Subsequently, the applicant will be engaged or rejected (see actions at the top area of Figure 3).

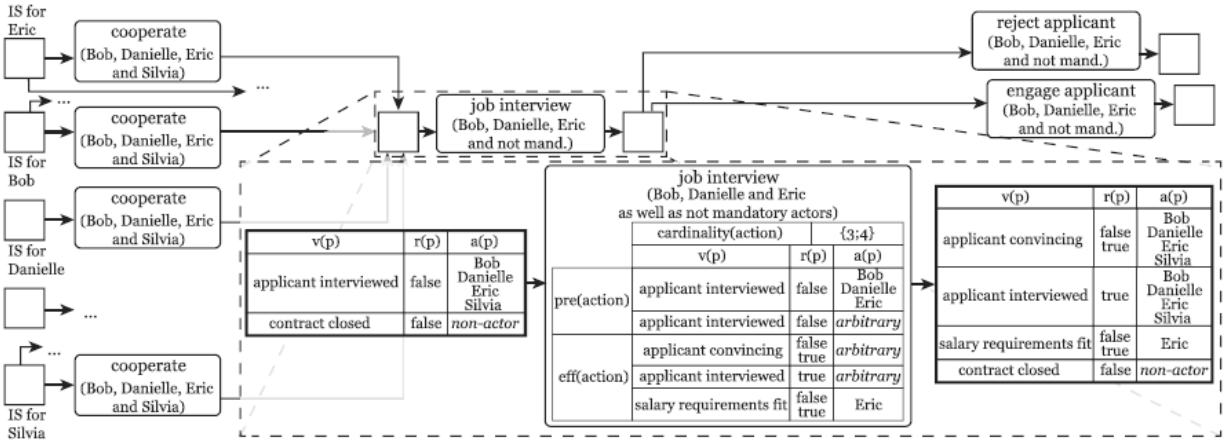

Figure 3. Excerpt of the Planning Graph of the Running Example

5 Evaluation

In order to provide a 'proof of the solution' (Bertrand and Fransoo, 2002), we evaluated the validity (E1) of our approach. Furthermore, as proposed by Meredith et al. (1989), we evaluated the technical and practical feasibility (E2) as well as the effectiveness of our approach (E3) by means of a prototypical implementation and its application in real-world scenarios.

5.1 Assessment of the Validity (E1)

To assess the validity (E1), we conducted a mathematical evaluation of our approach by proving the key properties termination, correctness (i.e., all planned paths are feasible) and completeness (i.e., all feasible paths from an initial state to a goal state are planned). Due to length restrictions, we refer to Appendix B for the proofs. The proofs show that our algorithm terminates and the multi-actor process models constructed by our approach in an automated manner

are indeed correct and complete (cf. contribution ❷).

5.2 Assessment of the Technical and Practical Feasibility (E2)

When evaluating the technical and practical feasibility of our approach (E2), we examined these criteria regarding the algorithm and the underlying conceptual foundation by analyzing the following three evaluation questions:

(E2.1) *Can the approach be instantiated in a prototypical implementation?*

(E2.2) *Is it possible to apply the approach to real-world scenarios and how can the necessary input data (i.e., the specification of actors, actions, initial states and conditions for goal states) be obtained?*

(E2.3) *What are the results of these applications in terms of correctness of the constructed multi-actor process models? What are the key properties of these models and how long does their automated planning take?*

With respect to (E2.1), a Java implementation of a single-actor process planning algorithm (cf. Bertoli et al., 2006; Heinrich and Schön, 2015) served as a basis for our work. We extended this implementation to incorporate the presented algorithm that enables the automated construction of multi-actor process models (see Appendix A for the pseudocode of the algorithm). Actors, actions, initial states and goal states can be imported into the prototype by means of XML files. We ensured the validity of our prototype by means of structured tests using the JUnit framework. Here we carried out extreme value tests, unit tests and regression tests (i.e., validation that single-actor process models could still be planned correctly). Further, persons other than the programmers validated the source code via a structured walkthrough. At the end of the test phase, the implementation did not show any errors, supporting the technical feasibility of our approach and providing "proof by construction".

In regard to (E2.2) we analyzed whether it is possible to apply the approach to real-world scenarios (i.e., the scenarios *Human Resources, Product Manufacturing, Healthcare* and five further scenarios from the European financial services providers discussed in the introduction) using our prototypical implementation. In particular, we analyzed whether and in which way it is possible to obtain the necessary input data to apply the approach. Our study showed that the necessary input data could be obtained in different ways. On the one hand, we, for instance, revised and extended existing single-actor planning specifications (i.e., input about participating actors such as employees or departments) to enable the planning of multi-actor process models. On the other hand, we have been able to formalize the informal information provided by domain experts so that our approach could be applied. This is of particular interest for process modeling projects in practice where domain experts and business analysts often closely work together to construct process models. In Table 2 we give details about how the necessary input data was obtained, similar to Siha and Saad (2008). Due to length restrictions, we concentrate on the scenarios *Human Resources, Product Manufacturing* and *Healthcare*. However,

we also applied our approach to five further scenarios from European financial services providers, where the data provided by these companies could successfully be used as input data.

Scenario	Human Resources	Product Manufacturing	Healthcare
Description of the scenario	Hiring at a university, starting from an application up to the processing of the hiring decision	Handling of incoming orders at a manufacturing company up to the shipping of the ordered products	Basic course of surgery in a hospital, starting from an preliminary talk up to the patient leaving the hospital
Context of the scenario	Real process, conducted at a chair of a university	Real process as conducted at a manufacturing company for electrical devices	Simplified version of real process as conducted at a hospital
Context of the data provisioning	Experimental case study with chair; aim of project: visualisation/documentation of conducted process	Experimental case study with the head of production department; aim of project: support of process improvement; three interviews	Experimental case study with an experienced intensive care surgical nurse of a hospital; aim: evaluation of the presented approach; two interviews
Basis of provided information	Existing scenario specified (in terms of XML files) for single-actor planning algorithm	Existing scenario specified (in terms of XML files) for single-actor planning algorithm	Informal information provided by intensive care surgical nurse
Formalization and refinement of provided information	Retrieval in two steps: *First step:* - Determination of responsible actors from action names of single-actor specification (e.g., *research project manager checks CV* ⇨ actor *research project manager*) - Retrieval of actions and initial states / goal states based on single-actor specification	Retrieval in two steps: *First step:* - Determination of responsible actors from action names of single-actor specification (e.g., *warehouse checks stock* ⇨ actor *warehouse*) - Retrieval of actions and initial states / goal states based on single-actor specification	Retrieval of formal and informal information based on two interviews: *First interview:* - Required staff for a typical surgery (participating actors) - His tasks during the surgery

Table 2. Evaluation of our Approach with regard to (E2.2)

	Second step: - Revision of actions regarding required cooperation of the conducting actors based on transcripts of an interview with according employees (e.g., the definition of the action *job interview* was extended as it, according the transcripts, has to be conducted jointly by multiple actors) - Preparation of XML specifications	*Second step:* - Revision of actions regarding required cooperation of the conducting actors based on interviews with the head of production department (e.g.: the definition of the action *hand over shipment notification* was extended as it, according the head of production department, has to be conducted jointly by the sales department and the warehouse department) - Preparation of XML specifications	*Second interview (after surgical nurse consulted colleagues):* - Actions conducted jointly with other actors - Individual initial and goal states of all actors; (e.g., the surgeon is no longer needed – formally: his/her goal state is met - as soon as the patient is transferred to recovery; ward nurse's goal is met when patient is in his/her room again) *After second interview:* - Refinement and formalization of the provided informal information in terms of XML specifications
Actors involved	Applicant / two research project managers / personnel officer / chair member / gender representative	Warehouse department / production department / sales department	Surgeon / anaesthesiologist / surgical nurse / ward nurse / patient

Table 2. Evaluation of our Approach with regard to (E2.2) (continued)

Scenario	Human Resources	Product Manufacturing	Healthcare
(A1) all individual starting points and goals of conducting actors are appropriately contained	Applicant, gender representative, personnel officer and chair member start from individual actor-specific initial states and possess individual actor-specific goal states; research project managers start at common actor-specific initial state and possess a common actor-specific goal state; example: process of personnel officer starts with incoming application and ends when information about applicant being hired or not is sent.	Each of the three actors starts from an actor-specific initial state and possesses at least one actor-specific goal state; example: process of sales department starts with an incoming order and ends when the order is declined or shipment information has been sent to customer.	Each of the five actors starts from an individual actor-specific initial state and possesses an actor-specific goal state; example: process of ward nurse starts with the patient being in in his/her room before transferring him/her to anteroom of surgery and ends when the patient is back in his/her room after the surgery.
	All six actors as well as their according initial states and goal states are correctly included in the resulting process model.	**All three participating departments (represented by actors) as well as their according initial states and goal states are correctly included in the resulting multi-actor process model.**	**All five actors as well as their according initial states and goal states are correctly included in the resulting process model.**
(A2) all partnerships conducting actions jointly are appropriately contained	During the process, different partnerships are needed to conduct actions; example: both research project managers as well as (not mandatory) chair member check application documents.	Partnerships consisting of particular actors are needed to conduct four actions; example: warehouse department and production department have to jointly conduct *create production requirements*; based on ordered amount, available warehouse stock and already dispatched production output, amount of required additional production output is retrieved.	During the process, the actors need to form different partnerships; example: surgeon and patient conduct a common *preliminary talk*.

Table 3. Evaluation of our Approach with regard to (A1)-(A3)

	15 actions are conducted jointly during the process; four distinct partnerships (consisting of up to four actors) that conduct these actions are correctly included in the resulting process model.	**Four actions are conducted in respective partnerships (max size: two actors); all required partnerships are correctly included in the resulting process model.**	**13 actions are conducted in respective partnerships (max size: four actors); all required partnerships are correctly included in the resulting process model.**
(A3) all required join actions and split actions are appropriately contained	To form resp. disband the partnerships required for the process (cf. Aspect (A2)), join actions resp. split actions are constructed by our approach in an automated manner. For instance, three join actions form the partnership that allows conducting the action *check application documents*. Two of these join actions incorporate the cooperation of the respective research project managers with each other as well as the chair member while the third join action incorporates the cooperation of the chair member with both research project managers.	To form resp. disband the partnerships required for the process (cf. Aspect (A2)), join actions resp. split actions are constructed by our approach in an automated manner. For instance, two join actions form the partnership that allows conducting the action *create production requirements*. One of the join actions incorporates the cooperation of the warehouse department with the production department and the other join action inversely incorporates the cooperation of the production department with the warehouse department.	To form resp. disband the partnerships required for the process (cf. Aspect (A2)), join actions resp. split actions are constructed by our approach in an automated manner. For instance, two join actions form the partnership that allows conducting the action *preliminary talk*. One of the join actions incorporates the cooperation of the patient with the surgeon and the other join action inversely incorporates the cooperation of the surgeon with the patient.
	The resulting multi-actor process model contains all 23 necessary join actions and all 23 necessary split actions to enable a correct conduction of the process.	**The resulting multi-actor process model contains all eight necessary join actions and all six necessary split actions to enable a correct conduction of the process.**	**The resulting multi-actor process model contains all 156 necessary join actions and all 24 necessary split actions to enable a correct conduction of the process.**

Table 3. Evaluation of our Approach with regard to (A1)-(A3) (continued)

In regard to (E2.3), we applied our approach to these scenarios and aimed at evaluating in how far our approach is suitable for providing a conceptual foundation (cf. contribution ❶) for multi-actor process models in real-world scenarios and to which extent the results of the approach correspond to the actually conducted processes in the scenarios. Thus, we evaluated in detail whether all Aspects (A1) to (A3) of contribution ❶ are appropriately taken into account in the resulting process models. Precisely, we evaluated whether (A1) all individual starting points and goals of conducting actors, (A2) all partnerships conducting actions jointly as well as (A3) all required join actions and split actions were appropriately contained in the real-world scenarios. Similar to the presentation in Siha and Saad (2008), we discuss the evaluation of our approach with regard to (E2.3) in Table 3, where we again focus on the three scenarios *Human Resources*, *Product Manufacturing* and *Healthcare* (comparable findings could be provided for the other evaluated scenarios as well).

In the application to these real-world scenarios, all necessary individual starting points and goals of actors as well as partnerships conducting actions jointly were represented and all join and split actions were constructed correctly according to the provided input. We evaluated this by a structured walkthrough of the constructed process models and by examining whether each applicable action as well as each necessary join and split action was planned and whether all planned actions were correct and actually necessary.

We further examined the key properties of the multi-actor processes and the according multi-actor process models resulting from applying our approach. As seen in Table 4, we first determined the number of actors conducting the processes as well as the number of belief states, join actions, split actions, actions conducted in a partnership and actions in total in the multi-actor process models. Additionally, we identified the number of partnerships as well as the minimum and maximum number of actors cooperating in a partnership for each process. Lastly, we determined the required runtime for planning the multi-actor process models (executed on an Intel Core i7-2640M, 2.80 GHz, Windows 8.1 64 bit, Kernel Version 6.3.9600, Java 8). The process models are of small to large size, containing between 20 and 212 actions in total. This is also reflected by the number of actors, which ranges from two to eight actors that form a maximum of up to seven different partnerships. These partnerships conduct between four and 19 actions throughout the respective processes and consist of two up to four actors. Our approach was capable of constructing the multi-actor process models regardless of their size and complexity. Overall, the required runtime for planning multi-actor process models comprising a significant number of actors, partnerships as well as join and split actions still was below four seconds, which supports the practical feasibility of our approach.

To sum up, our approach was prototypically implemented, provided a suitable conceptual foundation for the resulting small, medium-sized and large multi-actor process models in several real-world scenarios and their automated planning could be completed in appropriate time. These results support the technical and practical feasibility of our approach.

Scenario context	University	Manufacturing company	German regional hospital	European insurance company			European bank	
Scenario name	**Human Resources**	**Product Manufacturing**	**Healthcare**	**Allocation of project resources**	**Preparation of decision template for project application**	**Project closure**	**Transfer of investment deposit**	**Combination of multiple investment accounts**
Number of actors in process model	6	3	5	8	2	7	7	6
Number of belief states in process model	67	27	126	76	15	43	24	35
Number of join actions in process model	23	8	156	30	6	23	9	18
Number of split actions in process model	23	6	24	30	6	14	3	10

Table 4. Key Properties of the constructed Multi-Actor Process Models and Runtime for planning them

Number of actions conducted in a partnership in process model	15	4	13	19	4	11	11	9
Number of actions (in total) in process model	75	26	212	92	20	51	25	42
Number of partnerships (min. size / max. size) in process model	4 (2/4)	2 (2/2)	6 (2/4)	7 (2/3)	1 (2/2)	6 (2/3)	4 (2/4)	5 (2/2)
Runtime for planning the multi-actor process model	3.427 s	0.021 s	0.161 s	0.128 s	0.004 s	0.032 s	0.032 s	0.014 s

Table 4. Key Properties of the constructed Multi-Actor Process Models and Runtime for planning them (continued)

5.3 Assessment of Effectiveness (E3)

In order to assess the effectiveness of our approach, we discuss the following evaluation question:

(E3) Are the constructed multi-actor process models feasible according to the assessment of a practitioner in an experimental setting?

To evaluate the effectiveness in real-world scenarios, we applied our approach in an experiment (cf. Meredith et al., 1989). Due to length restrictions, we focus on the real-world scenario *Healthcare* by examining a surgery process and present our findings within this scenario.

The environment of this experiment as well as its results are presented in Table 5. In this setting, we aimed to evaluate whether our approach constructs feasible multi-actor process models that appropriately reflect processes as conducted in reality in regard to the assessment of a practitioner. Similar to process modeling in a real business environment, an experienced intensive care surgical nurse (domain expert) provided us with detailed information about the basic course of a surgery and the involved actors in two interviews (cf. also the corresponding description in Table 2). As the surgical nurse was not familiar with process modeling we refined and formalized the informal information he gave us and hence specified the actors, preconditions, cardinalities and effects of actions in terms of the aforementioned XML files. We thereafter were able to plan a multi-actor process model that comprised 156 join actions and 24 split actions (see Table 4) by means of our prototypical implementation.

We then asked him whether the constructed multi-actor process model appropriately reflects the starting point and goals of the surgical nurse in the process (Aspect (A1)), partnerships including the surgical nurse conducting actions jointly (Aspect (A2)) as well as the join actions and split actions in which the surgical nurse participates (Aspect (A3)). Table 5 describes the assessment of the surgical nurse regarding Aspects (A1) to (A3) in detail (structured in a similar way as Siha and Saad (2008) present their findings).

Scenario	**Healthcare**
Description of the scenario	Basic course of surgery in German hospital
Way of assessing the model	*Further (third) interview*: Step-by-step discussion of the model with the surgical nurse, focusing (primarily, not exclusively) on the actions he has to perform (walkthrough); brief description of the model so that he could understand it (as he was not familiar with process modeling notations); verbal discussion of Aspects (A1) to (A3); focus on the sequence of actions/tasks as well as the partnerships he joined throughout the process

Results (according to the assessment of the surgical nurse)	
(A1) individual starting points and goals of conducting actors correspond to those in the actually conducted process	We described the meaning of the belief state variables contained in the initial states and goal states. For instance: Discussion of the belief state tuples of the initial state in which his process starts; surgical nurse stated that he, as correctly represented in the initial state, starts in regular clothes in the anteroom. He further stated that his process ends when the paperwork is done after the actual surgery, which is also correctly represented in the according goal state.
	The surgical nurse confirmed that, in his view, initial state and goal state in the multi-actor process model accurately reflect the respective states compared to the conduction of the process in reality.
(A2) partnerships conducting actions jointly correspond to those in the actually conducted process	Discussion of the partnerships of the multi-actor process model in which the surgical nurse participates according to the model; in particular: clarification whether he actually participates in these partnerships in a real surgery. He agreed that, for instance, he brings the patient to and from the surgery room (formally: joins a partnership with the patient) and finishes the paperwork without any actor; he further stated that – at least spontaneously – he could not think of a partnership occurring in reality but not represented in the process model.
	The surgical nurse confirmed that, in his view, partnerships are appropriately contained in the multi-actor process model and reflect the partnerships as formed during the process in reality.
(A3) join actions and split actions correspond to those in the actually conducted process	Additional clarification about the meaning of the split actions was necessary; we elaborated that they tell an actor to "leave a partnership" and to continue with his/her individual tasks or with joining a different partnership with other actors; thereafter, he confirmed that, for instance, the partnership conducting the surgery is correctly split; anesthesiologist and surgeon leave the surgery room.
	The surgical nurse confirmed that, in his view, join and split actions are appropriately contained in the multi-actor process model and reflect the respective actions during the process in reality.

Table 5. Evaluation of our Approach with regard to (E3)

To sum up, the experimental evaluation together with a practitioner supported the effectiveness of our approach to construct feasible multi-actor process models since an actor-specific initial state and actor-specific goal states, partnerships as well as join actions and split actions were considered as valid by the practitioner.

To conclude, the analysis of the evaluation questions supports the validity, the technical and practical feasibility and the effectiveness of the presented approach. Table 6 summarizes the results.

Evaluation Question	Result
(E1) Does the approach terminate and provide correct and complete multi-actor process models?	A mathematical evaluation of the approach proves that these criteria hold.
(E2.1) Can the algorithm be instantiated in a prototypical implementation?	The algorithm was implemented and successfully integrated into a prototype for the automated planning of process models.
(E2.2) Is it possible to apply the algorithm to real-world scenarios and how can the necessary input data (i.e., the specification of actors, actions, initial states and conditions for goal states) be obtained?	The algorithm was applied to several real-world scenarios. The necessary input data could, for instance, be obtained by analyzing and refining existing specifications for single-actor process models or by interviewing a participant of the process and formalizing the provided data in terms of XML files.
(E2.3) What are the results of these applications in terms of correctness of the constructed multi-actor process models? What are the key properties of the constructed multi-actor process models and how long does it take to construct these models?	Multi-actor process models were constructed for each of the real-world scenarios. The Aspects (A1) to (A3) were fulfilled in each case. The constructed multi-actor process models have been of small to large size (regarding the number of actions, conducting actors and partnerships). The runtime for planning such multi-actor process models comprising a significant number of join and split actions was below four seconds.
(E3) Are the constructed multi-actor process models feasible according to the assessment of a practitioner in an experimental setting?	The practitioner confirmed that (A1) initial states and goal states of the multi-actor process model reflected the respective states in reality; (A2) all partnerships contained in the multi-actor process model constructed by the approach corresponded to those formed in the actually conducted process; (A3) the join and split actions contained in the multi-actor process model as well as the model itself were feasible.

Table 6. Results with regard to all Evaluation Questions

6 Conclusion, Limitations and Future Work

In this paper, we presented an approach for the automated planning of multi-actor process models (cf. contribution ❷) based on a conceptual foundation (cf. contribution ❶). We described how to extend a common planning domain in the literature to enable taking actor-specific information and individual starting points and goals into account (Aspect (A1)). Our approach can further cope with cardinalities of partnerships (i.e., sets of actors) required to conduct an action (Aspect (A2)). Moreover, we outlined how to construct join and split actions in an automated manner. These actions incorporate the cooperation of multiple actors in the control flow of process models and hence support individual actors by determining explicitly at which steps in a process they can or need to cooperate in partnerships to achieve their individual goals (Aspect (A3)). As our approach extends existing single-actor planning approaches, compatibility with prevalent works is supported. Our approach is evaluated by means of mathematical proofs of its key properties, a prototypical implementation, the application to real-world scenarios, a detailed analysis of the constructed multi-actor process models regarding Aspects (A1) to (A3), runtime analyses and the assessment of a practitioner in an experimental real-world scenario.

Our work addresses an important sub problem of the research field automated planning of process models, namely the automated planning of multi-actor process models. This issue has not been addressed so far and hence we believe that our work significantly increases the scope of that research field. Furthermore, it contributes to the general research field of business process modeling by presenting a new approach to represent multi-actor processes that comprise partnerships conducting parts of processes jointly. Existing modeling approaches and notations such as swimlanes have several shortcomings, resulting in ‘messy and difficult to understand’ process models (Pulgar and Bastarrica, 2017). Hence, we include the cooperation of actors in the control flow of process models by constructing explicit actions determining where in the process to form and to disband partnerships. Additionally, we address a relevant problem in practice as multi-actor process models are widespread in today’s business world. For instance, we strongly supported the analysis of about 600 core processes of two European financial services providers. In this context, over 60% of the analyzed processes of the insurance company comprise partnerships of three or more actors. The proposed approach enables practitioners to represent multi-actor process models and to denote actions that have to be performed by a partnership of multiple actors in contrast to existing approaches. Lastly, as runtimes for the automated planning of multi-actor process models were short, the proposed approach enables companies to construct multi-actor process models in appropriate time and thus to stay flexible and competitive.

However, there are some limitations of our work which have to be addressed in future research. First, to increase the acceptance of our approach in an industrial setting and hence to enable process modelers or even domain experts without expertise in process modeling to construct multi-actor process models, the prototypical implementation needs to be extended in terms of a graphical user interface. There exists a graphical user interface for the single-actor process

planning approach that served as a basis for our prototypical implementation. This enables modelers to specify actions, including preconditions and effects, as well as initial and goal states of single-actor processes. However, this graphical user interface needs to be extended to allow the definition of multi-actor process models, comprising partnerships of actors, with individual initial states and goal states as well as actions that have to be conducted by these partnerships.

Second, process models can be hard to grasp for humans, especially when they represent complex processes (e.g., many actions and control flow structures) that are conducted by a large number of actors. Thus, future research should strive to alleviate this issue. A promising idea could be to provide an "actor-specific view" of the multi-actor process models constructed by our approach by focusing on and representing only actions and belief states that are relevant for the conduction of a specific actor.

Third, multi-actor processes in practice can vary considerably with regard to participating actors, size, goals and additional criteria. While the application of our approach in multiple real-world scenarios showed its feasibility and effectiveness, an application in further contexts could provide a more thorough verification of its practical feasibility.

Fourth, when applying the approach in real-world scenarios, "noisy" preconditions or effects of actions may occur (e.g., an interviewee is uncertain to specify starting from what order amount a control by three different actors is necessary regarding regulatory compliance) and influence the multi-actor process model resulting from planning. To address this issue, multiple plannings with different preconditions and/or effects of respective actions could be conducted. Based on this, it can be evaluated whether and to what extent (i.e., which actions) the resulting process model is influenced by the "noise" at all. This supports the determination of a feasible process model under such circumstances.

Fifth, in this paper we presented how preconditions and effects of actions can be specified on a per-actor basis. The processes we analyzed together with European financial services providers oftentimes contain actors representing departments consisting of multiple individuals. However, planning process models on the basis of individuals may sometimes be preferable. For instance, this may be beneficial in the case of actions that require a particular number of actors of a department. For such a more fine-grained planning, it would be promising to allow a role-based specification (as it is possible, for instance, in security related topics like access control) of preconditions and effects. Following this, a department consisting of multiple persons could be represented by a role and the individual persons could be specified by role-based preconditions and effects. In future, the presented approach can be enhanced to incorporate such role-based specifications by subsuming actors as well as action-specifications under roles and requiring a subset of the actors of each role for role-based preconditions.

7 References

Becker, J., D. Beverungen, R. Knackstedt, M. Matzner, O. Müller and J. Pöppelbuß (2013). "Designing Interaction Routines in Service Networks" *Scandinavian Journal of Information Systems (SJIS)* 25 (1), 37–68.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2001). "Planning in nondeterministic domains under partial observability via symbolic model checking" *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)* 1, 473–478.

Bertoli, P., A. Cimatti, M. Roveri and P. Traverso (2006). "Strong planning under partial observability" *Artificial Intelligence* 170 (4–5), 337–384.

Bertrand, J. W. M. and J. C. Fransoo (2002). "Operations management research methodologies using quantitative modeling" *International Journal of Operations & Production Management (IJOPM)* 22 (2), 241–264.

Boukhedouma, S., Z. Alimazighi and M. Oussalah (2017). "Adaptation and Evolution Frameworks for Service Based Inter-Organizational Workflows" *International Journal of E-Business Research* 13 (2), 28–57.

Cabanillas, C., M. Resinas, J. Mendling and A. Ruiz-Cortés (2015). "Automated team selection and compliance checking in business processes". In: *Proceedings of the International Conference on Software and System Process (ICSSP 2015)*, pp. 42–51.

Chen, Q. and M. Hsu (2001). "Inter-enterprise collaborative business process management". In: *Proceedings of the International Conference on Data Engineering (ICDE 2001)*, pp. 253–260.

Chinosi, M. and A. Trombetta (2012). "BPMN: An introduction to the standard" *Computer Standards & Interfaces* 34 (1), 124–134.

Chiu, D.W., K. Karlapalem and Q. Li (2003). "Views for Inter-organization Workflow in an E-commerce Environment". In *Semantic Issues in E-Commerce Systems*, pp. 137–151: Springer US.

Chouhan, S. S. and R. Niyogi (2017). "MAPJA. Multi-agent planning with joint actions" *Applied Intelligence* 14 (4), 105.

Crosby, M., M. Rovatsos and R. P. A. Petrick (2013). "Automated Agent Decomposition for Classical Planning". In: *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS 2013)*.

Davenport, T. H. and J. Short (1990). *Information technology and business process redesign:* Taylor & Francis US.

de Weerdt, M. and B. Clement (2009). "Introduction to planning in multiagent systems" *Multiagent and Grid Systems - Planning in Multiagent Systems* 5 (4), 345–355.

Dimopoulos, Y. and P. Moraitis (2006). "Multi-Agent Coordination and Cooperation through Classical Planning". In: *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 398–402.

Ding, Z., Y. Sun, J. Liu, M. Pan and J. Liu (2015). "A genetic algorithm based approach to transactional and QoS-aware service selection" *Enterprise Information Systems*, 1–20.

Dobson, S., S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt and F. Zambonelli (2006). "A survey of autonomic communications" *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 1 (2), 223–259.

Ephrati, E. and J. S. Rosenschein (1994). "Divide and conquer in multi-agent planning". In *Proceedings of the 12th National Conference on Artificial Intelligence*, p. 80.

Fahland, D., C. Favre, J. Koehler, N. Lohmann, H. Völzer and K. Wolf (2011). "Analysis on demand. Instantaneous soundness checking of industrial business process models" *Data & Knowledge Engineering (DKE)* 70 (5), 448–466.

Falou, M. E., M. Bouzid, A.-I. Mouaddib and T. Vidal (2009). "Automated Web Service Composition: A Decentralised Multi-agent Approach". In: *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009)*.

Fikes, R. E. and N. J. Nilsson (1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". In: *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI 1971)*.

Fleischmann, A., U. Kannengiesser, W. Schmidt and C. Stary (2013). "Subject-Oriented Modeling and Execution of Multi-agent Business Processes". In: *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013)*.

Forstner, E., N. Kamprath and M. Röglinger (2014). "Capability development with process maturity models – Decision framework and economic analysis" *Journal of Decision Systems (JDS)* 23 (2), 127–150.

Ghallab, M., D. S. Nau and P. Traverso (2004). *Automated Planning: Theory & Practice:* Morgan Kaufmann.

Ghallab, M., D. S. Nau and P. Traverso (2016). *Automated Planning and Acting:* Cambridge University Press.

Ghrab, S., I. Saad, G. Kassel and F. Gargouri (2017). "A Core Ontology of Know-How and Knowing-That for improving knowledge sharing and decision making in the digital age" *Journal of Decision Systems (JDS)* 26 (2), 138–151.

Grefen, P., K. Aberer, Y. Hoffner and H. Ludwig (2000). "CrossFlow: cross-organizational workflow management in dynamic virtual enterprises" *International Journal of Computer Systems Science & Engineering (CSSE)* 15 (5), 277–290.

Havur, G., C. Cabanillas, J. Mendling and A. Polleres (2015). "Automated Resource Allocation in Business Processes with Answer Set Programming". In: *Proceedings of the 11th International Workshop on Business Process Intelligence*.

Havur, G., C. Cabanillas, J. Mendling and A. Polleres (2016). "Resource Allocation with Dependencies in Business Process Management Systems". In *Proceedings of the Business Process Management Forum 2016*, pp. 3–19.

Heinrich, B., M. Bolsinger and M.-A. Bewernik (2009). "Automated planning of process models: the construction of exclusive choices". In: *Proceedings of the 30th International Conference on Information Systems (ICIS 2009)*.

Heinrich, B., M. Klier and S. Zimmermann (2012). "Automated Planning of Process Models –Towards a Semantic-based Approach". In *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pp. 169–194: IGI Global.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models: Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

Heinrich, B. and D. Schön (2015). "Automated Planning of Context-aware Process Models". In: *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*.

Heinrich, B. and D. Schön (2016). "Automated Planning of Process Models: The Construction of Simple Merges". In: *Proceedings of the 24rd European Conference on Information Systems (ECIS 2016)*.

Henneberger, M., B. Heinrich, F. Lautenbacher and B. Bauer (2008). "Semantic-Based Planning of Process Models". In *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2008)*, pp. 1677–1689.

Hoffmann, J., I. Weber and F. M. Kraft (2009). "Planning@SAP: An application in business process management". In: *Proceedings of the 2nd International Scheduling and Planning Applications woRKshop (SPARK 2009)*.

Hoffmann, J., I. Weber and F. M. Kraft (2012). "SAP Speaks PDDL: Exploiting a Software-Engineering Model for Planning in Business Process Management" *Journal of Artificial Intelligence Research* 44 (1), 587–632.

Hornung, T., A. Koschmider and A. Oberweis (2007). "A Rule-based Autocompletion Of Business Process Models". In: *Proceedings of the 19th Conference on Advanced Information Systems Engineering (CAiSE 2007)*.

Huang, J. S., K. Hsueh and A. Reynolds (2013). "A framework for collaborative social, economic and environmental development: Building a digital ecosystem for societal empowerment". In: *Proceedings of 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST 2013) - Complex Environment Engineering*, pp. 166–171.

IEEE Task Force on Process Mining (2012). "Process Mining Manifesto". In *Business Process Management Workshops*, pp. 169–194: Springer Berlin Heidelberg.

Jennings, N. R., T. J. Norman, P. Faratin, P. O'Brien and B. Odgers (2000). "Autonomous agents for business process management" *Applied Artificial Intelligence* 14 (2), 145–189.

Kannengiesser, U. (2017). "The Future. Obstacles and Opportunities". In *S-BPM in the Production Industry: A Stakeholder Approach*, pp. 209–230: Springer International Publishing.

Katzmarzik, A., M. Henneberger and H. U. Buhl (2012). "Interdependencies between automation and sourcing of business processes" *Journal of Decision Systems (JDS)* 21 (4), 331–352.

Khan, F. H., S. Bashir, M. Y. Javed, A. Khan and M. S. H. Khiyal (2010). "QoS Based Dynamic Web Services Composition & Execution" *International Journal of Computer Science and Information Security (IJCSIS)* 7 (2), 147–152.

Klier, J., M. Klier, A.-L. Müller and C. Rauch (2016). "The impact of self-service technologies – towards an economic decision model and its application at the German Federal Employment Agency" *Journal of Decision Systems (JDS)* 25 (2), 151–172.

Kossak, F., C. Illibauer, V. Geist, C. Natschläger, T. Ziebermayr, B. Freudenthaler, T. Kopetzky and K.-D. Schewe (2016). "A Layered Approach for Actor Modelling". In *Hagenberg Business Process Modelling Method*, pp. 63–84: Springer International Publishing.

Lambert, D. M., M. A. Emmelhainz and J. T. Gardner (1996). "Developing and Implementing Supply Chain Partnerships" *The International Journal of Logistics Management (IJLM)* 7 (2), 1–17.

Leymann, F., D. Roller and M. T. Schmidt (2002). "Web services and business process management" *IBM Systems Journal* 41 (2), 198–211.

Liu, C., Q. Zeng, H. Duan and F. Lu (2015). "Petri Net Based Behavior Description of Cross-Organization Workflow with Synchronous Interaction Pattern". In *Process-Aware Systems*, pp. 1–10: Springer Berlin Heidelberg.

Liu, C. and F. Zhang (2016). "Petri Net Based Modeling and Correctness Verification of Collaborative Emergency Response Processes" *Cybernetics and Information Technologies* 16 (3).

Mendling, J., H. M. W. Verbeek, B. F. van Dongen, W. M. P. van der Aalst and G. Neumann (2008). "Detection and prediction of errors in EPCs of the SAP reference model" *Data & Knowledge Engineering (DKE)* 64 (1), 312–329.

Meredith, J. R., A. Raturi, K. Amoako-Gyampah and B. Kaplan (1989). "Alternative research paradigms in operations" *Journal of Operations Management (JOM)* 8 (4), 297–326.

Meyer, H. and M. Weske (2006). "Automated service composition using heuristic search". In *Business Process Management*, pp. 81–96: Springer Berlin Heidelberg.

Mitroff, I. I., F. Betz, L. R. Pondy and F. Sagasti (1974). "On Managing Science in the Systems Age. Two Schemas for the Study of Science as a Whole Systems Phenomenon" *Interfaces* 4 (3), 46–58.

Natschläger, C. and V. Geist (2013). "A layered approach for actor modelling in business processes" *Business Process Management Journal (BPMJ)* 19 (6), 917–932.

Nissim, R., R. I. Brafman and C. Domshlak (2010). "A general, fully distributed multi-agent planning algorithm". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, pp. 1323–1330.

Object Management Group (2013). *Business Process Model and Notation (BPMN). Version 2.0.2*. URL: http://www.omg.org/spec/BPMN/2.0.2/PDF (visited on 10/16/2014).

Object Management Group (2015). *OMG Unified Modeling Language TM (OMG UML). Version 2.5*. URL: http://www.omg.org/spec/UML/2.5 (visited on 05/04/2016).

Ou-Yang, C. and H. Winarjo (2011). "Petri-net integration – An approach to support multi-agent process mining" *Expert Systems with Applications* 38 (4), 4039–4051.

Paik, I., W. Chen and M. N. Huhns (2014). "A scalable architecture for automatic service composition" *IEEE Transactions on Services Computing* 7 (1), 82–95.

Peleteiro, A., J. C. Burguillo, J. L. Arcos and J. A. Rodriguez-Aguilar (2014). "Fostering Cooperation Through Dynamic Coalition Formation and Partner Switching" *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 9 (1), 1:1-1:31.

Pulgar, J. and M. C. Bastarrica (2017). "Transforming Multi-role Activities in Software Processes into Business Processes". In *Business Process Management Workshops 2016*.

Recker, J. C., M. Indulska, M. Rosemann and P. Green (2006). "How Good is BPMN Really? Insights from Theory and Practice". In: *Proceedings of the 14th European Conference on Information Systems (ECIS 2006).*

Rosemann, M. and J. vom Brocke (2015). "The Six Core Elements of Business Process Management". In *Handbook on Business Process Management 1. Introduction, Methods, and Information Systems*. 2nd Edition, pp. 107–122: Springer.

Roy, S., A. S. M. Sajeev, S. Bihary and A. Ranjan (2014). "An Empirical Study of Error Patterns in Industrial Business Process Models" *IEEE Transactions on Services Computing* 7 (2), 140–153.

Rozinat, A., S. Zickler, M. Veloso, W. M. P. van der Aalst and C. McMillen (2009). "Analyzing Multi-agent Activity Logs Using Process Mining Techniques". In *Distributed Autonomous Robotic Systems 8*, pp. 251–260: Springer Berlin Heidelberg.

Russell, N., W. M. P. van der Aalst, A. H. M. ter Hofstede and D. Edmond (2005). "Workflow Resource Patterns. Identification, Representation and Tool Support". In *Advanced Information Systems Engineering*, pp. 216–232: Springer Berlin Heidelberg.

Schönig, S., C. Cabanillas, S. Jablonski and J. Mendling (2015). "Mining the organisational perspective in agile business processes". In: *International Conference on Enterprise, Business-Process and Information Systems Modeling*, pp. 37–52.

Serve, M., D. C. Yen, J.-C. Wang and B. Lin (2002). "B2B-enhanced supply chain process: toward building virtual enterprises" *Business Process Management Journal (BPMJ)* 8 (3), 245–253.

Shapiro, R., S. A. White, C. Bock, N. Palmer, M. zur Muehlen, Brambilla. Marco and D. Gagné (2012). *BPMN 2.0 handbook second edition. Methods, concepts, case studies and standards in business process management notation.* 2nd ed.: Future Strategies.

Shoham, Y. and M. Tennenholtz (1995). "On social laws for artificial agent societies: off-line design" *Artificial Intelligence* 73 (1), 231–252.

Siha, S. M. and G. H. Saad (2008). "Business process improvement. Empirical assessment and extensions" *Business Process Management Journal (BPMJ)* 14 (6), 778–802.

Skjoett-Larsen, T., C. Thernøe and C. Andresen (2003). "Supply chain collaboration" *International Journal of Physical Distribution & Logistics Management (IJPDLM)* 33 (6), 531–549.

Stadtler, H., C. Kilger and H. Meyr (eds.) (2015). *Supply Chain Management and Advanced Planning:* Springer Berlin Heidelberg.

Štolba, M. and A. Komenda (2013). "Fast-forward heuristic for multiagent planning". In: *Proceedings of the 1st Workshop on Distributed and Multi-Agent Planning.*

Sycara, K., M. Paolucci, A. Ankolekar and N. Srinivasan (2003). "Automated discovery, interaction and composition of semantic web services" *Web Semantics: Science, Services and Agents on the World Wide Web* 1 (1), 27–46.

The Workflow Management Coalition Specification (WfMC) (1999). *Terminology & Glossary. WFMC-TC-1011 (Issue 3.0).*

Torreño, A., E. Onaindia and Ó. Sapena (2012). "An approach to multi-agent planning with incomplete information". In: *Proceedings of 20th Biennial European Conference on Artificial Intelligence (ECAI 2012)*.

Torreño, A., E. Onaindia and Ó. Sapena (2014a). "FMAP: Distributed cooperative multi-agent planning" *Applied Intelligence* 41 (2), 606–626.

Torreño, A., E. Onaindia and Ó. Sapena (2014b). "Integrating individual preferences in multi-agent planning". In: *Proceedings of the 2nd Workshop on Distributed and Multi-Agent Planning*.

van der Aalst, W. M. P. (1999). "Interorganizational Workflows: An Approach Based on Message Sequence Charts and Petri Nets" *Systems Analysis - Modelling - Simulation* 34 (3), 335–367.

van der Aalst, W. M. P. (2015). "Extracting Event Data from Databases to Unleash Process Mining". In *BPM - Driving Innovation in a Digital World*, pp. 105–128: Springer International Publishing.

van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns" *Distributed and Parallel Databases* 14 (1), 5–51.

van der Aalst, W. M. P. and K. M. van Hee (2002). *Workflow Management: Models, methods and systems:* MIT Press.

van der Aalst, W. M. P., A. J. M. M. Weijters and L. Maruster (2004). "Workflow mining: Discovering process models from event logs" *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 16 (9), 1128–1142.

Wang, P., Z. Ding, C. Jiang and M. Zhou (2014). "Automated web service composition supporting conditional branch structures" *Enterprise Information Systems* 8 (1), 121–146.

Weber, I. (2007). "Requirements for Implementing Business Process Models through Composition of Semantic Web Services". In *Enterprise Interoperability II*, pp. 3–14: Springer London.

Weber, I., J. Hoffmann and J. Mendling (2008). "Semantic business process validation". In: *Proceedings of the 3rd International Workshop on Semantic Business Process Management*.

Wetzstein, B., Z. Ma, A. Filipowska, M. Kaczmarek, S. Bhiri, S. Losada, J.-M. Lopez-Cob and L. Cicurel (2007). "Semantic Business Process Management: A Lifecycle Based Requirements Analysis". In: *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) in conjunction with the 3rd European Semantic Web Conference (ESWC 2007)*, pp. 7–17.

Wohed, P., W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede and N. Russell (2006). "On the Suitability of BPMN for Business Process Modelling". In *Business Process Management*, pp. 161–176: Springer Berlin Heidelberg.

Wooldridge, M. J. (2009). *An introduction to multiagent systems.* 2nd Edition: John Wiley & Sons Ltd.

Zhang, J. (2017). "The Technical Foundation of a Multi-Agent System". In *Multi-Agent-Based Production Planning and Control*, pp. 21–53: John Wiley & Sons Singapore Pte. Ltd.

Appendix A: Pseudocode of the Main Primitives of our Algorithm

Listing 1. CreateWorldStates – Primitive *main*

```
1       SUB main
2         StatesThatLeadToGoal = {};
3         FOR actor ∈ ACTORS
4           bs = actor.initial_state
5           result = plan(bs)
6           IF result == true
7             StatesThatLeadToGoal.add(bs)
8       ENDSUB
```

Listing 2. CreateWorldStates – Primitive *plan*

```
1       SUB plan(bs)
2         result = false
3         FOR a ∈ ACTIONS
4             IF checkForApplicability(a in bs) == true
5               stateTransition(bs, a)
6             ELSE IF checkApplicAfterDisband(a in bs)
7              == true
8             result = disband(bs, a)
9           ELSE IF checkApplicWithJoins(a in bs)
10           == true
11            result = join(bs, a)
12        RETURN result
13      ENDSUB
```

Listing 3. CreateWorldStates – Primitive *stateTransition*

```
1       SUB stateTransition(bs, a)
2         result = false
3         bsnew = R(bs, a)
4           IF bsnew is goal
5             result = true
6           ELSE
7             result = plan(bsnew)
8         RETURN result
9       ENDSUB
```

Listing 4. CreateWorldStates – Primitive *join*

```
1       SUB join(bs, a)
2         saveForFutureJoins(bs, a)
3         JointStates = retrieveValidJointStates(bs)
4           joinPossible = false
5           FOR jointState ∈ JointStates
6            IF stateTransition(jointState, a) == true
7              constructJoinActions(jointState)
8            joinPossible = true
9         RETURN joinPossible
10      ENDSUB
```

Listing 5. CreateWorldStates – Primitive *disband*

```
1       SUB disband(bs, a)
2         DisbandedStates = retrieveValidDisbStates(bs)
3         disbandPossible = false
4           FOR disbandedState ∈ DisbandedStates
5            IF stateTransition(disbandedState, a) == true
6              constructSplitAction(disbandedState)
7              disbandPossible = true
8           ELSE IF checkApplicabilityWithJoin(a in
9            disbandedState) == true
10           IF join(disbandedState, a) == true
11             constructSplitAction(disbandedState)
12             disbandPossible = true
13        RETURN disbandPossible
14      ENDSUB
```

Appendix B: Mathematical Evaluation

Theorem 1. The execution of the algorithm terminates.

Proof sketch. Termination is shown by proving that only a finite number of iteration steps is performed, and that each iteration step of the algorithm terminates. Let *s=1,2,..* be the iteration steps and *S* be the set of all performed iteration steps.

We first prove that $|S| < \infty$. Let *R(bs, a)* be the transition function which, for an action *a* applicable in a belief state *bs*, provides the belief state resulting from the application of *a* in *bs*, and *R(bs, a)*=∅ for an action *a* not applicable in *bs*. Let $\bigcup_{i \in ACTORS} \bigcup_{a \in ACTIONS} R(Init_i, a) =: bs_1$. Due to $|ACTIONS| < \infty$ and $|ACTORS| < \infty$, $|bs_1| < \infty$. Iteratively defining $bs_k := \bigcup_{bs \in bs_{k-1}} \bigcup_{a \in ACTIONS} R(bs, a)$ for each $k \in \mathbb{N}$, $k \geq 2$, it equivalently follows that $|bs_k| < \infty$ for each $k \in \mathbb{N}$ because of $|bs_{k-1}| < \infty$ and $|ACTIONS| < \infty$. There is a $l \in \mathbb{N}$ such that for all $bs \in bs_l$: $bs \in \bigcup_{k=1,\dots,l-1} bs_k \cup \bigcup_{i \in ACTORS} Init_i$, hence $|\bigcup_{k \in \mathbb{N}} bs_k| < \infty$. In other words, only a finite number of different belief states can be constructed based on the application of actions in *ACTIONS* to the initial states and the thereby constructed belief states (*). Additionally, join actions and split actions can be constructed during the course of the algorithm; let *C* be the set comprising all constructed join actions and *D* be the set comprising all constructed split actions. Due to $|ACTORS| < \infty$ and (*), $|C| < \infty$ and $|D| < \infty$. Thus, altogether, the number of actions considered and planned is finite because of $|ACTIONS|+|C|+|D| < \infty$. Following this, analogous to above, the number of different belief states constructed by the algorithm is finite (**).

If a belief state *bs* has already been considered in an earlier iteration step, the algorithm does not perform another iteration step for *bs*. Because of (**), there is a $t \in \mathbb{N}$ and an iteration step s_t in which all belief states have already been considered in the iteration steps $s_1, \dots, s_{t-1}$. Hence, altogether, $|S| < \infty$.

An iteration step of the algorithm consists of the sub steps ***1a)-1c)*** and ***2a)-2e)*** described in the algorithm subsection. Step ***1a)*** terminates since only a finite number of actions needs to be checked for applicability (as $|ACTIONS| < \infty$), and each such check terminates as just a finite number of simple set comparisons is required (cf. Definition 4''). Step ***1b)*** terminates because the criteria i. and ii. for splitting a belief state can be checked trivially and the termination of examining criterion iii. is equivalent to the termination of step ***1a)***, which was already proved. Step ***1c)*** terminates obviously as only a finite number of simple set comparisons is necessary. Steps ***2a)-2e)*** need to be performed only a finite number of times in each $s \in S$ because of $|ACTIONS| < \infty$. Step ***2a)*** terminates also due to this reason and because of $|eff(a)| < \infty$ for each $a \in ACTIONS$. Step ***2b)*** terminates as the creation of a split action only requires a finite number of simple set operations and the subsequent belief state is constructed just like in step ***2a)***. Step ***2c)*** is computationally trivial. Step ***2d)*** terminates because due to $|ACTIONS| < \infty$ and (**), only a finite number of combinations needs to be checked, and each check is equivalent to performing step ***1a)***. Step ***2e)*** terminates due to the same reasons as step ***2b)***. *q.e.d.*

Theorem 2. The algorithm constructs correct process models: All planned paths are feasible.

Proof sketch. To prove the correctness of the generated process models, it suffices to show that 1) the actions generated by the algorithm do not lead to logical contradictions within the process model, and that 2) in no belief state an action which is not applicable can be planned by the algorithm. We start with 1). In step ***2b)*** of the algorithm, split actions are generated; let *bs* be a belief state which is split into the belief states $bs_{1,\dots,}bs_n$. Because of criteria i. and ii., $A(bs_i) \cap A(bs_j) = \emptyset$ for all $i,j \in \{1,\dots,n\}$ such that $i \neq j$ and $\cup_i A(bs_i) = A(bs)$, which leads to $A(bs) = \uplus_i A(bs_i)$ (disjoint union). Hence, split actions do not lead to logical contradictions. The logical consistency of the join actions generated in step ***2e)*** of the algorithm is ensured by the respective criteria i. and ii. which prevent logically contradictory belief states: i. excludes belief states which contain the same actors from joining and ii. guarantees that belief states which contain non-actor-specific variables with contradicting restrictions cannot be joined. In regard to 2), actions are planned in the steps ***2a)***, ***2b)*** and ***2e)*** of the algorithm. Actions planned are either actions included in *ACTIONS* or are join/split actions generated by the algorithm. As all of these actions are actions in the sense of Definition 3' and Definition 4'' is used for the applicability check, we do not differentiate between them further. The actions planned in the steps ***2b)*** and ***2e)*** are applicable by definition (as their preconditions and cardinality match the considered belief state) and in step ***2a)***, an applicability check is performed before planning an action. *q.e.d.*

Theorem 3. The algorithm constructs complete process models: All feasible paths leading from an initial state to a goal state are being planned.

Proof sketch. It suffices to show that starting from a belief state *bs*, all actions *a* in *ACTIONS* that can possibly be applied as next action are planned by our algorithm. Let *R(bs, a)* be the transition function which, for an action *a* applicable in a belief state *bs*, provides the belief state resulting from the application of *a* in *bs*, and *R(bs, a)*=∅ for an action *a* not applicable in *bs*. There are four cases:

(1) *a* is applicable in *bs*

(2) *a* is applicable in *R(bs, d)*, where *d* is a split action

(3) *a* is applicable in *R(bs, c)*, where *c* is a join action

(4) *a* is applicable in *R(R(bs, d), c)*, where *d* is a disband action and *c* is a join action

Ad (1): The action *a* is identified in step ***1a)*** and planned in step ***2a)***.

Ad (2): The action *d* is identified in step ***1b)*** and planned in step ***2b)***; thereafter, the action *a* is planned accordingly in the belief state *R(bs, d)* (cf. (1)).

Ad (3): In steps ***1c)***, ***2c)*** and ***2d)*** possibilities for join actions are identified. In particular, the matching performed in step ***2d)*** guarantees the consideration of all possible state-combinations

that can be joined. Thus, *c* is planned in step ***2e)***, which enables the subsequent planning of *a* in the belief state *R(bs, c)* (cf. (1)).

Ad (4): The possibility is identified in step ***1b)*** in conjunction with steps ***1c)***, ***2c)*** and ***2d)*** and the actions are planned accordingly. *q.e.d.*

5 Conclusion

The major findings of the dissertation are summarized in Section 5.1. Section 5.2 discusses directions for further research. The points identified within this section address the dissertation as a whole and are not specific to single papers presented in the Sections 2, 3 and 4, since respective summaries and outlooks are contained directly within the papers.

5.1 Major Findings

The rising availability of uncertain data, the emergence of unstructured data and a complex, dynamically changing environment are influential developments which urgently require organizations to transform their decision-making and business processes in order to stay competitive. AI offers valuable concepts and methods to tackle these developments and address research gaps in existing literature. The dissertation utilizes and furthers some of these contributions from AI in the focal points assessment of data quality (Section 2), analysis of textual data (Section 3) and automated planning of process models (Section 4). In this way, it provides novel concepts and methods suitable to assist organizations in the improvement of their data-driven decision-making as well as their business process management. The main takeaways from the three focal points are as follows.

With respect to the first focal point, concrete probability-based approaches for the assessment of data quality in regard to semantic consistency (Section 2.1) and duplicates (Section 2.2) are presented, addressing RQ1 and RQ2 respectively. Formal definitions of the approaches and multiple possibilities for their instantiation are specified. Engaging the AI field decision-making under uncertainty, both approaches utilize concepts and methods from probability theory for the quantification of uncertainty. This leads to results that are interpretable as probabilities, which in turn enables well-founded support for data-driven decision-making and, in particular, their integration into expected value calculus. Moreover, both approaches are evaluated based on applications to real-world customer data from an insurer. Applying the metric for semantic consistency allows to identify a specific consistency issue in the data and, due to the interpretability of the metric values, to pinpoint which records are probably erroneous and which ones to treat as trustworthy. In this way, future data-driven decision-making by the insurer is supported, for instance by improved targeting in customer campaigns. Similarly, applying the probability-based approach for duplicate detection to the customer data facilitates the identification of duplicates caused by a real-world event and the utilization of the results for decision support. In essence, these two contributions show that probability-based methods can be highly beneficial for the assessment of data quality, and that the interpretability of their results in particular supports data-driven decision-making. To foster the development of further such approaches, five requirements for data quality metrics are proposed in Section 2.3, addressing RQ3. The requirements are condensed from literature, clearly defined and applied to evaluate well-known existing data quality metrics. Furthermore, they are justified based on a decision-oriented

framework and shown to be indispensable for metrics aiming to support an economically oriented management of data quality and decision-making under uncertainty. Overall, the dissertation thus proposes two explicit probability-based approaches as well as requirements for further approaches for the assessment of data quality striving to support data-driven decision-making.

In regard to the second focal point, the dissertation presents approaches for gaining insights from CVs (Section 3.1) and online customer reviews (Section 3.2). To discover knowledge from CVs, a topic modeling procedure consisting of five steps is introduced, while for online customer reviews, a model to explain and interpret the overall star ratings based on aspect-based sentiment analysis is proposed, addressing RQ4 and RQ5 respectively. Both works exploit concepts and methods from natural language processing and other (AI) fields. This allows them to take the specifics of the data to be analyzed into account and to provide interpretable results suitable for the support of data-driven decision-making. More precisely, the topic modeling procedure is adapted from a process suggested in literature and considers the characteristics of CVs in each step in order to discover interpretable topics describing fine-grained competences. The aspect-based sentiment analysis is based on a generalized ordered probit model and a likelihood-based pseudo R-squared measure which are adjusted to the context at hand to examine customer assessments and opinions expressed in the reviews' texts and (ordinal) rating scale. Both approaches are applied to real-world data. The topic modeling procedure discovers clearly interpretable topics representing specific competences (e.g., Java programming) when applied to CVs from IT experts. These topics can be used in a variety of ways to provide decision support in human resource management processes, for instance facilitating a topic-based search which is shown to proficiently identify candidates for job offers. The model for analyzing customer reviews is applied to a large database of restaurant reviews, leading to results that are easy to grasp and provide valuable insights. For instance, revealing why specific customer ratings were assigned to a product, service, company or competitor enables data-driven decision support for the development of customer-centric solutions to improve customer satisfaction. To sum up, the two proposed approaches allow for a well-founded analysis of two types of textual data, CVs and online customer reviews, yielding interpretable results which are readily available to support data-driven decision-making and business processes in organizations.

With respect to the third focal point, automated planning approaches for the construction of parallelizations (Section 4.1), the adaptation of process models (Section 4.2) and the construction of multi-actor process models (Section 4.3) are proposed, addressing RQ6, RQ7 and RQ8 respectively. The three approaches share a common conceptual basis from AI planning in form of the underlying planning domain. Moreover, each of the approaches comprises novel concepts as well as a concrete method (i.e., algorithm). These allow for an improved representation of processes conducted in a complex environment (by incorporating parallelizations and multiple actors in the process models) and an automated adaptation of process models. The key properties of the approaches are formally verified. Additionally, the approaches are further evaluated in real-world scenarios. In particular, applying the approach for an automated construction of

parallelizations leads to additional, feasible parallelizations being constructed (in comparison to manual planning), and, thus, increased business process flexibility. Moreover, the constructed parallelizations enhance the decision-making aspect of process models by allowing to select a beneficial way for process execution (e.g., based on temporal, economic and resource criteria constraints). The proposed automated adaptation of process models is shown to soundly and swiftly adapt existing process models to needs for change in advance and in this way strengthens business process agility and flexibility. Similarly, the presented conceptual foundation and approach for the construction of multi-actor process models are shown to adequately represent multi-actor processes conducted in practice and consequently promote business process agility in organizations. Further, the constructed multi-actor process models help individual actors to achieve their individual goals from a decision support perspective. Altogether, all three approaches proposed in the dissertation expand the boundaries of automated planning of process models, contribute to improved BPM and, in particular, support business process agility. Besides, the works also have implications for decision-making.

In summary, the dissertation provides concrete novel concepts and methods in three focal points, supporting organizations in transforming their decision-making and business processes as they face technology-driven developments. Yet, a plethora of interesting directions for further research remain.

5.2 Directions for Further Research

The dissertation has concentrated on eight specific research questions to address important issues with respect to the selected focal points in depth. Nevertheless, technology-driven developments certainly influence organizations in a broader sense than what could be covered by this work, and AI concepts and methods are not limited to the scope of the dissertation with respect to supporting data-driven decision-making and business processes. Some possible directions for further research in this area are outlined in the following.

To begin with, in regard to the first focal point, the dissertation has presented concrete approaches for the assessment of data quality with respect to semantic consistency (in Section 2.1) and duplicates (in Section 2.2). Yet, data quality is a multidimensional construct, and the assessment of data quality regarding further important dimensions such as accuracy, completeness and currency is also vital in both research and practice (Batini and Scannapieco, 2016; Fan, 2015; Wand and Wang, 1996; Wang and Strong, 1996). Thus, existing metrics for these dimensions (e.g., Blake and Mangiameli, 2011; Fisher et al., 2009; Zak and Even, 2017) should be examined and possibly refined to provide interpretable results which are helpful for data-driven decision-making. If needed, new metrics should be developed. The requirements for data quality metrics proposed in Section 2.3 offer valuable guidance for these tasks. Striving for approaches which allow the interpretation of the metric results as probabilities – similar to the research presented in Sections 2.1 and 2.2 and a large body of work dealing with currency (e.g., Heinrich et al., 2009b; Heinrich and Klier, 2015) – may be a promising starting point.

Yet, despite exclusively probability theory being used in this dissertation, it should be noted that there are further valid ways to model uncertainty, which may also be useful to assess data quality and support data-driven decision-making. This view is shared by AI research which has long deemed probability theory as insufficient to completely handle uncertainty (Zadeh, 1986). Fuzzy set theory and the related possibility theory (Liu, 2015; Zadeh, 1965; Zimmermann, 2011) lift assumptions of classical set theory and thus are suggested to be appropriate for modeling uncertainties stemming from the inability of humans to make precise estimations, for instance, in the context of data quality assessment (Bronselaer et al., 2018; Bronselaer and Tré, 2016; Hristova, 2016). In particular, as demonstrated by Heinrich and Hristova (2014) with respect to currency, fuzzy set theory can successfully be used to develop data quality metrics based on expert estimations. It is promising to extend this research strand and develop further data quality metrics based on fuzzy set theory, for instance, with respect to other dimensions such as semantic consistency. Taking up ideas from the metric presented in Section 2.1, a fuzzy metric for semantic consistency could be based on a comparison of rule fulfillments in the data to be assessed and the expected rule fulfillment modeled as a fuzzy set.

Moreover, the dissertation mainly contributes to the measure-phase of the Total Data Quality Management methodology (Wang, 1998). Still, the developed concepts and methods also offer starting points for analyzing the roots of data quality issues and a well-founded improvement of data quality, corresponding to the analyze-phase and the improve-phase. These phases are important to actually realize benefits in organizations that are enabled by the define-phase and the measure-phase, for instance with respect to data-driven decision-making and an economically oriented management of data quality as pointed out in Section 2.3. Yet, as the analysis of data quality issues and the cost-benefit-comparison conducted in Section 2.1 have shown, related tasks in these phases are neither trivial to execute nor to evaluate. Thus, to prevent these tasks from being performed in an inefficient, ad hoc manner, putting the success of data quality projects at risk, respective general guidelines should be suggested. Similar to the research conducted in Section 2.3, such a set of guidelines could be condensed from contributions in the literature (e.g., Batini et al., 2009; Hazen et al., 2017; Loshin, 2010) and subsequently justified.

Another interesting avenue for research in this direction is to develop approaches which integrate data quality into data analysis methods. Most data analysis methods per default consider their input data to be of perfect data quality, which rarely is the case in practice. The resulting negative impact for data-driven decision-making can be substantial (e.g., Blake and Mangiameli, 2011; Feldman et al., 2018), and the need for data quality-aware data analysis methods is thus urgent. This issue has been acknowledged in the literature and has brought forth approaches for so called uncertain data mining (Aggarwal, 2010; Schubert et al., 2015) and for assessing consequences of poor data quality to AI methods such as neural networks (Kavzoglu, 2009; Klein and Rossin, 1999; Zhang, 2006). Yet, these works mostly focus on handling a given uncertainty and do not concentrate on explicitly modeling the uncertainty. Still, these contributions can serve as valuable foundation for further research. Moreover, a blueprint for envisioned

work is given by Hristova (2014), who develops an approach for considering currency in decision tree classification.

In line with this aspect is also the opportunity to further studies on the impact of data quality on decision support systems. Previous work in this area (e.g., Blake and Mangiameli, 2011; Feldman et al., 2018; Woodall et al., 2014) mainly investigates the influence of data quality on data mining outcome and related decision support. Less research has been conducted in regard to recommender systems, an important and increasingly prevalent category of decision support systems (e.g., Adomavicius et al., 2018; Melville and Sindhwani, 2017; Power et al., 2015), which strives to guide users to their individually best choice when a large number of alternatives is available. Recommender systems have become indispensable especially in e-commerce and electronic markets and have contributed tremendously to the success of platforms such as *YouTube*, *Amazon*, *Netflix* and *Spotify* (cf., e.g., Gomez-Uribe and Hunt, 2016; Li and Karahanna, 2015; Lu et al., 2015). Here, while the importance of data quality for recommender systems is recognized in general (e.g., Berkovsky et al., 2012; Sar Shalom et al., 2015), existing works predominantly focus on selected aspects such as analyzing ratings with respect to currency (De Pessemier et al., 2010) or aggregating user data to obtain a more complete view on user behavior (Abel et al., 2013; Ozsoy et al., 2015). In recent work (Heinrich et al., 2019), the impact of completeness of item content data on recommendation quality has been studied, acknowledging the documented relevance of a comprehensive view on an item's characteristics for recommender systems (Lops et al., 2011; Picault et al., 2011). Still, these contributions leave interesting research gaps such as examining the impact of different data quality improvement measures on recommendation quality, trade-offs between these measures and associated costs, and the analysis of further data quality dimensions and recommender system quality measures (Bobadilla et al., 2018) to develop more comprehensive approaches.

A further direction for research in this focal point is to concentrate on another characteristic of big data: Velocity. In particular, data streams generated by, for instance, *Twitter* data, network traffic, GPS data, sensor networks, medical monitoring devices and customer click streams have become a valuable resource for data analysis and data-driven decision-making in organizations (Arasu et al., 2016; Gama, 2010; Kulkarni et al., 2015). Such data streams pose new challenges for data quality assessment (Cai and Zhu, 2015), most notably because data may dynamically evolve over time, for instance following underlying infrastructural or behavioral changes (Aggarwal, 2007). Data streams exhibiting data quality issues, so called uncertain data streams, have been studied in existing literature (e.g., Cao et al., 2015; Ma et al., 2016), but similar to the research field uncertain data mining, such works do not focus on explicitly modeling uncertainty but rather coping with it. Nevertheless, this body of research, especially works dealing with anomaly detection in data streams (e.g., Ahmad et al., 2017; Rettig et al., 2019), may provide helpful input for developing data quality assessment approaches for data streams.

Finally, another research opportunity in this focal point is work that links to the second focal point and considers the variety of big data: Data quality assessment of unstructured data, in

particular, textual data. As already pointed out by existing literature (e.g., Gandomi and Haider, 2015; Ghasemaghaei and Calic, 2019; Sivarajah et al., 2017) and this dissertation, unstructured data is becoming increasingly prevalent in organizational data analysis. Yet, just like structured data, unstructured data is often erroneous, causing detrimental effects to data-driven decision-making (Cai and Zhu, 2015). Thus, the need to develop concepts and methods for data quality assessment of unstructured data has been acknowledged, as existing metrics for structured data cannot be readily applied (Batini and Scannapieco, 2016; Cai and Zhu, 2015). While some contributions have been suggested in this regard (e.g., Batini et al., 2011; Kiefer, 2016, 2019), these metrics can only be considered to be first steps since they suffer from serious shortcomings such as limited interpretability and reliability due to unclear and choice-dependent definitions. A promising idea to tackle the issue may be to transfer existing metrics for structured data to specific unstructured data such as wikis (and related data structures such as knowledge graphs, which are often constructed based on wikis), where a significant amount of research with respect to detecting errors has already been conducted (e.g., Dalip et al., 2017; Färber et al., 2018; Wienand and Paulheim, 2014). In the future, such data quality assessment for unstructured data might even be expanded to other kinds of data such as images, audios or videos.

With respect to the second focal point, the focus of the dissertation has been on the analysis of two types of texts: CVs (in Section 3.1) and online customer reviews (in Section 3.2). Yet, these types of text evidently only represent a certain share of the manifold of textual data available to organizations for analysis. Indeed, in particular on online social media platforms such as *Facebook*, *WhatsApp*, *Instagram*, *YouTube* and *Twitter*, users communicate information and opinion about any subject imaginable. Posts and messages on these platforms thus provide a rich resource of textual data to gain insights from which should be further explored. There has already been a plethora of research on information systems related issues such as general sentiment analysis (e.g., Agarwal et al., 2011; Pak and Paroubek, 2010; Rosenthal et al., 2017), competitive analysis (e.g., Dey et al., 2011; He et al., 2013; He et al., 2015) and brand analysis (e.g., Camiciottoli et al., 2014; Ghiassi et al., 2013; Tirunillai and Tellis, 2014), aiming to support data-driven decision-making in organizations. These analyses are often facilitated by AI concepts and methods. However, these texts could also be used further to, for instance, support the estimation of customers' value, extending previous literature striving to value customers in networks (e.g., Baethge et al., 2017; Däs et al., 2017; Klier et al., 2014).

Furthermore, from a methodical point of view, both topic modeling (in Section 3.1) and sentiment analysis (in Section 3.2) have been successfully used in the dissertation to analyze textual data and gain valuable insights. As an expansion to the conducted research, it is promising to explore whether a combination of these two methods is able to produce results which support data-driven decision-making in an even more fine-grained way. Such a combination has initially been suggested by Lin and He (2009) in form of a "joint sentiment/topic model", a natural language processing approach which detects sentiments and topics simultaneously, thus enabling an understanding of positively and negatively discussed topics in text. This idea as well as similar proposals have been taken up by various researchers striving to, for instance, further

the analysis of online customer reviews (e.g., Diao et al., 2014; Jo and Oh, 2011; Linshi, 2014; McAuley and Leskovec, 2013; Wang et al., 2010; Wang et al., 2016). However, the results of these approaches often suffer from a lack of interpretability (e.g., due to employing latent topics). Moreover, most of the approaches do not seek to actually *explain* the overall star ratings of online customer reviews. It would thus be interesting to enrich the latter body of work with the findings from Section 3.2 to advance in this quest. In addition to that, a combined approach may also be applicable to other kinds of text worth investigating.

A further interesting direction in this area is to follow approaches which aim to identify emotions rather than just a positive, negative or neutral sentiment in textual data. Based on cognitive studies providing the theoretical background for the most relevant emotion categories (e.g., Ekman, 1992; Plutchik, 2001) and natural language processing, work in this field additionally associates emotions such as joy, surprise or disgust to text, extending classical sentiment analysis (e.g., Giatsoglou et al., 2017; Yadollahi et al., 2017). While hardly treated in literature, an aspect-based emotion analysis of texts is also feasible (Yadollahi et al., 2017). To this end, aspect extraction approaches (often enabled by neural networks; e.g., Poria et al., 2016; Rana and Cheah, 2016; Wang et al., 2016; Xu et al., 2019) can be used in conjunction with emotion lexicons (e.g., Bandhakavi et al., 2017; Mohammad and Turney, 2013; Strapparava et al., 2004). For instance, when analyzing online customer reviews, such an approach could provide detailed, refined insights into customer assessments and opinions, opening doors for improved data-driven decision-making in organizations. It is thus promising to transfer the concepts and methods developed in Section 3.2 to support an aspect-based emotion analysis which strives to explain the overall star ratings of online customer reviews in a methodically sound, interpretable and fine-grained way.

More generally, the endeavor of gaining insights from textual data in an automated manner could be furthered by establishing a more comprehensive view comprising different perspectives expressed as interpretable features. For the example of online customer reviews, such approaches have been suggested by, for instance, Chatterjee (2019), Luo and Tang (2019) and Siering et al. (2018). Indeed, while the aspect-based sentiments considered in Section 3.2 represent the most prevalent perspective (e.g., cf. also Chatterjee, 2019; Jabr et al., 2018; Liu et al., 2017a; Luo and Tang, 2019), the relevance of the customer's context (e.g., with respect to location, time, weather and social environment) has also been acknowledged and analyzed in literature (e.g., Gan et al., 2017; Luo and Tang, 2019; Xiang et al., 2015). Customer characteristics such as age or personality (e.g., Karumur et al., 2016; Radojevic et al., 2017) and item characteristics such as the cuisine of a restaurant (e.g., Liu et al., 2017b; Radojevic et al., 2017) may also be significant factors for customer assessments and should be taken into account as well. Developing a unified model which is based on the concepts and methods from Section 3.2 but includes all of the mentioned perspectives as interpretable features may facilitate a much more thorough understanding of online customer reviews and star ratings. This understanding can subsequently be leveraged to improve data-driven organizational decision-making, for instance by revealing which aspects are crucial for customer satisfaction in a certain context.

In regard to the third focal point, the concepts and methods developed in the dissertation have contributed to a comprehensive approach for an automated planning of process models. In particular, the proposed approach for the automated construction of parallelizations (Section 4.1) completes the efforts to enable a construction of all "basic" control flow patterns, which capture the elementary aspects of control flow (Migliorini et al., 2011; Russell et al., 2016; van der Aalst et al., 2003): Sequence (e.g., Heinrich et al., 2012), parallel split (Section 4.1), synchronization (Section 4.1), exclusive choice (Heinrich et al., 2009a; Heinrich et al., 2015) and simple merge (Heinrich and Schön, 2016). Hence, further research should pursue the realization of a consolidated approach integrating all of these contributions. This is still a non-trivial task, as existing work partly relies on a planning graph without control flow patterns as input and thus does not fully take into account possible interdependencies between different control flow patterns. For instance, exclusive choices and simple merges may need to be constructed within parallelizations. This issue can be tackled by analyzing sequences of actions in parallelizations commencing in the same belief state. Despite such hurdles, a realized, consolidated approach has the potential to considerably support process modelers in an automated manner and substantially advance business process management, in particular, business process agility.

Moreover, approaches for the automated adaptation of process models (in Section 4.2) and the automated construction of multi-actor process models (in Section 4.3) have been proposed in the dissertation. An interesting direction for further research in this area is to examine whether the concepts and methods presented in Section 4.2 are also feasible to adapt multi-actor process models in an automated manner, and if not, to develop the necessary enhancements. Similarly, the compatibility of the proposed approaches with the automated construction of context-aware process models (Heinrich and Schön, 2015) should be verified (or established), ultimately enabling an automated planning and adaptation of context-aware multi-actor process models based on a suitable conceptual foundation. Such an approach is in line with related work from (web) service selection (Bortlik et al., 2018). The resulting process models are envisioned to assist individual actors in achieving their personal goals from a decision support perspective, regardless of the context they are facing. More generally, this research would allow for improved business process agility and business process management by empowering organizations with an approach to better handle intricate business processes in complex, quickly changing environment.

While the approaches presented in Section 4.1-4.3 (and also the ones proposed by Heinrich et al. (2015), Heinrich and Schön (2015) and Heinrich and Schön (2016)) have been shown to be computationally feasible, the just suggested consolidated approaches may prove to be more challenging in this regard due to combining multiple complexities. This may impede these approaches from scaling up well to large problem sizes. For instance, planning multi-actor process models which are also context-aware may lead to an immense number of belief states. Yet, such difficulties could be alleviated by pursuing heuristic approaches which do not aim to construct complete process models (i.e., the process models would not necessarily contain all feasible paths to goal states). Based on existing related work from AI planning (e.g., Geffner and Bonet,

2013), such a heuristic approach should be designed to intelligently drive the search towards goal states (Marrella, 2017, 2018). For instance, a function could be developed which heuristically measures how much a single action contributes towards reaching a goal state. Based on this, the planning could be conducted so that paths containing promising actions are explored first, aiming to quickly discover paths to goal states.

A further direction for future research in this focal point is the integration of data quality considerations into concepts and methods for the automated planning of process models. Indeed, approaches in existing literature (e.g., Heinrich et al., 2012, 2015; Heinrich and Schön, 2015, 2016) and also the ones presented in Section 4.1-4.3 are defined as if their input data was of perfect quality. However, as indicated by the work in focal point 1 and related literature striving to enhance business process modeling with data quality considerations (Ofner et al., 2012; Rodríguez et al., 2012), this must not always be the case in practice. Yet, poor data quality may adversely affect the approaches' ability to construct correct process models and support business process agility. For instance, preconditions and effects of actions might be erroneously captured (e.g., due to human misconceptions), which may lead to erroneous process models being constructed. In turn, flaws in process models can cause a variety of severe problems such as impeding their execution (Roy et al., 2014). To address this issue, concepts and methods seeking the efficient initiation of multiple planning runs with different preconditions and effects of actions could be developed. Subsequently, the process models resulting from the planning runs could be assessed to select a feasible process model.

Finally and more generally, Section 4.1-4.3 mainly contribute to the process modeling-phase of the BPM lifecycle (e.g., Dumas et al., 2018; vom Brocke and Rosemann, 2015; Wetzstein et al., 2007). However, with AI planning known to be beneficial for various BPM fields (Marrella, 2017, 2018), the developed concepts and methods also offer starting points for research in other phases. For instance, the concepts and methods from Section 4.1 could also be of use for the construction of parallelizations in process mining (extending work of, e.g., Jin et al., 2016; Wen et al., 2009) and in (web) service composition (extending work of, e.g., Meyer and Weske, 2006; Rathore and Suman, 2015) and to check the correctness of parallelizations in process model verification (extending work of, e.g., Weber et al., 2010; Wynn et al., 2009). Similarly, the concepts and methods of Section 4.2 and 4.3 may prove useful for the development of corresponding approaches in other BPM fields, dealing with related tasks in the process implementation-, process execution- and process analysis-phase. The concepts and methods presented in the dissertation should thus be transferred to support the other phases of the BPM lifecycle and advance business process agility in a multi-faceted way.

Of course, future research opportunities in the spirit of the dissertation are not necessarily restricted to the discussed three focal points and can also be viewed from a broader perspective. To give just a single example, the focal points 1 and 2 have emphasized how important the interpretability of results is, which is rarely warranted when employing some of the most popular AI methods such as neural networks (Adadi and Berrada, 2018; Lundberg and Lee, 2017).

For instance, the language model BERT (Devlin et al., 2018) from natural language processing, which is current state-of-the-art for a manifold of tasks such as aspect-based sentiment analysis (Xu et al., 2019) and question answering (Alberti et al., 2019), relies heavily on sophisticated deep learning concepts and methods. For the most part, it remains non-transparent how it achieves its results. Thus, more generally, future work is needed in the quest to make AI methods explainable, their results interpretable and to develop respective information systems artifacts (Lipton, 2018; Montavon et al., 2018; Schneider and Handali, 2019). Besides offering advantages with respect to data-driven decision-making, such research also contributes to preventing unintended consequences of AI use and thus has implications for the social impact and ethics of AI (Makridakis, 2017; Rothenberger et al., 2019; Sharda et al., 2019).

To conclude, the dissertation opens doors to various interesting directions for further research as concepts and methods from AI have the potential to play an increasingly relevant role in modern information systems. The dissertation itself has provided concrete novel concepts and methods in three focal points, supporting organizations in transforming their decision-making and business processes as they face technology-driven developments. Yet, considerable challenges remain to be addressed in this exciting and seminal area, and the need for research is ongoing.

5.3 References

Abel, F., E. Herder, G.-J. Houben, N. Henze and D. Krause (2013). "Cross-system user modeling and personalization on the social web" *User Modeling and User-Adapted Interaction* 23 (2-3), 169–209.

Adadi, A. and M. Berrada (2018). "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)" *IEEE Access* 6, 52138–52160.

Adomavicius, G., J. C. Bockstedt, S. P. Curley and J. Zhang (2018). "Effects of Online Recommendations on Consumers' Willingness to Pay" *Information Systems Research (ISR)* 29 (1), 84–102.

Agarwal, A., B. Xie, I. Vovsha, O. Rambow and R. Passonneau (2011). "Sentiment analysis of twitter data". In: *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pp. 30–38.

Aggarwal, C. C. (2007). *Data streams: models and algorithms:* Springer Science & Business Media.

Aggarwal, C. C. (2010). *Managing and mining uncertain data:* Springer Science & Business Media.

Ahmad, S., A. Lavin, S. Purdy and Z. Agha (2017). "Unsupervised real-time anomaly detection for streaming data" *Neurocomputing* 262, 134–147.

Alberti, C., K. Lee and M. Collins (2019). "A BERT baseline for the natural questions" *arXiv preprint arXiv:1901.08634*.

Arasu, A., B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava and J. Widom (2016). "STREAM: The stanford data stream management system". In *Data Stream Management*, pp. 317–336: Springer.

Baethge, C., J. Klier, M. Klier and G. Lindner (2017). "Customers' Influence Makes or Breaks Your Brand's Success Story-Quantifying Positive and Negative Social Influence in Online Customer Networks". In: *Proceedings of the 38th International Conference on Information Systems (ICIS 2017)*.

Bandhakavi, A., N. Wiratunga, S. Massie and D. Padmanabhan (2017). "Lexicon generation for emotion detection from text" *IEEE Intelligent Systems* 32 (1), 102–108.

Batini, C., D. Barone, F. Cabitza and S. Grega (2011). "A Data Quality Methodology for Heterogeneous Data" *International Journal of Database Management Systems* 3 (1), 60–79.

Batini, C., C. Cappiello, C. Francalanci and A. Maurino (2009). "Methodologies for data quality assessment and improvement" *ACM Computing Surveys (CSUR)* 41 (3), 16.

Batini, C. and M. Scannapieco (2016). *Data and information quality:* Springer.

Berkovsky, S., T. Kuflik and F. Ricci (2012). "The impact of data obfuscation on the accuracy of collaborative filtering" *Expert Systems with Applications* 39 (5), 5033–5042.

Blake, R. and P. Mangiameli (2011). "The effects and interactions of data quality and problem complexity on classification" *Journal of Data and Information Quality (JDIQ)* 2 (2), 8.

Bobadilla, J., A. Gutiérrez, F. Ortega and B. Zhu (2018). "Reliability quality measures for recommender systems" *Information Sciences* 442, 145–157.

Bortlik, M., B. Heinrich and M. Mayer (2018). "Multi User Context-Aware Service Selection for Mobile Environments" *Business & Information Systems Engineering (BISE)* 60 (5), 415–430.

Bronselaer, A., J. Nielandt and G. de Tré (2018). "An incremental approach for data quality measurement with insufficient information" *International Journal of Approximate Reasoning* 96, 95–111.

Bronselaer, A. and G. de Tré (2016). "A possibilistic treatment of data quality measurement". In: *Proceedings of the 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2016)*, pp. 367–378.

Cai, L. and Y. Zhu (2015). "The challenges of data quality and data quality assessment in the big data era" *Data Science Journal* 14.

Camiciottoli, B. C., S. Ranfagni and S. Guercini (2014). "Exploring brand associations: an innovative methodological approach" *European Journal of Marketing* 48 (5/6), 1092–1112.

Cao, K., G. Wang, D. Han, J. Ning and X. Zhang (2015). "Classification of uncertain data streams based on extreme learning machine" *Cognitive Computation* 7 (1), 150–160.

Chatterjee, S. (2019). "Explaining customer ratings and recommendations by combining qualitative and quantitative user generated contents" *Decision Support Systems (DSS)* 119, 14–22.

Dalip, D. H., M. A. Gonçalves, M. Cristo and P. Calado (2017). "A general multiview framework for assessing the quality of collaboratively created content on web 2.0" *Journal of the Association for Information Science and Technology* 68 (2), 286–308.

Däs, M., J. Klier, M. Klier, G. Lindner and L. Thiel (2017). "Customer lifetime network value: customer valuation in the context of network effects" *Electronic Markets (EM)* 27 (4), 307–328.

De Pessemier, T., S. Dooms, T. Deryckere and L. Martens (2010). "Time dependency of data quality for collaborative filtering algorithms". In: *Proceedings of the 4th ACM RecSys*.

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova (2018). "BERT: Pre-training of deep bidirectional transformers for language understanding" *arXiv preprint arXiv:1810.04805*.

Dey, L., S. M. Haque, A. Khurdiya and G. Shroff (2011). "Acquiring competitive intelligence from social media". In: *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*.

Diao, Q., M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang and C. Wang (2014). "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Dumas, M., M. La Rosa, J. Mendling and H. A. Reijers (2018). *Fundamentals of business process management (2nd edition):* Springer.

Ekman, P. (1992). "An argument for basic emotions" *Cognition & Emotion* 6 (3-4), 169–200.

Fan, W. (2015). "Data quality: from theory to practice" *ACM SIGMOD Record* 44 (3), 7–18.

Färber, M., F. Bartscherer, C. Menne and A. Rettinger (2018). "Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago" *Semantic Web* 9 (1), 77–129.

Feldman, M., A. Even and Y. Parmet (2018). "A methodology for quantifying the effect of missing data on decision quality in classification problems" *Communications in Statistics-Theory and Methods* 47 (11), 2643–2663.

Fisher, C. W., Lauria, Eitel J. M. and C. C. Matheus (2009). "An Accuracy Metric: Percentages, randomness, and probabilities" *Journal of Data and Information Quality (JDIQ)* 1 (3), 1–21.

Gama, J. (2010). *Knowledge discovery from data streams:* Chapman and Hall/CRC.

Gan, Q., B. H. Ferns, Y. Yu and L. Jin (2017). "A text mining and multidimensional sentiment analysis of online restaurant reviews" *Journal of Quality Assurance in Hospitality & Tourism* 18 (4), 465–492.

Gandomi, A. and M. Haider (2015). "Beyond the hype: Big data concepts, methods, and analytics" *International Journal of Information Management* 35 (2), 137–144.

Geffner, H. and B. Bonet (2013). "A concise introduction to models and methods for automated planning" *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8 (1), 1–141.

Ghasemaghaei, M. and G. Calic (2019). "Does big data enhance firm innovation competency? The mediating role of data-driven insights" *Journal of Business Research (JBR)* 104, 69–84.

Ghiassi, M., J. Skinner and D. Zimbra (2013). "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network" *Expert Systems with Applications* 40 (16), 6266–6282.

Giatsoglou, M., M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis and K. C. Chatzisavvas (2017). "Sentiment analysis leveraging emotions and word embeddings" *Expert Systems with Applications* 69, 214–224.

Gomez-Uribe, C. A. and N. Hunt (2016). "The netflix recommender system: Algorithms, business value, and innovation" *ACM Transactions on Management Information Systems (TMIS)* 6 (4), 13.

Hazen, B. T., F. K. Weigel, J. D. Ezell, B. C. Boehmke and R. V. Bradley (2017). "Toward understanding outcomes associated with data quality improvement" *International Journal of Production Economics* 193, 737–747.

He, W., H. Wu, G. Yan, V. Akula and J. Shen (2015). "A novel social media competitive analytics framework with sentiment benchmarks" *Information & Management* 52 (7), 801–812.

He, W., S. Zha and L. Li (2013). "Social media competitive analysis and text mining: A case study in the pizza industry" *International Journal of Information Management* 33 (3), 464–472.

Heinrich, B., M. Bolsinger and M.-A. Bewernik (2009a). "Automated planning of process models: the construction of exclusive choices". In *Proceedings of the 30th International Conference on Information Systems (ICIS 2009)*.

Heinrich, B., M. Hopf, D. Lohninger, A. Schiller and M. Szubartowicz (2019). "Data quality in recommender systems: The impact of completeness of item content data on prediction accuracy of recommender systems" *Electronic Markets (EM)*, to appear.

Heinrich, B. and D. Hristova (2014). "A Fuzzy Metric for Currency in the Context of Big Data". In: *Proceedings of the 22nd European Conference on Information Systems (ECIS 2014)*.

Heinrich, B. and M. Klier (2015). "Metric-based data quality assessment—Developing and evaluating a probability-based currency metric" *Decision Support Systems (DSS)* 72, 82–96.

Heinrich, B., M. Klier and M. Kaiser (2009b). "A procedure to develop metrics for currency and its application in CRM" *Journal of Data and Information Quality (JDIQ)* 1 (1), 5.

Heinrich, B., M. Klier and S. Zimmermann (2012). "Automated Planning of Process Models –Towards a Semantic-based Approach". In *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pp. 169–194: IGI Global.

Heinrich, B., M. Klier and S. Zimmermann (2015). "Automated planning of process models. Design of a novel approach to construct exclusive choices" *Decision Support Systems (DSS)* 78, 1–14.

Heinrich, B. and D. Schön (2015). "Automated Planning of Context-aware Process Models". In: *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*.

Heinrich, B. and D. Schön (2016). "Automated Planning of Process Models: The Construction of Simple Merges". In: *Proceedings of the 24th European Conference on Information Systems (ECIS 2016)*.

Hristova, D. (2014). "Considering Currency in Decision Trees in the Context of Big Data". In: *Proceedings of the 35th International Conference on Information Systems (ICIS 2014)*.

Hristova, D. (2016). “Quantitative Approaches for Modeling Information Quality in Information Systems”. Dissertation. University of Regensburg.

Jabr, W., Y. Cheng, K. Zhao and S. Srivastava (2018). “What Are They Saying? A Methodology for Extracting Information from Online Reviews”. In: *Proceedings of the 39th International Conference on Information Systems (ICIS 2018)*.

Jin, T., J. Wang, Y. Yang, L. Wen and K. Li (2016). “Refactor business process models with maximized parallelism” *IEEE Transactions on Services Computing* 9 (3), 456–468.

Jo, Y. and A. H. Oh (2011). “Aspect and sentiment unification model for online review analysis”. In: *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*.

Karumur, R. P., T. T. Nguyen and J. A. Konstan (2016). “Exploring the value of personality in predicting rating behaviors: A study of category preferences on movielens”. In: *Proceedings of the 10th ACM RecSys*.

Kavzoglu, T. (2009). “Increasing the accuracy of neural network classification using refined training data” *Environmental Modelling & Software* 24 (7), 850–858.

Kiefer, C. (2016). “Assessing the Quality of Unstructured Data: An Initial Overview”. In: *Proceedings of the Lernen.Wissen.Daten.Analysen. 2016 (LWDA 2016)*, pp. 62–73.

Kiefer, C. (2019). “Quality indicators for text data”. In: *Proceedings of the BTW2019 - Datenbanksysteme für Business, Technologie und Web*.

Klein, B. D. and D. F. Rossin (1999). “Data quality in neural network models: effect of error rate and magnitude of error on predictive accuracy” *Omega* 27 (5), 569–582.

Klier, J., M. Klier, F. Probst and L. Thiel (2014). “Customer lifetime network value”. In: *Proceedings of the 35th International Conference on Information Systems (ICIS 2014)*.

Kulkarni, S., N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy and S. Taneja (2015). “Twitter heron: Stream processing at scale”. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*.

Li, S. S. and E. Karahanna (2015). “Online recommendation systems in a B2C E-commerce context: a review and future directions” *Journal of the Association for Information Systems (JAIS)* 16 (2), 72.

Lin, C. and Y. He (2009). “Joint sentiment/topic model for sentiment analysis”. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*.

Linshi, J. (2014). *Personalizing Yelp star ratings: A semantic topic modeling approach.* Yelp Dataset Challenge Winner. URL: https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf (visited on 06/24/2019).

Lipton, Z. C. (2018). “The Mythos of Model Interpretability” *Communications of the ACM* 61 (10), 36–43.

Liu, B. (2015). *Uncertainty theory:* Springer.

Liu, Y., P.-y. Chen, Y. Hong and G. Yong (2017a). “The impact of rating system design on opinion sharing”. In: *Proceedings of the 38th International Conference on Information Systems (ICIS 2017)*.

Liu, Y., T. Teichert, M. Rossi, H. Li and F. Hu (2017b). “Big data for big insights: Investigating language-specific drivers of hotel satisfaction with 412,784 user-generated reviews” *Tourism Management* 59, 554–563.

Lops, P., M. de Gemmis and G. Semeraro (2011). “Content-based recommender systems: State of the art and trends”. In *Recommender Systems Handbook*, pp. 73–105: Springer.

Loshin, D. (2010). *The practitioner's guide to data quality improvement:* Morgan Kaufmann.

Lu, J., D. Wu, M. Mao, W. Wang and G. Zhang (2015). “Recommender system application developments: a survey” *Decision Support Systems (DSS)* 74, 12–32.

Lundberg, S. M. and S.-I. Lee (2017). “A unified approach to interpreting model predictions”. In: *Proceedings of the Neural Information Processing Systems Conference (NIPS 30)*.

Luo, Y. and R. L. Tang (2019). “Understanding hidden dimensions in textual reviews on Airbnb: An application of modified latent aspect rating analysis (LARA)” *International Journal of Hospitality Management* 80, 144–154.

Ma, J., Le Sun, H. Wang, Y. Zhang and U. Aickelin (2016). “Supervised anomaly detection in uncertain pseudoperiodic data streams” *ACM Transactions on Internet Technology (TOIT)* 16 (1), 4.

Makridakis, S. (2017). “The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms” *Futures* 90, 46–60.

Marrella, A. (2017). “What automated planning can do for business process management”. In: *Proceedings of the 15th International Conference on Business Process Management*.

Marrella, A. (2018). “Automated Planning for Business Process Management” *Journal on Data Semantics* 8 (2), 1–20.

McAuley, J. and J. Leskovec (2013). “Hidden factors and hidden topics: understanding rating dimensions with review text”. In: *Proceedings of the 7th ACM RecSys*.

Melville, P. and V. Sindhwani (2017). “Recommender systems” *Encyclopedia of Machine Learning and Data Mining*, 1056–1066.

Meyer, H. and M. Weske (2006). “Automated service composition using heuristic search”. In *Business process management*, pp. 81–96: Springer.

Migliorini, S., M. Gambini, M. La Rosa and A. H. M. ter Hofstede (2011). “Pattern-based evaluation of scientific workflow management systems”. Technical Report. Queensland University of Technology.

Mohammad, S. M. and P. D. Turney (2013). “Crowdsourcing a word-emotion association lexicon” *Computational Intelligence* 29 (3), 436–465.

Montavon, G., W. Samek and K.-R. Müller (2018). “Methods for interpreting and understanding deep neural networks” *Digital Signal Processing* 73, 1–15.

Ofner, M. H., B. Otto and H. Österle (2012). “Integrating a data quality perspective into business process management” *Business Process Management Journal (BPMJ)* 18 (6), 1036–1067.

Ozsoy, M. G., F. Polat and R. Alhajj (2015). “Modeling individuals and making recommendations using multiple social networks”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

Pak, A. and P. Paroubek (2010). “Twitter as a corpus for sentiment analysis and opinion mining”. In: *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC 2010)*.

Picault, J., M. Ribiere, D. Bonnefoy and K. Mercer (2011). “How to get the recommender out of the lab?”. In *Recommender Systems Handbook*, pp. 333–365: Springer.

Plutchik, R. (2001). “The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice” *American Scientist* 89 (4), 344–350.

Poria, S., E. Cambria and A. Gelbukh (2016). “Aspect extraction for opinion mining with a deep convolutional neural network” *Knowledge-Based Systems* 108, 42–49.

Power, D. J., R. Sharda and F. Burstein (2015). *Decision support systems:* John Wiley & Sons.

Radojevic, T., N. Stanisic and N. Stanic (2017). “Inside the rating scores: a multilevel analysis of the factors influencing customer satisfaction in the hotel industry” *Cornell Hospitality Quarterly* 58 (2), 134–164.

Rana, T. A. and Y.-N. Cheah (2016). “Aspect extraction in sentiment analysis: comparative analysis and survey” *Artificial Intelligence Review* 46 (4), 459–483.

Rathore, M. and U. Suman (2015). “An Inheritance based Service Execution Planning Approach using Bully Election Algorithm”. In: *Proceedings of the International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*.

Rettig, L., M. Khayati, P. Cudré-Mauroux and M. Piórkowski (2019). “Online anomaly detection over big data streams”. In *Applied Data Science*, pp. 289–312: Springer.

Rodríguez, A., A. Caro, C. Cappiello and I. Caballero (2012). “A BPMN extension for including data quality requirements in business process modeling”. In: *Proceedings of the International Workshop on Business Process Modeling Notation*, pp. 116–125.

Rosenthal, S., N. Farra and P. Nakov (2017). “SemEval-2017 task 4: Sentiment analysis in Twitter”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*.

Rothenberger, L., B. Fabian and E. Arunov (2019). “Relevance of ethical guidelines for artificial intelligence - a survey and evaluation”. In: *Proceedings of the 27th European Conference on Information Systems (ECIS 2019)*.

Roy, S., A. S.M. Sajeev, S. Bihary and A. Ranjan (2014). “An empirical study of error patterns in industrial business process models” *IEEE Transactions on Services Computing* 7 (2), 140–153.

Russell, N., W. M. P. van der Aalst and A. H. M. ter Hofstede (2016). *Workflow Patterns. The Definitive Guide:* MIT Press.

Sar Shalom, O., S. Berkovsky, R. Ronen, E. Ziklik and A. Amihood (2015). “Data quality matters in recommender systems”. In: *Proceedings of the 9th ACM RecSys*.

Schneider, J. and J. P. Handali (2019). “Personalized Explanation for Machine Learning: A Conceptualization”. In: *Proceedings of the 27th European Conference on Information Systems (ECIS 2019)*.

Schubert, E., A. Koos, T. Emrich, A. Züfle, K. A. Schmid and A. Zimek (2015). "A framework for clustering uncertain data" *Proceedings of the VLDB Endowment* 8 (12), 1976–1979.

Sharda, R., A. Gupta, J. Marsden, P. Brey and I. Heimbach (2019). "Understanding the dark side of analytics: Primum non nocere!". In: *Proceedings of the 27th European Conference on Information Systems (ECIS 2019)*.

Siering, M., A. V. Deokar and C. Janze (2018). "Disentangling consumer recommendations: Explaining and predicting airline recommendations based on online reviews" *Decision Support Systems (DSS)* 107, 52–63.

Sivarajah, U., M. M. Kamal, Z. Irani and V. Weerakkody (2017). "Critical analysis of Big Data challenges and analytical methods" *Journal of Business Research (JBR)* 70, 263–286.

Strapparava, C., A. Valitutti and others (2004). "Wordnet affect: an affective extension of wordnet". In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*.

Tirunillai, S. and G. J. Tellis (2014). "Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation" *Journal of Marketing Research* 51 (4), 463–479.

van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns" *Distributed and Parallel Databases* 14 (1), 5–51.

vom Brocke, J. and M. Rosemann (eds.) (2015). *Handbook on business process management 1: Introduction, methods, and information systems:* Springer Publishing Company, Incorporated.

Wand, Y. and R. Y. Wang (1996). "Anchoring data quality dimensions in ontological foundations" *Communications of the ACM* 39 (11), 86–95.

Wang, H., Y. Lu and C. Zhai (2010). "Latent aspect rating analysis on review text data: a rating regression approach". In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Wang, R. Y. (1998). "A product perspective on total data quality management" *Communications of the ACM* 41 (2), 58–66.

Wang, R. Y. and D. M. Strong (1996). "Beyond accuracy: What data quality means to data consumers" *Journal of Management Information Systems (JMIS)*, 5–33.

Wang, S., Z. Chen and B. Liu (2016). "Mining aspect-specific opinion using a holistic lifelong topic model". In: *Proceedings of the 25th International Conference on World Wide Web*.

Weber, I., J. Hoffmann and J. Mendling (2010). "Beyond soundness: on the verification of semantic business process models" *Distributed and Parallel Databases* 27 (3), 271–343.

Wen, L., J. Wang, W. M. P. van der Aalst, B. Huang and J. Sun (2009). "A novel approach for process mining based on event types" *Journal of Intelligent Information Systems* 32 (2), 163–190.

Wetzstein, B., Z. Ma, A. Filipowska, M. Kaczmarek, S. Bhiri, S. Losada, J.-M. Lopez-Cob and L. Cicurel (2007). "Semantic Business Process Management: A Lifecycle based Requirements Analysis". In: *Proceedings of the Workshop on Semantic Business Process and*

Product Lifecycle Management (SBPM 2007) in conjunction with the 3rd European Semantic Web Conference (ESWC 2007).

Wienand, D. and H. Paulheim (2014). "Detecting incorrect numerical data in dbpedia". In: *Proceedings of the 11th European Semantic Web Conference (ESWC 2014)*, pp. 504–518.

Woodall, P., A. Borek, J. Gao, M. A. Oberhofer and A. Koronios (2014). "An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics". In: *Proceedings of the 19th International Conference on Information Quality (ICIQ 2014)*.

Wynn, M. T., H. M.W. Verbeek, W. M. P. van der Aalst, A. H. M. ter Hofstede and D. Edmond (2009). "Business process verification-finally a reality!" *Business Process Management Journal (BPMJ)* 15 (1), 74–92.

Xiang, Z., Z. Schwartz, J. H. Gerdes Jr and M. Uysal (2015). "What can big data and text analytics tell us about hotel guest experience and satisfaction?" *International Journal of Hospitality Management* 44, 120–130.

Xu, H., B. Liu, L. Shu and P. S. Yu (2019). "BERT post-training for review reading comprehension and aspect-based sentiment analysis" *arXiv preprint arXiv:1904.02232*.

Yadollahi, A., A. G. Shahraki and O. R. Zaiane (2017). "Current state of text sentiment analysis from opinion to emotion mining" *ACM Computing Surveys (CSUR)* 50 (2), 25.

Zadeh, L. A. (1965). "Fuzzy sets" *Information and Control* 8 (3), 338–353.

Zadeh, L. A. (1986). "Is probability theory sufficient for dealing with uncertainty in AI: A negative view". In *Machine Intelligence and Pattern Recognition*, pp. 103–116: Elsevier.

Zak, Y. and A. Even (2017). "Development and evaluation of a continuous-time Markov chain model for detecting and handling data currency declines" *Decision Support Systems (DSS)* 103, 82–93.

Zhang, G. P. (2006). "Avoiding pitfalls in neural network research" *IEEE Transactions on Systems, Man, and Cybernetics* 37 (1), 3–16.

Zimmermann, H.-J. (2011). *Fuzzy set theory—and its applications:* Springer Science & Business Media.