Concepts and Methods for Decision Support

Contributions to Item and Service Recommendations for Users in Web and Mobile Environments

DISSERTATION

zur Erlangung des Grades eines Doktors

der Wirtschaftswissenschaft

eingereicht an der

Fakultät für Wirtschaftswissenschaften

der Universität Regensburg

vorgelegt von

Daniel Lohninger, M.Sc. with Honors

Berichterstatter:

Prof. Dr. Bernd Heinrich

Prof. Dr. Mathias Klier

Tag der Disputation: 05.06.2024

Acknowledgements

First and foremost, I would like to express my deep gratitude to Prof. Dr. Bernd Heinrich, who has supported me at all times over the years and enriched my thesis with valuable advice, detailed feedback, fruitful discussions and encouraged me to a structured and precise way of working. Second, I would like to express my gratitude to Prof. Dr. Mathias Klier for the guidance and support for my dissertation over the past years.

Furthermore, I would like to thank my co-authors, colleagues and in the meantime friends Michael Bortlik, Marcus Hopf, Alexander Schiller and Michael Szubartowicz for inspiring discussions, interesting solution ideas and cooperation based on trust at all times.

Finally, I would like to thank my family and my wife from the bottom of my heart for all the support over the many years and for always believing in me - even in difficult times.

Thank you!

April 2024 *Daniel Lohninger*

Contents

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

1 Introduction

In this section, first, the motivation for the dissertation is presented. Subsequently, the main topics of the dissertation and the associated research questions and approaches are discussed. Finally, the structure of the dissertation and details of each individual paper are provided.

1.1 Motivation

1.1.1 The Rising Availability of Data

In recent years, the volume and diversity of digital data has increased massively (Hashem et al. 2015), which is in line with future predictions showing that the volume of digital data will more than double by 2027 and thus the yearly data volume consists of 284 zettabytes (Statista 2023). This increase is also characterized by the rising popularity of developments in *web environments* such as social media, the internet of things and multimedia (Betty Jane and Ganesh 2020; Hashem et al. 2015) and is further reinforced by the growing importance of mobile technologies (such as mobile devices) and mobile business (Muhammad et al. 2018; Statista 2019) in *mobile environments*.

The resulting data of this developments are often termed as “Big Data” (Saxena and Lamest 2018), which is usually classified by the “5Vs”: Volume, variety, velocity, veracity and value (Abbasi et al. 2016; Anuradha 2015). *Volume* displays the size of the data. *Variety* describes the diversity of different file structures such as structured, semi-structured and unstructured. *Velocity* represents the speed at which data is created or modified. *Veracity* refers to the credibility and reliability of data. Finally, v*alue* represents content-related insights of data such as for users.

The large amount of processed data in web environments can be illustrated by the following examples: TripAdvisor, as the world’s largest travel web platform, provides reviews and recommendations of nearly eight million businesses (e.g., restaurants, hotels, activities; i.e., *items*; cf. TripAdvisor LLC 2024), whereas amazon already offered around 229 million products for sale in Germany in 2016 (Statista 2017). Furthermore, the development of mobile technologies in mobile environments is leading to a rapid increase in mobile services as well as mobile applications, which can relate, for example, to the following areas: Transaction (e.g., banking, shopping and auctions), communication (e.g., email and instant messaging) or information services (e.g., navigation, traffic and tourist guides). (Mayer 2019) As a result, users in mobile environments have almost unlimited access to services (Deng et al. 2016).

These developments require specific concepts and methods to extract valuable information for users out of this high volume of data (Mauro et al. 2015).

1.1.2 The Role of Decision Support

The large volume and diversity of digital data leads to decisions becoming more difficult. This is due to so-called *decision stressors* (i.e., information overload, time pressure, complexity, uncertainty). Decision stressors leads to *stress* (i.e., feeling of overcoming personal resources) and impact the *decision quality* (i.e., performance of a decision) mostly in a negative way. (Phillips-Wren and Adya 2020) Figure 1-1 briefly illustrates the relationship between decision stressors, stress and decision quality.

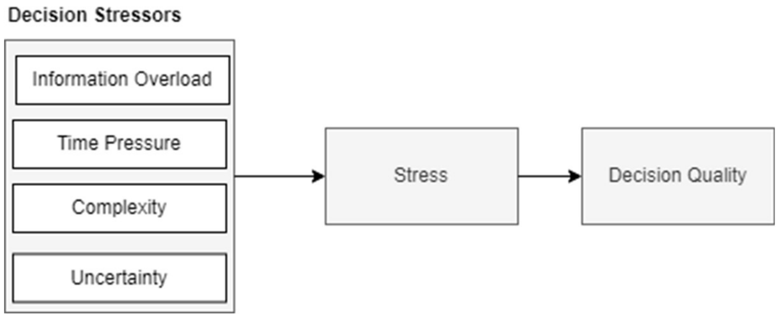

Figure 1-1 *Impact of Decision Stressors on Decision Quality (Phillips-Wren and Adya 2020)*

The two decision stressors *information overload* and *uncertainty* relevant to this dissertation are described in detail in the following.

(I): Users are often faced with the problem of *information overload* (Saxena and Lamest 2018). Information overload describes the phenomenon regarding an individual's ability to make effective decisions appropriately when more information is available than the individual can process (Edmunds and Morris 2000). It is evident that this decision stressor can reduce the decision quality (Phillips-Wren and Adya 2020).

(II): As the volume, variety and velocity of digital data increases, so does the *uncertainty*, leading to a lack of user confidence in the resulting analytical processes and the decisions made thereof (Hariri et al. 2019). Generally, "uncertainty is a situation which involves unknown or imperfect information" (Knight 1921). For instance, most of the attribute values of big data are missing due to noise and incompleteness (Hariri et al. 2019). Moreover, changes in context information (e.g., change of the current user location, cf. Ilarri et al. 2015) also lead to unpredictable events, which further increases uncertainty (Dinh et al. 2020).

Concepts and methods for decision support help users to ease these decision stressors and thus support decision-making (Power et al. 2015). To this end, decision aids such as *decision support systems* have been proposed to mitigate the effects of decision stressors (Phillips-Wren and Adya 2020), resulting in a significant increase in the decision quality (Rötzel and Fehrenbacher 2019). Decision support systems are widely used in various domains such as manufacturing, production scheduling, process optimization, medicine or e-commerce (Liu and Zaraté 2014). The aim of these systems is among other things to extract suitable *items* or *services* for users from a wide range of data (Alter 2012). Two important categories of decision support systems are relevant in this dissertation and described in the following.

(I): Recommender systems are an important category of decision support systems (Power et al. 2015) and one of the most accepted applications of big data (Hu et al. 2020). As a form of *information filtering*, a recommender system enables predictions based on user preferences (Konstan and Riedl 2012). These systems are used for decision support in a wide variety of use

cases in different domains such as the recommendation on e-commerce web platforms for products, movies, songs, books, restaurants or hotels (i.e., items, cf. Hu et al. 2020). Therefore, recommender systems have long been in the focus of research and practice.

(II): Service systems can also be used in the broader context of decision support systems (i.e., service-oriented decision support systems, cf. Demirkan and Delen 2013; Heinrich et al. 2015). In this regard, service systems enable the realization and support of individual user processes by selecting the optimal services (i.e., *service selection*) within a high number of available services (Moghaddam and Davis 2014). Thus, selecting services for processes in mobile environments often results in a decision problem of high complexity (Heinrich and Mayer 2018). Such decision problems can be found in a variety of domains such as healthcare (Ventola 2014) or tourism (Neville et al. 2016).

The following illustration (cf. Figure 1-2) summarizes the relationship between big data (cf. Section 1.1.1) and decision support (cf. Section 1.1.2) and thus forms the theoretical foundation for the underlying concepts and methods of this dissertation (cf. Section 1.1.3, 1.1.4 and 1.2).

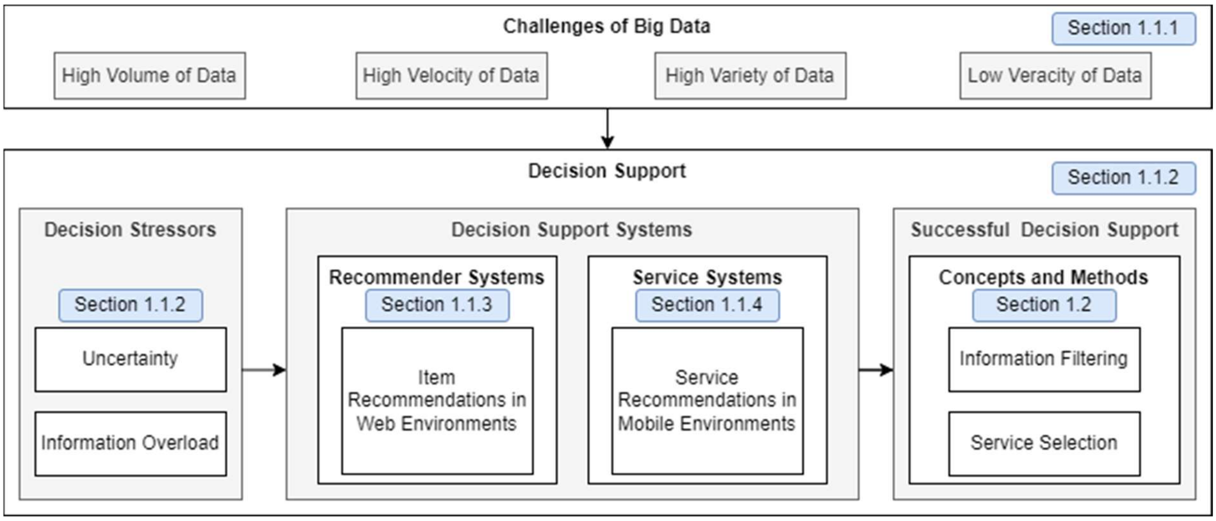

Figure 2-2 *Relationship between Big Data and Decision Support*

In the following, Section 1.1.3 briefly describes the theoretical foundation of recommender systems for recommending items in web environments, whereas Section 1.1.4 shortly presents the state of the art of service systems for recommending services in mobile environments. Selecting the appropriate items and services for the user from a wide range of data, while taking information overload and uncertainty into account, requires new concepts and methods to support successful decision support (cf. Section 1.2).

1.1.3 Item Recommendations in Web Environments

Recommender systems as a form of information filtering systems (Konstan and Riedl 2012; Mehta and Rana 2017) help users to mitigate the information overload problem by filtering the relevant information from a countless amount of data in web environments (Mehta and Rana 2017). In this regard, recommender systems enable successful decision support for each individual user. There are a variety of algorithms for recommender systems, especially with the focus on constantly improving the performance of a recommender system (i.e., recommendation quality; cf. Madadipouya and Chelliah 2017). In general, recommender algorithms enable an estimation for the target user as a list of item preferences and use for this purpose the relationship between neighbouring users as well as the history of user ratings. Recommender systems can be divided into content-based, collaborative filtering and hybrid algorithms and therefore have either a user-based and/or an item-based approach. (Patel and Patel 2020) In this regard, matrix factorization or deep neural network approaches became very popular in recent years (Mehta and Rana 2017; Mu 2018).

The development of big data and the associated decreasing *veracity* of data leads to a continuous increase in data quality problems (Saha and Srivastava 2014). Furthermore, research in the field of data quality has shown that there is a growing tendency to investigate the impact of data quality on the decision quality of decision support systems (Feldman et al. 2018; Woodall et al. 2015). As recommender systems are data-driven tools (Bunnell et al. 2019), examining the quality of data and its impact on recommendation quality seems particularly promising. In this regard, data quality is a multidimensional construct comprising several dimensions such as accuracy, completeness or currency (Batini and Scannapieco 2016; Pipino et al. 2002; Wand and Wang 1996). Recommender systems face major challenges in terms of data quality. Central data quality problems in the context of recommender systems are data sparsity and the cold-start problem (Patel and Patel 2020). In this regard, data sparsity describes the insufficient availability of data to match users for the recommendation (Idrissi and Zellou 2020). Furthermore, the cold-start problem is common with new users because the system has no data about preferences in order to make recommendations (Lika et al. 2014). Both data quality issues have a significant effect on the performance of a recommender system (Patel and Patel 2020).

Thus, one of the main contributions in this dissertation is the topic: *Data Quality in Recommender Systems* and is discussed in more detail in the sections 1.2.1 and 2.

1.1.4 Service Recommendations in Mobile Environments

Service systems in mobile environments have long been part of research and practice. In particular, service systems aim to extract (near-)optimal services for each process step of an individual user process (i.e., selection of a service composition in a user process such as a city day trip in the tourism domain) from a wide range of available services (Moghaddam and Davis 2014). Current state of the art service systems select the (near-)optimal service composition at *planning time* (i.e., before the user process is executed at runtime) due to the high complexity of the underlying decision problem (Moghaddam and Davis 2014). In this regard, service systems enable successful decision support for each individual user. In the literature, mainly *global optimization methods* (i.e., optimal solution; cf. Heinrich and Mayer 2018) but also *heuristic techniques* (i.e., near-optimal solution; cf. Bortlik et al. 2018) were discussed, considering context information (e.g., time of day), multiple users, user preferences (e.g., favourite cuisine) and constraints (e.g., available budget).

In recent years, mobile technologies such as mobile devices have increasingly enabled the recording of various user and environmental contexts via physical sensors such as the current user location, time of day or weather conditions (Mayer 2019). These complex contextual constellations and the associated increasing *uncertainty* in the big data environment (Hariri et al. 2019) lead to unpredictable disruptive events during the execution of a process. A disruptive event is a real-world event at runtime that significantly change planned values (Bearzotti et al. 2012) and therefore has an effect on the selected service composition for all users. In this case, the appropriate re-selection of an alternative service composition is becoming increasingly important. Specifically, modern service systems face major challenges as the consideration of central runtime features (e.g., performance and solution quality) is elementary for users when re-selecting a service composition at runtime (Di Napoli et al. 2021; Wang et al. 2019). Furthermore, proactive user interaction is crucial and consequently has a positive effect on the performance of applications in mobile environments (Böhmann et al. 2014; Demirkan et al. 2015). Thus, the development of heuristics is vital for a high performance of service re-selection at runtime (Heinrich and Mayer 2018; Mayer 2017). In addition to a high-performance re-selection of services, the systematic processing of events in mobile environments is decisive (Flouris et al. 2017). As a result, the literature suggests *fault tolerance strategies* (cf., e.g., Fekih et al. 2019) to avoid the occurrence of events and *event processing strategies* to systematically process events (cf., e.g., Kum 2020). In this respect, disruptive events at runtime are avoided or processed by event handling strategies (cf. Section 3.1) and, if necessary, an alternative service composition is re-selected at runtime by service re-selection algorithms (cf. Section 3.2).

Therefore, the second main contribution in this dissertation is the topic: *Disruptive Events in Service Systems* and is discussed in more detail in the sections 1.2.2 and 3.

1.2 Research Questions and Approach

The dissertation focuses on the two research topics *Data Quality in Recommender Systems* (i.e., Topic 1) and *Disruptive Events in Service Systems* (i.e., Topic 2), as shown in Figure 1-3.

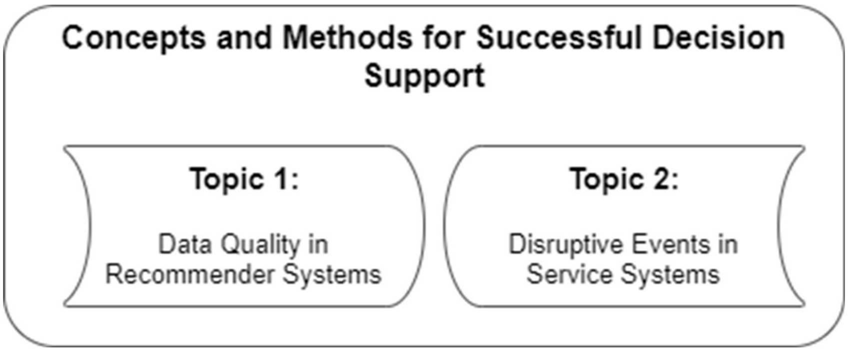

Figure 3-3 *Overview of the two Research Topics in this Dissertation*

Against this background, the purpose of this thesis is to develop concepts and methods for recommender and service systems and thus the further development of decision support systems, considering information overload and uncertainty caused by challenges in the broader context of big data. The research questions and approaches of the main topics are discussed in greater detail in the following.

1.2.1 Topic 1: Data Quality in Recommender Systems

Recommender systems have many practical applications in the field of e-commerce and electronic markets (cf. Li and Karahanna 2015). Especially in e-commerce, providers such as TripAdvisor or Netflix maintain their own individual data sets and thus store a huge amount of data regarding the offered *items* (e.g., restaurants or movies). As discussed in Section 1.1.3, recommender systems suffer from central *data quality problems* such as data sparsity or the cold start problem, which have a significant effect on the performance of a recommender system (Patel and Patel 2020). Similarly, *item content data* often lacks from missing data (i.e., features, e.g., cuisine or actor) or data values (i.e., feature values; cf. Picault et al. 2011) and thus represents a further data quality problem. In this regard, achieving a more complete view on items in relation to the corresponding features (e.g., adding features) and their values (e.g., filling up missing feature values) is of high relevance (Adomavicius and Tuzhilin 2005; Picault et al. 2011). For recommender systems, it is state of the art to increase data quality through the additional integration of external data (Batini et al. 2009; Picault et al. 2011). This provides the opportunity to improve the *completeness* of data via the extension of an item content data set with features and feature values from another item content data set from the same domain (cf. Patel and Patel 2020). Research in the field of decision support systems shows, that an extension of data does not necessarily lead to an improvement of the decision quality (Vanaja and Mukherjee 2019). Therefore, especially for recommender systems, it is extremely important to understand the effects of *increased completeness* on the *impact of recommendation quality* (i.e., prediction accuracy of recommender systems), which leads to the following research questions:

RQ1. *How can an item content data set be systematically extended with respect to the data quality dimension completeness, aiming to improve recommendation quality?*

RQ2. *Does the amount of available item features and the amount of filled up missing item feature values influence the prediction accuracy of recommender systems?*

To address the first research question (i.e., RQ1), we develop a novel procedure for the systematic extension of an item content data set with an additional item content data set from the same domain in the field of decision support systems and in particular recommender systems. The procedure generally consists of the following two steps: *Duplicate Detection* and *Data Integration*. The procedure is evaluated by two real-world scenarios from the restaurant and movie domains, which represent two common areas in web environments especially in e-commerce. The results prove that the presented procedure lead to an increase in recommendation quality. Finally, the resulting effects on recommendation quality are analyzed and discussed.

To address the second research question (i.e., RQ2), we present a theoretical model based on the literature and derive ten hypotheses addressing the impact of additional features and feature values on the prediction accuracy of recommender systems from the perspective of items, users and features. The hypotheses are assessed based on two real-world item content data sets in the restaurant and movie domains. The results of the evaluation show that the prediction accuracy is positively influenced by an increasing amount of features and feature values. In addition, the impact of completeness on the prediction accuracy of a recommender system depends on the number of additional features and their feature values per item or user. Finally, adding features with a high diversity does not necessarily lead to a higher recommendation quality.

1.2.2 Topic 2: Disruptive Events in Service Systems

Service systems are used in many domains in mobile environments such as healthcare or tourism (cf. Gavalas et al. 2014; Ventola 2014). As described in 1.1.4, complex contextual constellations lead to the occurrence of disruptive events at runtime and thus require a re-selection of services at runtime. The underlying decision problem (i.e., Multi-Choice Multi-Dimensional Knapsack Problem (MMKP), cf. Hifi et al. 2004) is NP-hard (non-deterministic polynomial-time hardness; cf. Abu-Khzam et al. 2015) and the occurrence of disruptive events further increases this complexity. In this regard, meta-heuristics such as *local selection* are promising for solving the MMKP and can be used for decision support at runtime (Gendreau and Potvin 2010). Furthermore, the use of world states (i.e., stateful representation, cf. Heinrich and Schön 2015) enable the representation of context information, including contextual changes caused by disruptive events. Thus, the development of a heuristic (based on the meta-heuristic local selection) that is applicable to a stateful representation seems promising for successful decision support. Moreover, the service re-selection at the occurrence of each event is not practicable due to the complexity of the underlying decision problem. Therefore, it is state of the art in mobile environments to systematically process events (cf., e.g., Kum 2020) via event handling strategies (Flouris et al. 2017) in such a way that adaptations of the originally planned service composition are minimized. In this regard, user participation is crucial for the performance of service systems in mobile environments (Ye and Kankanhalli 2020), which leads to the following research questions:

RQ3. *How to develop an event handling strategy for multi user context-aware service systems based on user preferences?*

RQ4. *How to design an optimization-based heuristic technique for re-selecting services under consideration of multiple users, context information and in particular disruptive events at runtime?*

To address the third research question in this dissertation (i.e., RQ3), we develop a user-based event handling strategy for multi user context-aware service systems in mobile environments. In particular, we interviewed 201 participants using an online survey to identify rules and mechanisms based on user preferences, especially for context information and multiple users. As a result, we define a two-stage procedure, which in the first step tries to avoid an event (i.e., fault tolerance) and in the second step processes unavoidable events (i.e., event processing) in such a way that the number of adjustments during service re-selection (cf. RQ4) is minimized.

To address the fourth research question in this dissertation (i.e., RQ4), we present an optimization-based heuristic technique for service re-selection in consideration of multiple users, context-awareness and disruptive events at runtime. In particular, the heuristic adopts a stateful representation that uses a region-based approach to provide the user with alternative solutions in order to support the runtime features performance, solution quality and robustness. Furthermore, novel state space measures and feasibility checks enable to significantly reduce the number of time-consuming re-selection steps. The heuristic is evaluated by a real-world scenario from the tourism domain. The evaluation results show, that the heuristic achieves significantly better results in the evaluation criteria compared to competing artefacts (cf. Bortlik et al. 2018).

1.3 Structure of the Dissertation

The dissertation comprises four research papers addressing the research questions from section 1.2. An overview of these papers including topic, research question, author(s), publication medium and current status can be found in the following Table 1.

Topic	Research Question	Paper	Author(s)	Publication Medium	Current Status
Topic 1: Data Quality in Recommender Systems	RQ1	Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems (Heinrich et al. 2022)	Bernd Heinrich Marcus Hopf Daniel Lohninger Alexander Schiller Michael Szubartowicz	Information Systems Frontiers (ISF)	This paper is **accepted** and **published** in Volume 24, Issue 1 in the journal Information Systems Frontiers (2022, online in 2020)
	RQ2	Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems (Heinrich et al. 2021)	Bernd Heinrich Marcus Hopf Daniel Lohninger Alexander Schiller Michael Szubartowicz	Electronic Markets (EM)	This paper is **accepted** and **published** in Volume 31, Issue 2 in the journal Electronic Markets (2021, online in 2019)
Topic 2: Disruptive Events in Service Systems	RQ3	User-based Event Handling Strategy for Multi User Context-Aware Service Systems in Mobile Environments (Lohninger 2024)	Daniel Lohninger	Internationale Tagung Wirtschaftsinformatik (WI)	This paper **is under review** at the 19th Internationale Tagung Wirtschaftsinformatik (2024)
	RQ4	Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime (Bortlik et al. 2023)	Michael Bortlik Bernd Heinrich Daniel Lohninger	Information Systems Frontiers (ISF)	This paper is **accepted** and **published** in the journal Information Systems Frontiers (2023)

***Table 1** Overview of Papers contained in the Dissertation*

The remainder of the thesis is organized as follows: Subsequent to the introduction, the Sections 2 and 3 describe the main topics with the corresponding four papers and therefore form the main contribution of the dissertation. Finally, section 5 summarizes the major findings, implications and limitations as well as directions for further research.

2 Data Quality in Recommender Systems

2.1 Paper 1: Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems

Current Status	Full Citation
This paper is **accepted** and **published in** Volume 24, Issue 1 in the journal Information Systems Frontiers.	Heinrich, Bernd; Hopf, Marcus; Lohninger, Daniel; Schiller, Alexander; Szubartowicz, Michael (2022): Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems. In Inf Syst Front 24 (1), pp. 267–286. DOI: 10.1007/s10796-020-10071-y.

Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems

Bernd Heinrich

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Marcus Hopf

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Daniel Lohninger

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Alexander Schiller

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Michael Szubartowicz

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Abstract

The rapid development of e-commerce has led to a swiftly increasing number of competing providers in electronic markets, which maintain their own, individual data describing the offered items. Recommender systems are popular and powerful tools relying on this data to guide users to their individually best item choice. Literature suggests that data quality of item content data has substantial influence on recommendation quality. Thereby, the dimension completeness is expected to be particularly important. Herein resides a considerable chance to improve recommendation quality by increasing completeness via extending an item content data set with an additional data set of the same domain. This paper therefore proposes a procedure for such a systematic data extension and analyzes effects of the procedure regarding items, content and users based on real-world data sets from four leading web portals. The evaluation results suggest that the proposed procedure is indeed effective in enabling improved recommendation quality.

1 Introduction

In line with the emergence and proliferation of the internet, e-commerce has developed into a major disruptor for retail business. Indeed, in 2020, retail e-commerce sales worldwide are estimated to hit $4.2 trillion, with its share of global retail reaching 16.1% and rising further to 22% in 2023 (Statista 2019). This rapid development of e-commerce has implied a swiftly increasing number of competing providers in electronic markets (e.g., *Amazon* and *Walmart* in general retail, *Booking.com* and *HRS* in hotel bookings, *Yelp* and *TripAdvisor* in restaurant bookings). Providers – even of the same domain – maintain their own, individual data sets containing information regarding the offered items (e.g., products or services), which usually vary in their attributes (content) to describe even the same items. For instance, *Booking.com* provides detailed data on location score and furniture of hotels, which is not offered by *HRS*. This data as well as the recommender systems commonly present on such e-commerce platforms aim at guiding users to their individually best item choice, improving user stickiness and increasing platform revenue (Zhou 2020). Such supporting systems are mandatory as customers regularly need to make a choice between a plethora of items (e.g., songs, movies, restaurants, hotels) on e-commerce platforms (Kamis et al. 2010; Levi et al. 2012; Richthammer and Pernul 2018; Tang et al. 2017; Vargas-Govea et al. 2011). It is thus hardly surprising that recommender systems in particular have been established as one of the most powerful and popular tools in the field of e-commerce in recent years (Ricci et al. 2015a; Scholz et al. 2017; Smith and Linden 2017).

As recommender systems are data-driven tools, the quality of the data which a recommender system is based on is assessed to be one of the issues recommender systems research is strongly involved with (Bunnell et al. 2019) and may have substantial influence on the resulting recommendations (Picault et al. 2011; Sar Shalom et al. 2015). Here, data quality is a multidimensional construct comprising several dimensions such as accuracy, completeness and currency of data (Batini and Scannapieco 2016; Pipino et al. 2002; Wand and Wang 1996), with each dimension providing a distinct view on data quality (e.g., Heinrich et al. 2018). For recommender systems examining the item content data (attributes and attribute values of items), achieving a more complete view on these items seems to be especially important (Adomavicius and Tuzhilin 2005; Picault et al. 2011), as "some representations capture only certain aspects of the content, but there are many others that would influence a user's experience" (Picault et al. 2011). This means that the data quality dimension completeness is of particular relevance for recommender systems.

Herein resides a considerable chance to improve recommendation quality by increasing completeness via extending an item content data set (e.g., from an e-commerce platform such as *TripAdvisor*) with additional attributes and attribute values from another data set in the same domain (e.g., from an e-commerce platform such as *Yelp*). This opportunity is particularly promising for search portals offering a meta view by compiling information from various platforms (e.g., *trivago.com*), which currently simply juxtapose the data and do not use an extended data set for the application of a recommender system. Yet, how to systematically achieve more complete item content data sets and realize the expected advantages for recommender systems is left unanswered in existing research. Thus, the paper at hand investigates the following research question:

How can an item content data set be systematically extended with respect to the data quality dimension completeness, aiming to improve recommendation quality?

As recommender systems are an important category of decision support systems (Power et al. 2015), this research is in line with recent works which have revealed a significant impact of data quality dimensions, especially completeness, on data-driven decision support systems (e.g., Feldman et al. 2018; Heinrich et al. 2019; Woodall et al. 2015).

The remainder of the paper is organized as follows. In the next section, the general and theoretical background as well as the related work are discussed. Thereafter, a procedure for the systematic extension of an item content data set with attributes and attribute values from another item content data set is presented, providing the basis for determining recommendations. In the fourth section, the proposed procedure is evaluated in two e-commerce real-world scenarios and resulting effects on recommendation quality are analyzed. The final section summarizes the work and discusses limitations as well as directions for future research.

2 Foundation

This section first discusses the positioning of recommender systems in the field of decision support systems in e-commerce as general background of our research. The second part of this section presents a theoretical model regarding the relationship between data quality and decision support systems – especially recommender systems – based on a discussion of existing literature. The third part of the section discusses related work and identifies the research gap addressed by this paper.

2.1 General Background

Recommender systems have become a highly relevant category of decision support systems (Power et al. 2015). In particular, in e-commerce, recommender systems are often necessary as users regularly need to make decisions for purchase, consumption or utilization of items (e.g., songs, movies, restaurants or hotels) from a plethora of possible alternatives available in information systems (IS) on e-commerce platforms (Kamis et al. 2010; Levi et al. 2012; Richthammer and Pernul 2018; Tang et al. 2017; Vargas-Govea et al. 2011).

More precisely, the high number of items together with the high number of users on e-commerce platforms lead to the problem of information overload, which is widely discussed by many researchers in the past decades and thus, constitutes a major subject of IS research in fields such as e-commerce (Lu et al. 2015) or management of business organizations (Edmunds and Morris 2000). In particular, information overload denotes the phenomenon regarding an individual's ability to appropriately cope with solving problems (e.g., making a choice) when more information is available than the individual can assimilate (Edmunds and Morris 2000). This is, users often do not have the skills and experience to adequately evaluate the large number of available alternatives for making their choice (Ricci et al. 2015b; Scholz et al. 2017). The resulting problem leaves users of e-commerce IS unable to make effective decisions due to this large volume of information (e.g., items) to which users are exposed to (Hasan et al. 2018; Lu et al. 2015; Richthammer and Pernul 2018; Scholz et al. 2017). In order to address the problem of information overload, the literature suggests for IS providers in e-commerce to incorporate decision support systems, in particular recommender systems, to assist users in their decision-making (Bunnell et al. 2019; Karimova 2016; Lu et al. 2015). Therefore, recommender systems aim at individually preselecting smaller sets of relevant items for each single user (i.e., information filtering; cf. Lu et al. 2015) to allow for good decision-making in a personalized and comfortable way avoiding to overwhelm the user (Manca et al. 2018).

Here, recommender systems are especially suitable to tackle the information overload problem, since they constitute data-driven systems, which enables them to individually support each user's decision-making in an automated manner (Bunnell et al. 2019; Karumur et al. 2018; Lu et al. 2015). A variety of IS research aims to tackle the information overload problem in the field of e-commerce by developing different approaches for recommender systems (e.g., Content-Based Filtering; cf. Aggarwal 2016; Jannach et al. 2012; Ricci et al. 2015a). In particular, recommender systems process different types of data (e.g., user rating data or item content data) in order to derive the individual users' preferences, which are stored in a user profile, based on data such as users' historical evaluations of other items (cf. Peska and Vojtas 2015; Ricci et al. 2015a). To enable recommendations of high precision, the matching of the user profile against item profiles (i.e., the content data of an item) or against other user profiles is highly relevant (Ricci et al. 2015a). This further emphasizes the key role of data (e.g., item content data) for recommender systems to enable individualized decision support for a large number of users in e-commerce settings (e.g., during shopping experiences on e-commerce websites; cf. Heinrich et al. 2019; Kamis et al. 2010).

In e-commerce, recommender systems not only assist users and make their experience on e-commerce platforms more comfortable, but they also create business value for the IS providers (Bunnell et al. 2019). By integrating recommender systems into a wide variety of e-commerce activities such as browsing, purchasing, rating or reviewing items, the resulting diversity of generated data (e.g., item content data, user rating data or click-stream data) can be used for modeling of user profiles and thus support certain marketing activities such as cross-selling, advertising or product promotion (Karimova 2016; Lu et al. 2015). It is thus hardly surprising that in recent years, recommender systems as data-driven tools have emerged to be among the most frequently applied decision support systems in the field of IS in e-commerce (Ricci et al. 2015a; Scholz et al. 2017; Smith and Linden 2017).

As recommender systems support user choices mainly on the basis of data, it seems promising to investigate how the data quality (e.g., completeness of item content data) influences the quality of recommender systems in the field of e-commerce.

2.2 Theoretical Background

The systematic procedure presented in this paper aims to contribute to further research investigating the relationship between data quality and (data-driven) decision support systems. At first glance, it might seem natural and obvious to suggest that more data always has a positive influence on decision support (especially when provided by a system). However, research in different areas shows that more data does not always lead to better results of decision support systems in general (e.g., when selecting features based on which a decision is obtained; cf. Mladenić and Grobelnik 2003; Vanaja and Mukherjee 2019), as different data sets (e.g., with more or fewer attributes) may lead to varying results of decision support. Thus, the impact of the data quality of data values on different evaluation criteria of decision support systems such as decision quality or data mining outcome has been studied in existing literature (e.g., Bharati and Chaudhury 2004; Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Heinrich et al. 2019; Woodall et al. 2015). Yet, this research neither focuses on how to systematically achieve more complete item content data sets nor on how to define a well-founded procedure, but instead tries to explain the relationship between data quality and evaluation criteria of decision support systems. In this regard, such explanatory models are the theoretical background in data quality research which we aim to support by our work. Thus, this background is briefly discussed in the following.

Bharati and Chaudhury (2004) assess the effects of the data quality dimensions accuracy, completeness and currency on the ability of an online analytical processing system to sustain decision-making. Ge (2009) discusses accuracy, completeness and (Heinrich and Mayer 2018)consistency and their impact on decision quality. Blake and Mangiameli (2011) assess the impact of accuracy, completeness, consistency and currency on data mining results in order to support decision-making in companies. Woodall et al. (2015) analyze the impact of completeness on classification outcomes used for supporting users in their decision process. Feldman et al. (2018) propose an analytical framework to investigate the effects of incomplete data sets on a binary classifier that serves for decision support. Heinrich et al. (2019) examine the impact of the amount of available attributes and attribute values on the prediction accuracy of recommender systems.

Summing up, the focus of these papers is to investigate in which way and to what extent improving the quality of data values, especially the dimension completeness, leads to an improvement in evaluation criteria of particular decision support systems. A relevant and widely used category of decision support systems which assists users facing decision-making problems are recommender systems (Porcel and Herrera-Viedma 2010; Power et al. 2015). Based on this and in line with Heinrich et al. (2019), we refer to the theoretical model for describing the relationship between data quality and decision support systems, presented in Fig. 1.

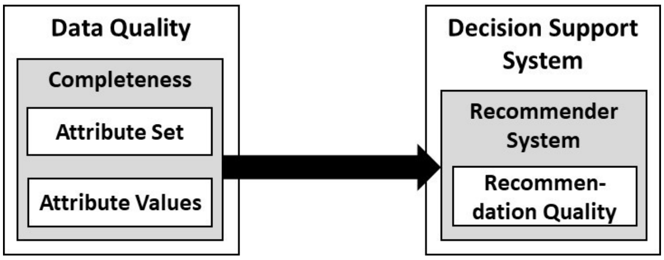

Fig. 1 Theoretical Model (according to Heinrich et al. 2019)

The theoretical model in Fig. 1 indicates a direct relationship between data quality and decision support systems. In particular, the theoretical model suggests this relationship between completeness of item content data (attributes and attribute values) and recommendation quality of recommender systems. With this model as theoretical background, the procedure presented in this paper proposes how to systematically extend items in an item content data set with attributes and attribute values of the same items from a second item content data set in order to gain a more complete view on the considered real-world entities (e.g., movies or restaurants). Thus, this systematic procedure forms the basis for an even more precise and well-founded investigation of the impact of completeness on the recommendation quality of data-driven decision support systems (especially recommender systems) in the future.[1] In particular, it enables theoretical relationships (i.e., similar to Fig. 1) for different data sets to be analyzed in a transparent and comprehensible manner. Furthermore, this procedure can serve as an already evaluated template for future procedures in order to support the investigation of further data quality dimensions (e.g., consistency) in other data-driven decision support systems.

[1] In this regard, an implementation of the procedure is available on GitHub (GitHub 2020).

2.3 Related Work and Research Gap

In this section, we present approaches dealing with data extension in the context of recommender systems and analyze relevant works discussing data quality aspects related to recommender systems.[2] Thereafter, we summarize existing contributions and identify the research gap addressed by this paper.

To prepare the related work, we followed the guidelines of standard approaches (e.g., Levy and Ellis 2006). In particular, we performed a literature search on the databases *ACM Digital Library*, *AIS Electronic Library*, *IEEE Xplore*, *ScienceDirect* and *Springer* as well as the proceedings of the *European and International Conference on Information Systems*, the *International Conference on Information Quality* and the *ACM Conference on Recommender Systems*. Subsequently, we examined whether these works represent relevant approaches for our research by reading title, keywords, abstract and summary and also conducted a forward and backward search in order to find further relevant works. After analyzing the resulting papers in detail, eighteen articles were deemed relevant. These papers could be organized within two separate categories, with each category containing nine works.

(1): The first category of works copes with some kind of data extension in the context of recommender systems. For these works, the effect on decision quality and in particular recommendation quality is vital ("fitness for use"). This is a crucial difference to general approaches for data extension (e.g., in the context of data warehouses), where the effect on decision quality is often unclear or difficult to assess. Although all papers of the first category consider data extension and its effect on recommendation quality, none of the approaches describes the systematic extension of an item content data set with additional data from the same domain in the form of a procedure in the context of recommender systems, which is the contribution of our research. Moreover, the approaches differ in the kind of extended data (1A), the entities extended with data (1B) and in the usage of different methods for data extension (1C).

(1A): Several recent articles focus on the extension of data with data from a distinct area, for example, data from different domains such as music and film (cross-domain data sets; Abel et al. 2013; Ntoutsi and Stefanidis 2016; Ozsoy et al. 2016), context information such as time and location (multi-dimensional data sets; Abel et al. 2013; Kayaalp et al. 2009) or data from different social and semantic web sources such as *Wikipedia*, *Facebook* and *Twitter* (heterogeneous data sets; Abel et al. 2013; Bostandjiev et al. 2012; Chang et al. 2018; Kayaalp et al. 2009; Ozsoy et al. 2016). These approaches examine whether the diversity of data types leads to improved recommendation quality but do not systematically extend item content data with additional data from the same domain.

(1B): Other works in literature analyze user profiles from different social networks (Abel et al. 2013; Li et al. 2018; Ozsoy et al. 2016; Raad et al. 2010). The matching user profiles are merged across different networks to produce a positive effect on recommendation quality. However, these works do not focus on item content data at all.

(1C): Finally, some recent works focus on the extension of item or user data from multiple data sources in the context of recommender systems (Abel et al. 2013; Bostandjiev et al. 2012; Bouadjenek et al. 2018; Ozsoy et al. 2016). These approaches rely on tools such as *BlogCatalog*, *Google Social Graph API*, *Google Search API* or *OpenID*, which provide information for the matching of users or items. However, these works do not focus on describing the systematic extension of an item content data set and instead use external, non-transparent methods for data extension, which severely limits their applicability in other scenarios.

(2): The second category of works explicitly recognizes the importance of data quality for recommender systems (Amatriain et al. 2009; Basaran et al. 2017; Berkovsky et al. 2012; Burke and Ramezani 2011; Heinrich et al. 2019). In particular, Heinrich et al. (2019) examine the impact of the number of available attributes and attribute values on prediction accuracy of recommender systems by testing hypotheses but do not provide a procedure for extending an item content data set with additional attributes and attribute values. Further approaches give rise to concepts that deal with data quality issues in the context of recommender systems. For instance, data sources used by a recommender system can be chosen user-dependently as data sparsity and inaccuracy have been identified to impact recommendation quality (Lathia et al. 2009). Sar Shalom et al. (2015) tackle sparsity and redundancy issues by deleting or omitting certain users or items while Pessemier et al. (2010) analyze consumption data such as

[2] Some approaches for data extension with regard to completeness (e.g., cf. Naumann et al. 2004; Bleiholder and Naumann 2008; Scannapieco and Batini 2004) mainly deal with technical issues (e.g., wrapper architecture, database architecture) or model-oriented aspects (e.g., schema mapping, operators, join approaches), which are not within the scope of this work.

ratings in regard to currency. Further, Levi et al. (2012) use text mining on user reviews from various sources to alleviate the cold start problem of new users by assigning them to so called context groups.
In summary, none of these works provides a systematic procedure for the extension of a data set with item content data of another data set from the same domain. The works in category (1A) focus on the extension with data from a different area, but they do not target on data representing the *same* items, which is a decisive characteristic of our research. The works in category (1B) do not focus on item content data but instead analyze user profiles from various social networks. In contrast to this, we provide a procedure for the matching and extension of *item* content data. The works in category (1C) use existing tools for data extension, especially for user data. Such an extension is non-transparent, highly dependent on these tools as well as the application scenario and does not allow to support the analysis of theoretical relationships (cf. Fig. 1) between different data sets in a verifiable and comprehensible manner. Additionally, no explicit procedure for extending an item content data set with additional attributes and attribute values in detail is given. The works of the second category analyze the impact of data quality on recommender systems. However, only Heinrich et al. (2019) analyze effects of a more complete view on items by data set extension. Yet, this work does not aim to provide a procedure for the extension of item content data in the context of recommender systems. In contrast, the authors present an explanatory analysis based on hypotheses testing. To conclude, none of these approaches presents a systematic procedure for the extension of a data set with item content data of another data set from the same domain.

3 A Procedure for Extending an Item Content Data Set

In this section, we propose a procedure for the systematic extension of a data set in the context of recommender systems, aiming to improve the quality of the resulting recommendations. We discuss and substantiate in detail how to extend a data set DS1 containing items and item attributes from a certain domain (e.g., movies, restaurants or hotels) by using a data set DS2 containing items and item attributes from the same domain.[3] In particular, items in DS1 are extended with attributes and attribute values of the same items from DS2. This means that in a first step *duplicates have to be detected* before in a second step, the *data sets can be actually integrated into one data set*. The exact elaboration of these two steps in the context of recommender systems addresses our research question and thus represents the contribution of this paper. In a subsequent step, the resulting data set extension can be evaluated by *determining recommendations* based on the extended data set and assessing the resulting recommendation quality. Since different existing content-based or hybrid recommender systems can be used for this step, it is not a core element of the procedure. The procedure is illustrated in Fig. 2 and described in the following.

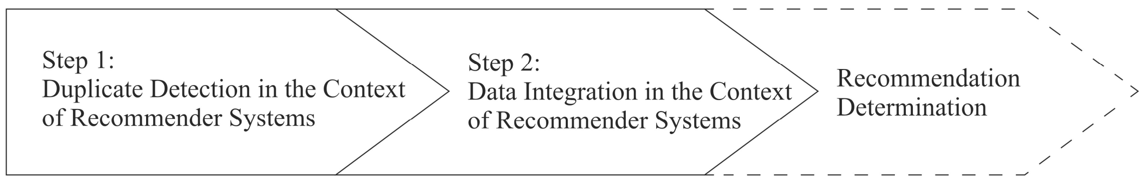

Fig. 2 Procedure to Extend an Item Content Data Set in the Context of Recommender Systems

3.1 Duplicate Detection in the Context of Recommender Systems

An item in a data set DS1 usually has different attributes and attribute values compared to its corresponding duplicate item in a data set DS2 (e.g., because the portals have heterogeneous data policies), making duplicate detection in the context of recommender systems a non-trivial task. Here, duplicate detection is a binary classification of item pairs (one item from DS1 and one item from DS2) with the two classes *duplicate* and *non-duplicate*. Due to a potentially large number of items per data set, duplicate detection should be carried out in a widely automated manner. To assist this task, literature proposes *similarity measure functions* (SMFs; e.g., the Jaro-Winkler function; Winkler 1990) to determine the similarity of *key attributes* (e.g., "Name" and "Geolocation" of a restaurant) between items from DS1 and DS2, with high similarity values indicating possible

[3] If more than two data sets are available, the procedure can be applied iteratively.

duplicates. We propose the following four Tasks 1.1-1.4 to configure and perform duplicate detection, acknowledging peculiarities in the context of recommender systems (cf. Fig. 3).

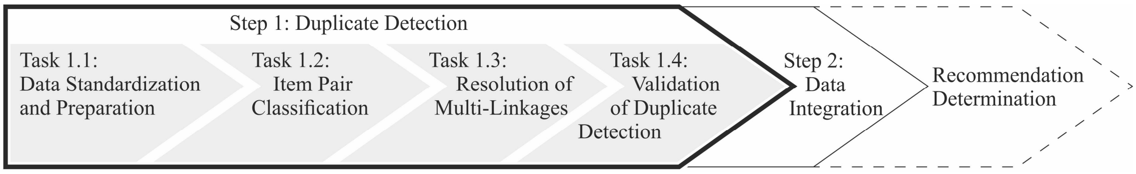

Fig. 3 The Step Duplicate Detection in Detail

In **Task 1.1**, the data for duplicate detection is standardized and prepared. This is necessary because different portals often specify varying values for (key) attributes (e.g., due to heterogeneous data policies). Furthermore, as the data is usually decentrally generated by many different users, these users often enter attribute values on their very own interpretation, leading to data quality problems in e-commerce platforms. These issues make duplicate detection for recommender systems data sets highly complex. For example, one and the same US phone number could be entered as "+1-212-283-1100" in one data set and as "(212) 283-1100" in the other data set. Here, it is clear that a standardization of both phone numbers to "area code: 212, phone number: 2831100" helps determining that these numbers refer to the same phone connection in an automated manner. The standardization of the key attributes can be conducted by utilizing specific parsing tools which standardize the values of the key attributes (e.g., the python package "phonenumbers" for the key attribute "Phone"). After standardization, the values for all key attributes of both data sets DS1 and DS2 are stored in a common standard format. Nevertheless, even after standardization, duplicate items in DS1 and DS2 may differ in key value attributes caused by varying entered values (e.g., "283-100" instead of "283-1100"). Subsequent to standardization, item pairs are prepared for binary classification in the next task. Here, each item from DS1 in combination with each item from DS2 is considered as an item pair. It is clear that most of these pairs are non-duplicates. Therefore, it is beneficial to discard the item pairs which are obvious non-duplicates (e.g., restaurants with a GPS distance larger than 1,000 meters), which is referred to as blocking in literature (Steorts et al. 2014).

Task 1.2 comprises the binary classification of item pairs as duplicates or non-duplicates. In many contexts, this classification can be performed rather easily in a supervised manner. However, in the context of recommender systems, generally, no substantial amount of labeled training data (i.e., item pairs labelled as (non-)duplicates) is available for a supervised classification. Therefore, it is crucial to perform item pair classification in an unsupervised manner, not requiring any labeled training data (cf., e.g., Jurek et al. 2017). In the following, we describe the basic ideas of such an algorithm and emphasize the crucial peculiarities of the algorithm in the context of recommender systems. The algorithm starts with an initialization, followed by the proper classification and ends with all item pairs being classified as duplicate or non-duplicate.

The initialization consists of the selection of SMFs that are used for the classification. For each key attribute available in both data sets DS1 and DS2, adequate SMFs have to be specified. The choice of SMFs primarily depends on the data type of the respective key attribute. In particular, for key attributes containing string values and key attributes containing numerical values, different SMFs have to be used (e.g., the haversine SMF for GPS data values and the Jaro-Winkler SMF for string data values; cf. Table 1). Here, it is important to not only select one SMF per key attribute, but to select multiple SMFs with different characteristics, since the compared values of the key attributes may also exhibit varying deviations and specifications. For string attribute values with different suffixes (e.g., a restaurant is represented by "Fluffy's New York" in DS1 and by "Fluffy's Café & Pizzeria" in DS2), a SMF that focuses on the initial characters of a string such as the Jaro-Winkler SMF is appropriate. Further, for string attribute values with typographical errors (e.g., a restaurant is represented by "Fulffy's" in DS1 and by "Fluffys" in DS2), a SMF addressing this special deviation such as the Levenshtein SMF is suitable. Therefore, it is important to utilize multiple SMFs for item pair classification to cope with the challenges of highly diverse data values in the context of recommender systems. To further elaborate on the specification of SMFs for item pair classification, we give a broader discussion of selected SMFs with different characteristics in Table 1 based on Christen (2012) and state their properties and examples in the context of recommender systems.

The proper classification is then conducted via an unsupervised ensemble self-learning algorithm, which improves results compared to just using the values of SMFs for classification (Jurek et al. 2017). This self-learning algorithm starts with training a certain binary classifier. The training is conducted on a small set of training data, which consists of the item pairs with the highest similarity values (implicitly labeled as duplicates) and item pairs with the lowest similarity values (implicitly labeled as non-duplicates) and thus does not need to be labeled manually. This binary classifier is then used to predict the classes of all other item pairs. The item pairs classified with a high certainty are then added to the training data. Subsequently, the binary classifier is trained again and the steps are gradually repeated until all item pairs are classified as either duplicates or non-duplicates by this certain binary classifier. To further increase the robustness of the classification result, multiple such binary classifiers are used with the described self-learning method and the obtained results are then aggregated to obtain the final stable result of the item pair classification.

Table 1. Selected Similarity Measure Functions and their Application in the Context of Recommender Systems

Similarity measure functions	*Properties*	*Examples in the context of recommender systems*
Levenshtein The Levenshtein SMF is based on the minimum number of edit operations of single characters necessary to transform a string s_1 into a string s_2.	• Appropriate for misspellings/ typographical errors • Inappropriate for truncated/ shortened strings and divergent pre-/suffixes • Complexity: $O(\lvert s_1\rvert * \lvert s_2\rvert)$	Typographical error in the attribute "Restaurant Name": "Fulffy's" vs. "Fluffys".
Jaro The Jaro SMF is based on the number of agreeing characters c contained in the strings s_1 and s_2 within half the length of the longer string, and the number of transpositions t in the set of common substrings.	• Appropriate for misspellings/ typographical errors • Inappropriate for long divergent pre-/suffixes • Complexity: $O(\lvert s_1\rvert + \lvert s_2\rvert)$	Misspelling in the attribute "Restaurant Name": "Fluffy's Café" vs. "Flufy's Café".
Jaro-Winkler The Jaro-Winkler SMF extends the Jaro SMF, putting more emphasis on the beginning of the strings.	• Appropriate for misspellings/typographical errors and divergent suffixes • Inappropriate for long divergent prefixes • Complexity: $O(\lvert s_1\rvert + \lvert s_2\rvert)$	Divergent suffixes of the attribute "Restaurant Name": "Fluffy's New York" vs. "Fluffy's Café & Pizzeria".
Haversine This SMF is based on the haversine formula, which measures the distance between two locations on earth.	• Appropriate for geographical coordinates given in latitude/longitude	"40.711, -73.966" vs. "40.710, -73.965".

In **Task 1.3,** it is necessary to resolve multi-linkages of duplicates resulting from Task 1.2. This problem may arise as an item from DS1 can be contained in more than one item pair classified as a duplicate. Thus, this item from DS1 is linked to more than one item from DS2. Similarly, an item from DS2 can be linked to more than one item from DS1. As the matched items will be proposed to users in the recommendation step, it is important to resolve these multi-linkages of items to avoid redundant and multiple recommendations of individual items. To resolve the multi-linkages, the prediction scores of the ensemble classifier from Task 1.2 are used. Considering an item from DS1 linked to multiple items from DS2, only the linkage with the highest prediction score is retained and all other linkages are discarded. Analogously, only one linkage is kept when an item from DS2 is linked to multiple items from DS1. In this way, the n-to-n reference of items from DS1 and DS2 is firstly reduced to 1-to-n references and then to 1-to-1 references.

Step 1 concludes with the validation of the results of the duplicate detection in **Task 1.4**, which is necessary to assess the quality of the duplicate detection. This quality plays an important role in the context of recommender systems, as false duplicates would result in erroneous data integrations in the next step of the procedure, and thereby, to negative effects on item recommendations. On the other hand, false negatives would result in feasible data integrations not being carried out, thus reducing the benefit of the procedure. Therefore, a small excerpt of item pairs, serving as test data, needs to be labeled as duplicates or non-duplicates for validation purposes. Here, a random selection of item pairs to be labeled would result in the vast majority of these item pairs being labeled as non-duplicates, since most item pairs are indeed non-duplicates. Therefore, it is important to take the calculated values of the SMFs into account and to also label item pairs which are more likely to be a real duplicate. Building on this labeled test data, the number of correct classifications (i.e., “true positives” and “true negatives”) and the number of errors (i.e., “false positives” and “false negatives”) can be determined. Based on these numbers, evaluation metrics such as precision, recall and F1-measure can be assessed. If these evaluation metrics report unsatisfactory results, the classification errors may be analyzed and tackled. The evaluation metrics thus enable to ensure a high quality of the conducted duplicate detection and to provide data suitable for the next step of the procedure, which concludes Task 1.4 and thus Step 1.

3.2 Data Integration in the Context of Recommender Systems

In Step 2 of the procedure, attributes and attribute values of DS1 and DS2 are integrated to obtain the envisioned more complete view on items. In particular, *matching* attributes (i.e., attributes of DS2 also existing in DS1) and *additional* attributes (i.e., attributes only existing in DS2) have to be identified and the items’ attribute values have to be extended. To perform this integration in the context of recommender systems, we propose the following three Tasks 2.1-2.3 (cf. Fig. 4).

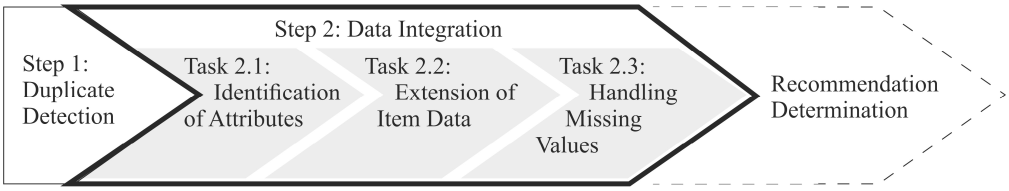

Fig. 4 The Step Data Integration in Detail

The goal of **Task 2.1** is to identify matching attributes. To do so, the attributes of DS2 have to be compared to the attributes of DS1. The automated identification of matching attributes can prove to be non-trivial in the context of recommender systems because different portals often use varying names for the same attribute (e.g., “Artist” and “Performer”) due to heterogeneous data policies. An incorrect matching of attributes can lead to items being assigned wrong data and thus have a direct detrimental impact on recommendation quality. As this task is of relatively low complexity for humans, the identification may be performed in a manual manner (e.g., the manual matching of 143 attributes in DS1 to 251 attributes in DS2 in the application scenario regarding restaurants of our evaluation took approximately one hour and exhibited almost perfect inter-coder reliability). In contrast, an automated identification (e.g., using WordNet) may be error-prone, as it is difficult for an algorithm to directly identify attributes such as “Artist” and “Performer” as matching attributes. Furthermore, an automated identification requires a subsequent manual verification by humans, which is also time-consuming. Overall, an automated identification should only be performed when the number of attributes is extremely high, rendering a manual identification ineffective. In any case, all attributes of DS2 not matched to an attribute of DS1 are identified as additional attributes.

In **Task 2.2**, the item content data is extended for each item in DS1. More precisely, the item content data subsequently consists of the attributes of DS1 and the additional attributes of DS2. Additional attributes allow a more complete view on the considered item and may improve recommendation quality. In particular, additional attribute values can have enormous leverage for users with many item reviews in the context of recommender systems, since a large number of affected rated items can be described in more detail with the additional content. Depending on the recommender system used or under trade-off considerations, it may be helpful to limit the number of the additional attributes considered for data extension. To identify a subset of additional attributes for which a strong improvement of recommendation quality is expected (e.g., attributes with very many missing values may hardly impact recommendation quality), several options are possible (e.g., the use of an attribute selection algorithm; cf. Chandrashekar and Sahin 2014; Molina et al. 2002). These options are discussed in more detail in Section 4.3. After selecting the additional attributes, for each item in DS1 for which a duplicate in DS2 was identified and for each additional attribute chosen, the respective attribute values of the duplicate are inserted into the item content data.

After Task 2.2, some attribute values of items in the extended data set may still be missing because they are not provided by either data set (e.g., the values of the attribute "Genres" are not given for all items in the movie domain). These missing values have to be addressed in **Task 2.3**, since many recommender systems cannot operate on missing attribute values. Moreover, missing attribute values may be detrimental to recommendation quality. Therefore, a further extension of item content data is enabled by imputation methods. More precisely, missing attribute values can be inferred via imputation based on non-missing attribute values in the extended data set. Here, the presented procedure provides an advantage compared to imputing values based on just DS1 as the attribute values from both data sets DS1 and DS2 are available and can be used as basis for the imputation. Table 2 discusses selected imputation methods and their relevance in the context of recommender systems based on Enders (2010). In addition to these imputation methods, it is also feasible to impute values in a user-specific way which is more flexible than assigning fixed values for the missing values in the extended data set. In this case, the missing values of all items rated by a user can be handled by an imputation method from Table 2 (e.g., Arithmetic Mean Imputation) to capture the user's preferences more accurately when generating her/his user profile.

Table 2. Selected Methods for Handling Missing Values and their Application in the Context of Recommender Systems

Imputation methods	*Properties*	*Examples in the context of recommender systems*
Arithmetic Mean Imputation (AMI) Missing attribute values are replaced with the mean attribute value of all items, where the values for this attribute are not missing.	• AMI is convenient to implement • AMI attenuates standard deviation and variance	Each missing value of the attribute "Runtime" is replaced with the mean value of "Runtime" (as an indicator) over all movies that do have a value for "Runtime".
Regression Imputation (RI) Missing values are replaced with predicted scores from regression equations. The regression equations are estimated by analyzing the extended data set.	• RI is complicated to implement • RI attenuates standard deviation and variance (but less than AMI)	For two hotel attributes "Price" (P_i) and "Service" (S_i), there are only missing values for "Service". A regression equation $\hat{S}_i = \hat{\beta}_0 + \hat{\beta}_1(P_i)$ for the attribute "Service", depending on the attribute "Price", is estimated by analyzing the hotels with given values for "Service". The missing values S_i of "Service" are replaced by $\hat{S}_i$.
Hot Deck Imputation (HDI) Missing attribute values of an item are replaced with the corresponding values of the most similar item.	• HDI is convenient to implement • HDI attenuates standard deviation and variance (but less than AMI)	The movie "The Dark Knight" is the most similar movie to "The Dark Knight Rises", as both movies belong to the batman trilogy of the director "Christopher Nolan". The value of "The Dark Knight" for the attribute "Genres" is "Action" and thus, the missing value of "The Dark Knight Rises" for "Genres" is inferred with the value "Action".

3.3 Subsequent Step: Recommendation Determination

Subsequent to duplicate detection and data integration, recommendations for users on e-commerce platforms can be inferred by applying a recommender system based on the extended data set and evaluating the resulting recommendations. This step is also necessary to analyze the effects of data set extension on recommendation quality. As our approach is tailored to data sets containing item content data in addition to rating data, it is feasible to apply both content-based as well as hybrid recommender systems that leverage both data types (Ricci et al. 2015b). Handling item content data is very important in e-commerce settings, because the recommender system can map the potentially extensive needs of customers more accurately due to the more precise description of the items (e.g., proposal of tailored products based on product preferences). Therefore, for this *subsequent* step of our procedure, we suggest to apply the state-of-the-art hybrid recommender system approach Content-Boosted Matrix Factorization (CBMF; cf. Forbes and Zhu 2011), which utilizes both rating data and, in particular, item content data and is thus more comprehensive than collaborative filtering recommender systems. Matrix factorization approaches have become very popular through the Netflix contest, which started in 2006 and ended in 2009 (Koren 2009; Koren et al. 2009), and now constitute state-of-the-art recommender systems (Kim et al. 2016; Ning et al. 2017). As a matrix factorization approach, CBMF learns a model by optimizing a loss function based on training data and therefore, preliminary steps such as attribute weighting or attribute selection are not necessary for CBMF (Koren 2009; Nguyen and Zhu 2013).

The basic idea of matrix factorization recommender systems is to decompose the rating matrix R (users as rows; items as columns) into two low-rank matrices P (representing users) and Q (representing items), with $PQ \approx R$. Then, the idea of CBMF is to further decompose the matrix Q into a low-rank matrix A and the matrix F, with $AF^T = Q$ and F containing the attribute vectors of items (items as rows; attributes as columns). Hence, the overall idea is that the rating matrix R can be approximated by $R \approx PAF^T$. In particular, CBMF learns a n-dimensional vector of latent factors $p_u \in \mathbb{R}^n$ for each user u and a n-dimensional vector of latent factors $a_f \in \mathbb{R}^n$ for each attribute f, such that the actual rating r_{ui} for a user-item pair (u, i) is approximated by the predicted star rating $\hat{r}_{ui} = p_u^T q_i$, with $q_i = \sum_{f \in F_i} a_f$ and F_i being the set of attributes that are assigned to the item i. Finally, to evaluate the effects of the data set extension on recommendation quality, the rating data is split into training data for learning the parameters of the CBMF model (p_u and a_f) and test data to assess the recommendation quality via quality measures such as Root-Mean-Square-Error (RMSE; cf. Shani and Gunawardana 2011).

4 Evaluating the Procedure in Real-world Scenarios

In this section, we evaluate the proposed procedure in two real-world e-commerce scenarios. First, the reasons for selecting these scenarios are discussed and the used data sets are described. Thereafter, the evaluation of the procedure with respect to these data sets is outlined. Finally, important effects of the data set extension regarding items, content and users on recommendation quality are presented.

4.1 Selection and Description of the Real-world Scenarios

We evaluated the procedure in two real-world e-commerce scenarios regarding the domains of restaurants and movies. While these domains are frequent subjects of IS research in e-commerce (Chang and Jung 2017; Nguyen et al. 2018; Wei et al. 2013; Yan et al. 2015), both domains exhibit versatile facets and different challenges for a procedure for data set extension. Thereby, analyzing these two domains allows for a broader evaluation of the proposed procedure in e-commerce application scenarios.

First, we selected the domain of restaurants because this domain is very challenging regarding duplicate detection (i.e., Step 1 of the procedure, e.g., the resolution of multi-linkages of duplicates (Task 1.3)) in the context of recommender systems. In comparison to other domains (e.g., the domain of movies as second scenario) there are items with the same name being found in the immediate vicinity (i.e., in the case of restaurant chains such as McDonald’s or Subway), which makes this domain especially challenging. For the real-world scenario in the domain of restaurants, we prepared data sets of two leading advertising web portals which provide crowd-sourced ratings about businesses (e.g., restaurants). The first portal (Portal R1) focuses on travel opportunities and businesses such as restaurants and provided over 650 million ratings whereas the second portal (Portal R2) specializes on local businesses such as bars or restaurants and provided over 150 million ratings by 2020. These portals were chosen because an initial check revealed that, while both portals contain data about an overlapping set of real-world entities, they offer an interestingly different view (i.e., different attributes) on these entities. In particular, we selected the area of New York City (USA) as both portals provided a large number of items, users and ratings for this area. In this way, the evaluation of the procedure and the analysis regarding its effects on recommendation quality could be performed on a sufficiently large data basis. Here, the data from Portal R1 consists of more than 8,900 items representing restaurants in the area of New York City, rated by over 380,000 users with approximately 850,000 ratings. The data from Portal R2 consists of over 18,500 items representing restaurants in the same area, rated by more than 580,000 users with around 2.4 million ratings. Each item of Portal R1 is described by the key attributes “Name”, “Postal Code”, “Geolocation”, “Address”, “Phone” and “District”, category attributes such as “Italian Cuisine” or “Pizza”, and business information attributes such as “Parking Available” or “Waiter Service”. In Portal R2, items are described by key attributes equaling those in Portal R1 as well as (partly different) category attributes and business information attributes. The data from Portal R1 contains around 3,000 missing values for one attribute whereas the data from Portal R2 contains more than 190,000 missing values for 26 attributes. In our evaluation, we extended the data from Portal R1 with the data from Portal R2 (i.e., the data from Portal R1 served as DS_{R1} and the data from Portal R2 served as DS_{R2}). Table 3 describes the restaurant data sets.

Table 3. Description of the Restaurant Data Sets

	Portal R1 (DS_{R1})	**Portal R2 (DS_{R2})**
# of items (restaurants)	8,909	18,507
# of users	386,958	583,815
# of ratings	855,357	2,396,643
# of key attributes	6	6
# of further attributes (category attributes and business information attributes)	143	251
# of possible attribute values	1,247,260	4,589,736
# of missing values	3,253 (0.26%)	190,789 (4.16%)

In addition, we selected the domain of movies because this domain exhibits further but different challenges regarding item content data extension in the context of recommender systems. In comparison to the restaurant domain, the detection of duplicates and in particular the resolution of multi-linkages of duplicates is less challenging in the movie domain, since different movies have usually different titles (as key attribute) due to copyright standards. Nevertheless, Step 1 of the procedure is still favorable for movies in order to detect non-trivial movie duplicates in case the movie titles do not exactly match, as key attributes can (slightly) vary between different portals in some cases (e.g., the movie titles "Mission: Impossible – Ghost Protocol" and "Mission: Impossible – Ghost Protocol (2011)" represent the same item). Moreover, an initial check revealed that the amount of missing values in the data sets of both movie web portals (Portal M1 and Portal M2) is very high compared to other domains (e.g., restaurants). This means that Step 2 of the procedure including the task of handling missing values is even more important for the real-world scenario in the movie domain. Hence, we prepared data sets of two leading web portals which provide crowd-sourced ratings about movies. Here, the data from Portal M1 consists of approximately 29,000 movie items, rated by over 425,000 users with nearly 530,000 ratings. The data from Portal M2 consists of over 12,500 movie items, rated by approximately 230,000 users with nearly 410,000 ratings. Each item of Portal M1 is described by the key attribute "Title" and further attributes such as "Brand". In Portal M2, items are described by the same key attribute as in Portal M1 as well as by further attributes such as "Cast" and "Language". The data from Portal M1 contains over 245,000 missing values for all attributes whereas the data from Portal M2 contains more than 1 million missing values for all attributes. In our evaluation, we extended the data from Portal M1 with the data from Portal M2 (i.e., the data from Portal M1 served as DS_{M1} and the data from Portal M2 served as DS_{M2}). Table 4 describes the movie data sets.

Table 4. Description of the Movie Data Sets

	Portal M1 (DS_{M1})	**Portal M2 (DS_{M2})**
# of items (movies)	28,973	12,842
# of users	428,519	230,151
# of ratings	528,777	409,935
# of key attributes	1	1
# of further attributes	13	103
# of possible attribute values	376,649	1,322,726
# of missing values	247,341 (65.67%)	1,082,387 (81.83%)

4.2 Evaluation of the Procedure

In this section, we discuss the evaluation of the procedure for extending data sets with item content data in the restaurant and movie domain and present the evaluation results for each step for both domains.

4.2.1 Evaluation of Step 1 – Duplicate Detection

In the following, we outline the evaluation of the duplicate detection step. More precisely, the goal of this section is to assess the evaluation metrics precision, recall and F1-measure of duplicate detection. Therefore, we first discuss how we conducted and validated the tasks of this step and then present the evaluation results.

Since this step is more challenging for restaurants, we especially focus on this domain.

To begin with, in Task 1.1, the key attribute values (cf. Table 5) of DS_{R1} and DS_{R2} were standardized due to inconsistent values caused by heterogeneous data policies among restaurant portals. For example, the postal code in DS_{R1} was given in the format “ZIP+4” (containing the standard five-digit postal code with four additional digits for postal delivery, e.g., “10019-2132”) and in DS_{R2} in the format “ZIP” (containing the standard five-digit postal code, e.g., “10019”). Hence, “Postal Code” was restricted to only the standard five-digit postal code “ZIP” (e.g., “10019”) to achieve standardized key attribute values. In the data preparation subtask, pairs of restaurants which were more than 1,000 meters apart from each other based on the key attribute “Geolocation” were removed, due to these restaurant pairs being obvious non-duplicates. This led to a total of 11,492 item pairs, constituting the data for the next task “Item Pair Classification”. Task 1.2 was initialized by selecting adequate SMFs for all key attributes, following the argumentations given in Section 3. For example, the SMFs “Jaro-Winkler” and “Levenshtein” were proved as useful for the key attributes “Name” and “Address” and the SMF “Haversine” was beneficial for “Geolocation” (Kamath et al. 2013). These key attributes were selected as no natural unique IDs for the restaurants were available across DS_{R1} and DS_{R2}. The duplicate detection then yielded at first 6,226 pairs classified as duplicates and 5,266 item pairs classified as non-duplicates. In Task 1.3, multi-linkages of items were resolved. For example, the restaurant “Sushi You” in DS_{R1} was contained in two item pairs classified as duplicates (with the restaurant “Sushi You” from DS_{R2} in the first pair and with the restaurant “Sushi Ko” from DS_{R2} in the second pair). Here, the prediction score of the first pair was higher than the score of the second one and therefore, only the first pair was retained. After resolving such multi-linkages, the number of duplicate item pairs decreased to 5,919. With regard to Task 1.4, 500 item pairs (250 items presumed to be duplicates and 250 items presumed to be non-duplicates) were selected to validate our duplicate detection step. Thereby, the item pairs were examined by a web-based search which involved 1) visiting the homepages of the restaurants, 2) searching the restaurants via *Google Maps* and 3) using *Google Street View* to check the identity of restaurants. This method was necessary to reliably determine actual duplicates and non-duplicates as some non-duplicate item pairs were hard to identify. For example, the restaurants “Murray’s Cheese Shop” in DS_{R1} located at “254 Bleecker St” in “West Village” and “Murray’s Cheese Bar” in DS_{R2} at “264 Bleecker St” in “West Village”, which seem to be very similar at first sight, turned out to be non-duplicates after the examination. The validation of the duplicate detection yielded a precision of 95.9% (i.e., 235 of 245 classified duplicates were real duplicates; 240 of 255 classified non-duplicates were real non-duplicates), a recall of 94.0% (i.e., 235 of 250 real duplicates were classified as duplicates; 240 of 250 real non-duplicates were classified as non-duplicates) and a F1-measure of 94.9%, demonstrating a very high quality. Summing up, the first step of the procedure yielded 5,919 duplicate restaurant item pairs of high quality constituting the basis for Step 2 of the procedure.

Table 5. Key Attributes of both Restaurant Portals

Key attributes	*Data type*	*Example key attribute values from both portals for a duplicate*
Name	String	“9 Ten Restaurant” (in DS_{R1}), “9 10 Restaurant” (in DS_{R2})
Postal Code	Number	“10019-2132” (in DS_{R1}), “10019” (in DS_{R2})
Geolocation	Geographic coordinates (latitude and longitude)	“N 40.76591° / W -73.97979°” (in DS_{R1}), “N 40.7659964050293° / W -73.9797178100586°” (in DS_{R2})
Address	String	“910 Seventh Avenue” (in DS_{R1}), “910 7th Av” (in DS_{R2})
Phone	Number	“+1 917-639-3366” (in DS_{R1}), “(917) 639 3666” (in DS_{R2})
District	String	“Midtown” (in DS_{R1}), “Midtown West” (in DS_{R2})

Next, we briefly outline the first step of the procedure for the movie domain. As described before, the duplicate detection step for the movie domain is in general less challenging than for the restaurant domain due to copyright standards. However, titles of movie duplicates do not always exactly match, since different movie portals have heterogeneous data policies (e.g., the movie titles “Mission: Impossible – Ghost Protocol” and “Mission: Impossible – Ghost Protocol (2011)” represent the same item). Hence, standardization of the key attribute “Title” in both data sets DS_{M1} and DS_{M2} is necessary (e.g., removing the year of the movie’s release). Thereafter, many duplicates can be detected directly by matching the standardized “Title” of movies in a large part of the cases (cf. Section 4.1). Similar as for restaurants, pairs of movies which were obvious non-duplicates (based on similarities of the key attribute “Title”) were removed during blocking leading to 10,160 item pairs as result of Task 1.1. Since DS_{M1} also contained items going beyond regular cinematographic movies (e.g., other film material such as “The Theory of Evolution: A History of Controversy”), item pairs could only be identified for the mentioned 10,160 items in DS_{M1}. In Task 1.2, SMFs such as “Jaro-Winkler” and “Levenshtein” were used for the key attribute “Title” for conducting item pair classification similarly as for restaurants. With no multi-linkages present in the result of Task 1.2 (i.e., Task 1.3 could be skipped), 9,438 movie item pairs were detected as duplicates. Similarly, as for restaurants, 500 item pairs were prepared to validate duplicate detection by a manual web-based search. The validation of the duplicate detection for movies in Task 1.4 yielded a precision of 95.1%, a recall of 96.7% and a F1-measure of 95.9%, demonstrating a very high quality for detecting duplicates. Summing up, the first step of the procedure yielded 9,438 duplicate movie item pairs of high quality constituting the basis for Step 2 of the procedure.

4.2.2 Evaluation of Step 2 – Data Integration

In this section, we outline the evaluation of the data integration step. The goal of this section is to assess how the completeness of the item content data could be increased through data integration. Therefore, we first establish how we conducted and validated the tasks of Step 2 of the procedure and then present the results of the evaluation. Since the number of further attributes in DS_{M2} (compared to DS_{M1}) and the numbers of missing attribute values in DS_{M1} and DS_{M2} are very high (cf. Table 4), Step 2 is of particular relevance for the real-world scenario regarding the movie domain. Nevertheless, Step 2 is also crucial for the real-world scenario regarding restaurants, as in this step the actual data set extension is performed.

Following Task 2.1, as heterogeneous data policies among portals in the restaurant domain had led to different names of the same attribute and different levels of granularity used across DS_{R1} and DS_{R2}, all attributes of DS_{R2} were compared to the attributes of DS_{R1} to identify matching and additional attributes. Thereby, 57 attributes of DS_{R2} such as “Japanese”, “Pizza” or “Vegan” were identified as matching attributes and 194 attributes of DS_{R2} such as “Attire”, “Karaoke” or “Take Out” were identified as additional attributes in a manual check requiring approximately one hour of work, exhibiting almost perfect inter-coder reliability. According to Task 2.2, these additional attributes are to be analyzed regarding an extension of DS_{R1}. Here, for a first evaluation regarding the effects on recommendation quality, we used all additional attributes for the extension of DS_{R1}. Thus, the extended data set contained all attributes of DS_{R1} and all additional attributes of DS_{R2}. Thereafter, the item content data of DS_{R1} was extended and attribute values of duplicates were inserted. Further, we validated Task 2.3, which means, the remaining missing attribute values were imputed in a first step. To this end, we evaluated the use of the Hot Deck Imputation method (cf. Table 2), allowing the replacement of all missing values and yielding an item content data set without missing values. In total, the evaluation shows that the completeness of the item content data of DS_{R1} can be increased by integrating 194 additional attributes from DS_{R2} and by imputation of 3,253 values in DS_{R1} and 190,789 values in DS_{R2}. This emphasizes the superior data quality of the resulting extended data set compared to the basis data set DS_{R1} regarding the dimension completeness.

In the case of the movie data sets, all 103 attributes of DS_{M2} such as “Genres”, “Cast” or “Language” were identified as additional attributes in Task 2.1. In Task 2.2, for a first evaluation regarding the effects on recommendation quality, we used all additional attributes of DS_{M2} for the extension of DS_{M1} similar to the case of restaurants. Thus, the attributes and values were inserted for the identified duplicates and thus, the extended data set contained all attributes of DS_{M1} and all attributes of DS_{M2}. In Task 2.3, the remaining missing attribute values were imputed by means of the Hot Deck Imputation method (cf. Table 2) yielding an item content data set without missing values. In total, the evaluation shows that the completeness of the item content data of DS_{M1} can be increased by integrating 103 additional attributes from DS_{M2} and by imputation of 247,341 values in DS_{M1} and 1,082,387 values in DS_{M2}. Therefore, the resulting extended data set shows strongly increased data quality compared to the basis data set DS_{M1} regarding the dimension completeness.

Table 6. Challenges in the Context of Recommender Systems

Topics	*Challenges in the context of recommender systems*	*References to procedure step / task*
Data / Content	• Decentral data capturing by many different users results in data quality problems requiring standardization • Heterogeneous data policies among portals lead to different characteristics of the data across data sets, also requiring standardization • Item content data is a central decisive factor for e-commerce business models and respective recommender systems	**1.1 Data Standardization and Preparation**
Key Attributes and Item Pair Classification	• Labeled training data is missing in the context of recommender systems for a supervised item pair classification • No natural unique IDs are available for items (e.g. restaurants) • Values of key attributes are entered in a decentral way and depend on the users' own interpretation leading to highly diverse data values • Items with the same name referring to the same organization (e.g., "McDonald's") and items with similar names referring to different organizations (e.g., "Sushi You" vs. "Sushi Ko") in the restaurant domain are potentially in close proximity in urban areas; however, they have to be distinguished as separate items	**1.2 Item Pair Classification**
Matching Attributes	• Heterogeneous data policies among portals lead to different names of the same attribute (e.g., "Bar" vs. "Pub") • Portals potentially use different levels of granularity when describing the attributes (e.g., "Asian Cuisine" vs. "Japanese Cuisine")	**2.1 Identification of Attributes**
Additional Attributes	• Attributes and their values (e.g., eight times more attributes after data set extension in the movie domain) directly affect the quality of the recommender system and the resulting recommendations	**2.2 Extension of Item Data**
Missing Values	• Many recommender system techniques cannot handle missing values (e.g., 75% missing attribute values had to be imputed in the movie domain)	**2.3 Handling Missing Values**

4.2.3 Evaluation of Subsequent Step – Recommendation Determination

Finally, we discuss the evaluation of the recommendation determination based on the extended data sets with increased completeness regarding both domains. After the data set extension in the first two steps of the procedure, the recommendations based on the extended data sets could be computed. As indicated in Section 3, we validated whether the hybrid recommender system approach CBMF (Forbes and Zhu 2011; Nguyen and Zhu 2013) can be utilized. We followed Nguyen and Zhu (2013) in regard to the default configuration for CBMF, with the only exception being the regularization penalty factor λ, which has to be adjusted depending on the data set at hand (Koren et al. 2009). To this end, we compared the results of cross-validation tests of different values for λ as described by Koren et al. (2009). In these tests, the value $\lambda = 10^{-5}$ yielded the best results in terms of RMSE. After the execution of CBMF, the recommendations were evaluated by the following standard technique (cf., e.g., Shani and Gunawardana 2011). The ratings of DS_{R1} and DS_{M1} were randomly split into a training set (67% of ratings) to learn the parameters of the CBMF model (p_u and a_f, cf. Section 3) and a test set (33% of ratings) for assessing recommendation quality. We quantified recommendation quality by the RMSE between the real ratings and the predicted ratings of the CBMF in the test set. To assess the recommendation quality based on the extended data sets compared to just data sets DS_{R1} or DS_{M1}, respectively, the training of the CBMF parameters and the assessment of recommendation quality were validated on either the item content data of the extended data set or just on the item content data of DS_{R1} or DS_{M1}. Here, in both cases (extended data set compared to the basis data set) the train-test-split remained the same such that a meaningful comparison of both cases was possible for both domains. The recommendation determination could be applied in each case without restrictions and yielded recommendations for each user. In particular, our procedure was able to successfully navigate numerous challenges in this context (cf. Table 6), which are common when trying to extend an item content data set with respect to the data quality dimension completeness. This successful validation of the determined recommendations concludes the evaluation of the proposed procedure in both real-world scenarios.

4.3 Effects on Recommendation Quality

In addition to the evaluation of the procedure itself in Section 4.2, we observed and examined effects of our procedure on recommendation quality in both e-commerce real-world scenarios. These effects can serve as a starting point for further investigations of the impact of completeness on the recommendation quality based on our procedure (cf. Section 2.2). In particular, besides evaluating the general impact of increased completeness on recommendation quality when applying the proposed procedure (Effect 1), we also investigated effects in detail on the results of the procedure from the three major dimensions related to (content-based and hybrid) recommendations in e-commerce (Heinrich et al. 2019): Items (Effect 2), content in form of attributes (Effect 3) and attribute values (Effect 4), and users (Effect 5). An overview of the results regarding these effects for both the restaurant and the movie domain is given in Table 7.

Table 7. Overview of Improvements in Recommendation Quality for each Effect

Effects	*Evaluation configurations*		*Relative improvements in recommendation quality (RMSE) by procedure application*	
			Restaurants	*Movies*
1	Standard procedure configuration (as outlined in section 4.2)		13.2%	24.6%
2	Procedure with simplified rule-based duplicate detection		9.8%	23.9%
3	Procedure without imputation and …	additional attributes with low number of available attribute values (Set 1)	0.1%	1.7%
		additional attributes with high number of available attribute values (Set 2)	12.6%	17.4%
		all additional attributes (Set 3)	12.7%	17.4%
		all attributes of DS2 (Set 4)	12.6%	17.4%
4	Standard procedure configuration (as outlined in section 4.2) (Setting 1)		13.2%	24.6%
	Procedure without imputation (Setting 2)		12.7%	17.4%
	Procedure without imputation and further removed attribute values (Setting 3)		6.5%	13.7%
5	Procedure for users with high rating numbers (Group 1)		17.1%	45.4%
	Procedure for users with moderate rating numbers (Group 2)		16.3%	42.7%
	Procedure for users with low rating numbers (Group 3)		9.9%	6.0%

Effect 1. Extending the basis data set (DS_{R1} and DS_{M1}, respectively) by applying the proposed procedure improved recommendation quality considerably.

Scenario regarding restaurants: Indeed, the more complete view on restaurants provided by the extended data set led to an improvement in recommendation quality of 13.2% (the RMSE achieved for the extended data set is 0.89, while the RMSE for just DS_{R1} is 1.02). The more complete view and its effect can be illustrated by an example considering the user "Michelle", who had submitted 43 ratings overall. This user had, in reality, rated the restaurant "ShunLee" with a score of 4 stars. The rating of this restaurant as estimated by CBMF based on just DS_{R1} was 1 star, which means that there was a huge discrepancy between the real and the estimated rating. In the extended data set, the item vector of "ShunLee" was extended by all additional attributes and attribute values of its duplicate in DS_{R2} as described above. This extension led to a large improvement, as CBMF based on the extended data set determined a rating of 3 stars, which is much closer to the real rating of the user. Overall, the recommendations for "Michelle" based on the extended data set and based on just DS_{R1} resulted in RMSEs of 0.56 and 3.78, respectively. This example further illustrates the (considerable) improvement of recommendation quality.

Scenario regarding movies: Compared to the restaurant domain, the overall effect of the procedure in the movie domain is even stronger, as the extension of DS_{M1} led to an improvement in recommendation quality of 24.6%. However, the baseline RMSE of 3.15 based on just DS_{M1} is inferior for the movie domain compared to the restaurant domain with a baseline RMSE of 1.02, which means, improving a higher baseline RMSE is comparatively easier. This puts the high improvement in recommendation quality in perspective. Besides this, individual analyses of users regarding improvements in recommendation quality can be performed analogously to the description above for restaurants.

Effect 2. A sophisticated duplicate detection as proposed by our procedure yielded a high improvement in recommendation quality.

Scenario regarding restaurants: In order to investigate the importance of duplicate detection (cf. Section 3.1) on the resulting recommendation quality, we further instantiated and evaluated the procedure with an alternative rule-based duplicate detection algorithm (cf. Christen 2012). To evaluate this alternative algorithm, we performed Task 1.1, Task 1.3 and Task 1.4 in the same way, but for Task 1.2, we chose the following decision-rule aiming for a simple but transparent classification of item pairs (A, B):

If $jaro\_winkler\_similarity_{name}(A, B) > T_1$ **and** $haversine\_similarity_{geolocation}\ (A, B) > T_2$ **then** item B is classified as a duplicate of item A **else** item B is not classified as a duplicate of item A.

We evaluated different threshold configurations for T_1 and T_2 resulting in the best validation results for the thresholds $T_1 = 0.9$ and $T_2 = 0.909$ (corresponding to a distance of 100 meters), which were used for the rule-based item pair classification. As the rule-based duplicate detection was rather restrictive with judging pairs of items to be a duplicate, the fewer pairs of items identified as duplicates by the rule-based duplicate detection were almost all correctly classified, resulting in a high precision of 96.8% (compared to 95.9% precision of the sophisticated duplicate detection). However, the rule-based duplicate detection mainly just identified the rather obvious duplicates, leading to this high precision but a quite low recall. More precisely, it was only able to identify 72.8% of duplicates as indicated by the recall (compared to 94.0% recall of the sophisticated duplicate detection). Thus, the rule-based duplicate detection also exhibited an overall lower F1-measure of 83.1% compared to 94.9% for the sophisticated duplicate detection, demonstrating the higher quality of the sophisticated duplicate detection. The assessed improvement in recommendation quality when conducting the remainder of the procedure using this duplicate detection with lower quality was only 9.8% (compared to 13.2% improvement for the sophisticated duplicate detection with higher quality assessed on the same test set of ratings as in Effect 1). These results show that the sophisticated duplicate detection algorithm proposed by our procedure led to a significantly higher improvement in recommendation quality.

Scenario regarding movies: Similarly, as for restaurants, we instantiated and evaluated a rule-based duplicate detection algorithm in the movie domain yielding 85.3% for F1-measure (compared to 95.9% for the sophisticated duplicate detection). Nevertheless, even the procedure with the rule-based duplicate detection yields an improvement in recommendation quality by 23.9%, which is smaller than the improvement based on the sophisticated duplicate detection, which is 24.6%.

Effect 3. The extension of the basis data set (DS_{R1} and DS_{M1}, respectively) with further attributes (of DS_{R2} and DS_{M2}, respectively) generally supported the increase in recommendation quality, with the extent of improvement depending on the attribute set used for the extension.

Scenario regarding restaurants: To analyze and separate the effect of additional attributes for extension in Task 2.2, we split all additional attributes from DS_{R2} into two equally sized groups based on the absolute number of available values per attribute. First, we extended DS_{R1} with the set of additional attributes from DS_{R2} with a low number of available attribute values (Set 1), leading to an improvement in recommendation quality of just 0.1%. Second, the extension of DS_{R1} with the set of additional attributes with a high number of available attribute values

(Set 2) achieved an improvement of 12.6%. In comparison, the extension of DS_{R1} with all additional attributes of DS_{R2} (Set 3) led to an improvement of 12.7%.[4] These results show that while the extension with additional attributes generally contributed to an improvement of recommendation quality, the extent of improvement depended on the number of available attribute values of the additional attributes. Thus, these results indicate that the increase in recommendation quality could mainly be traced back to attributes with a high number of available attribute values. Moreover, we investigated the extension of DS_{R1} with *all attributes* of DS_{R2} (Set 4; i.e., additional attributes *and* matching attributes from DS_{R2}) in order to further analyze this effect. This means, we omitted the identification of matching attributes (cf. Task 2.1) and extended DS_{R1} with all attributes of DS_{R2} (i.e., additional and matching attributes). Although another 57 (matching) attributes were added compared to the extension with only additional attributes, the improvement of recommendation quality decreased slightly by 0.1% to 12.6%. This finding based on our chosen real-world scenario supports that more data (i.e., more attributes and attribute values) does not always lead to better results of decision support systems and, in particular, recommender systems (cf. Section 2.2). Therefore, the additional and more complete data provided by the matching attributes did not yield any added value, which is in line with works such as Bleiholder and Naumann (2008). In our application context, the matching of attributes led to just a slight improvement of the recommendation quality (0.1%), however, there may be application areas in which the matching of attributes contributes even more to an improvement of the recommendation quality and therefore Task 2.1 of the procedure is essential.

Since both adding attributes and identifying matching attributes may cause effort, it would be interesting to further investigate how to choose an adequate balance between these efforts and the resulting benefits of improved recommendation quality. For instance, when the efforts for adding attributes are low, all additional attributes can be selected for extension. Otherwise, a limitation to a smaller set of (additional) attributes (e.g., attributes with a high number of available attribute values) may be reasonable to reduce high efforts while simultaneously accomplishing a similarly high improvement of recommendation quality.

Scenario regarding movies: As for restaurants, we analyzed four sets of additional attributes (Set 1-4) from DS_{M2} regarding an improvement in recommendation quality. Since the scenario regarding movies did not yield matching attributes, all attributes of DS_{M2} constituted additional attributes and thus, the attribute sets Set 3 and Set 4 were identical. Here, the results regarding this effect for movies further underline the findings identified for restaurants as the improvement of 1.7% in recommendation quality for Set 1 was small compared to high improvements of 17.4% for the Sets 2-4. That is, the increase in recommendation quality could mainly be traced back to attributes with a high number of available attribute values.

Effect 4. More attribute values (i.e., less missing values) resulted in increased recommendation quality.

Scenario regarding restaurants: In addition to the analysis of the set of attributes, we also investigated effects of item content data with respect to (missing) attribute values. We fixed the set of attributes in the extended data set and focused on the imputation of missing attribute values (cf. Task 2.3) in order to separate Effect 4. We examined three settings with a varying number of (missing) attribute values. In the first setting, we imputed all missing values according to Task 2.3, resulting in no missing values in the item content data set used. The second setting used the extended data set without imputing missing values. In our real-world scenario regarding restaurants, however, only four percent of attribute values were missing, which could limit the extent of potential effects of missing attribute values. Therefore, we considered a third setting, in which we randomly removed an additional ten percent of attribute values from the extended item content data set to examine the effect of missing attribute values more generally in the restaurant domain. This led to a total of fourteen percent of missing attribute values in this third setting. We evaluated all three settings regarding resulting improvements in recommendation quality (i.e., RMSE based on the extended data set vs. RMSE based on just DS_{R1}). The results showed an improvement in recommendation quality of 13.2% for the first setting, 12.7% for the second setting and 6.5% for the third setting.

Scenario regarding movies: In contrast to the scenario regarding restaurants, the movie data sets showed high numbers of missing attribute values (cf. Table 4) making this scenario especially promising for analyzing the effect of imputing missing values (in Step 2 of the procedure) on recommendation quality in a real-world e-commerce application scenario. Similarly, as for restaurants, we examined the three settings with a varying number of missing attribute values. The results showed an improvement in recommendation quality of 24.6% for the first setting (i.e., the extended data set with imputed missing values), 17.4% for the second setting (i.e., the extended data set without

[4] The difference between the improvement of 12.7% in Effect 3 and the improvement of 13.2% in Effect 1 can be attributed to the fact that imputation of missing values is omitted in Effect 3.

imputed missing values) and 13.7% for the third setting (i.e., the extended data set without imputed missing values and 10% further removed attribute values).

These results emphasize that recommendation quality benefits significantly from having more attribute values and, in particular, from imputing missing values, which constitutes a main task in the proposed procedure (cf. Task 2.3).

Effect 5. Users with a high number of submitted ratings benefitted more from the data set extension than users with a low number of submitted ratings.

Scenario regarding restaurants: For the analysis of this effect, we examined the relation between the number of ratings submitted by users and the increase in recommendation quality. To do so, we grouped all users into three equally sized groups based on their number of submitted ratings in the training set and examined the three groups individually regarding their improvement in recommendation quality. The first group containing users with the highest number of ratings (averaging about 29 ratings submitted per user) achieved a RMSE improvement of 17.1%. The second group, whose users had on average submitted about 15 ratings, recorded a RMSE improvement of 16.3%. Finally, the third group of users, with an average of about 10 ratings submitted per user, achieved the lowest improvement of recommendation quality, accomplishing a RMSE improvement of 9.9%.

Scenario regarding movies: Analogous as for restaurants, we grouped the users in the movie scenario into three equally sized groups. The first group, whose users had on average submitted about 4 ratings, achieved the highest RMSE improvement of 45.4%. The second group, whose users had submitted about 2 ratings on average, still recorded a high RMSE improvement of 42.7%. Finally, the third group of users, with an average of about 1 rating submitted per user, achieved the lowest improvement of recommendation quality, accomplishing a RMSE improvement of only 6.0%. Although the improvement for the third user group is small, it is still noteworthy as these users with just 1 submitted rating have only rating data in either the training set or the test set. In particular, this means that even users without ratings at all (i.e., without ratings in the training set) benefit from extending the item content data set, which is of high relevance for e-commerce applications, as the case of new users occurs very frequently.

Overall, these results indicate that the improvement of recommendation quality depended on the number of ratings submitted by users, and that users with a higher number of submitted ratings benefitted more. In a detailed analysis, we concluded that this effect can be attributed to the fact that users with a higher number of submitted ratings mainly rated items for whom more item content was added. Thus, the extended data set enabled the recommender system to infer these users' ratings even more accurately.

5 Conclusion, Limitations and Directions for Future Work

Researchers have highlighted the relationship between data quality and decision support systems, and in particular recommender systems, in the field of IS. Based on a theoretical model, we present a procedure for the systematic extension of a data set DS1 with additional item content (attributes and attribute values) from another data set DS2 in the same domain. Thereby, the procedure aims to address data quality, especially by increasing the completeness of data sets and, in consequence, to improve recommendation quality of recommender systems. In a first step, an approach to detect duplicate items across data sets DS1 and DS2 is proposed. In a second step, we outline how item content data in DS1 can be extended by integrating the item content data of a data set DS2 as well as by imputing missing values. Based on these two steps, the resulting extended data set can be used by an arbitrary content-based or hybrid recommender system to determine recommendations in a subsequent step. We evaluate the procedure by using two real-world data sets regarding restaurants and movies, which constitute commonly analyzed domains in IS research on e-commerce, and discuss effects on recommendation quality. Here, the results show that the presented procedure is indeed capable of improving recommendations considerably by means of item content data extension, which is in line with existing research (cf. Heinrich et al. 2019). Furthermore, we investigate different effects on the results of the procedure from the three dimensions items, content and users, revealing that the procedure was valuable in each investigated case and indicating under which circumstances a substantial improvement in recommendation quality was achieved. Complementary to existing research proposing general relationships between data quality and decision support systems, this work provides and evaluates a tangible procedure which enables to increase data completeness with the aim of improving recommendation quality. Moreover, this procedure serves an evaluated template for future procedures to support the investigation of further data quality dimensions (e.g., consistency) for decision support systems in various e-commerce applications.

The rapid proliferation of e-commerce has cemented the tremendous relevance of recommender systems. These systems are powerful data-driven decision support systems incorporated in many e-commerce platforms guiding users to their individually best item choice among a plethora of alternatives. Thereby, recommender systems address the problem of information overload, which constitutes a major subject of IS research in the field of e-commerce. While the steady increasing volume of information (e.g., about attributes of items) would further aggravate the problem of information overload for users, recommender systems actually can somehow invert this effect. In contrast to the limited cognitive capabilities of users, for recommender systems as automated data-driven systems, more information (e.g., item content data; i.e., attributes and attribute values) is highly useful to individually support the user's decision-making and thus to further reduce the problem of information overload. To do so, increasing the completeness of the data (i.e., item content data) a recommender system is based on seems to constitute a promising way, which is studied in this paper by proposing a procedure for data set extension. Especially in established e-commerce domains (e.g., restaurants and movies), a higher completeness can significantly improve the recommendation quality for users (e.g., the selection of restaurants and movies), which in the long run strengthens the relationship between providers and users.

Here, our evaluation encourages IS providers in e-commerce (e.g., online portals) to improve data quality by providing a straightforward way to increase completeness without the need of manual tasks such as visiting items' websites or social media pages. Our procedure shows that achieving high data quality is indeed beneficial for companies, as the resulting improved recommendations support the various goals and purposes of recommender systems such as promoting cross- and up-selling or increasing customer loyalty (Jannach and Adomavicius 2016). Moreover, our results open up a way for portals with limited resources to balance the efforts and benefits associated to the procedure. For instance, as recommending items based on massively extended item content data can prove to be time-consuming, portals may prefer to focus on a subset of users or additional attributes based on the evidence found in Section 4.

However, our work also has some limitations, which could be starting points for future research. First, while we focused on completeness as a highly relevant data quality dimension, extensions of data sets in the context of recommender systems could also take into account other data quality dimensions such as accuracy or currency. Second, we considered the extension of item content data based on additional structured data in this paper. Here, it would be promising to leverage modern information extraction approaches, such as aspect extraction with language models (e.g., BERT; cf. Xu et al. 2019). Thereby, data sets already used by IS providers could be extended by extracted features from unstructured textual data sources (e.g., online customer reviews). Moreover, another interesting perspective might be to incorporate the extension of user data into the procedure, which could in some cases be realized by, for instance, user linkage based on online social network accounts. Finally, the approach could also be applied to further data sets, possibly from other domains outside the field of e-commerce, in order to validate and substantiate the resulting effects on recommendation quality.

References

Abel, F., Herder, E., Houben, G.-J., Henze, N., & Krause, D. (2013). Cross-system user modeling and personalization on the Social Web. *User Modeling and User-Adapted Interaction, 23*, 169–209. https://doi.org/10.1007/s11257-012-9131-2.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*, 734–749. https://doi.org/10.1109/TKDE.2005.99.

Aggarwal, C. C. (2016). *Recommender Systems*. Cham: Springer International Publishing.

Amatriain, X., Pujol, J. M., Tintarev, N., & Oliver, N. (2009). Rate it again. In L. Bergman, A. Tuzhilin, R. Burke, A. Felfernig, & L. Schmidt-Thieme (Eds.), *The third ACM conference on Recommender systems, New York, New York, USA* (pp. 173–180). New York, NY: ACM. https://doi.org/10.1145/1639714.1639744.

Basaran, D., Ntoutsi, E., & Zimek, A. (2017). Redundancies in Data and their Effect on the Evaluation of Recommendation Systems: A Case Study on the Amazon Reviews Datasets. In N. Chawla & W. Wang (Eds.), *The 2017 SIAM International Conference on Data Mining, Houston, Texas, USA* (pp. 390–398). Philadelphia, PA: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611974973.44.

Batini, C., & Scannapieco, M. (2016). *Data and Information Quality*. Cham: Springer International Publishing.

Berkovsky, S., Kuflik, T., & Ricci, F. (2012). The impact of data obfuscation on the accuracy of collaborative filtering. *Expert Systems with Applications, 39*, 5033–5042. https://doi.org/10.1016/j.eswa.2011.11.037.

Bharati, P., & Chaudhury, A. (2004). An empirical investigation of decision-making satisfaction in web-based decision support systems. *Decision Support Systems, 37*, 187–197. https://doi.org/10.1016/S0167-9236(03)00006-X.

Blake, R., & Mangiameli, P. (2011). The Effects and Interactions of Data Quality and Problem Complexity on Classification. *Journal of Data and Information Quality, 2*, 1–28. https://doi.org/10.1145/1891879.1891881.
Bleiholder, J., & Naumann, F. (2008). Data fusion. *ACM Computing Surveys, 41*, 1–41. https://doi.org/10.1145/1456650.1456651.
Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). TasteWeights: a visual interactive hybrid recommender system. In P. Cunningham, N. Hurley, I. Guy, & S. S. Anand (Eds.), *The sixth ACM conference on Recommender systems, Dublin, Ireland* (pp. 35–42). New York, NY: ACM. https://doi.org/10.1145/2365952.2365964.
Bouadjenek, M. R., Pacitti, E., Servajean, M., Masseglia, F., & Abbadi, A. E. (2018). A Distributed Collaborative Filtering Algorithm Using Multiple Data Sources. *arXiv preprint arXiv:1807.05853*.
Bunnell, L., Osei-Bryson, K.-M., & Yoon, V. Y. (2019). RecSys Issues Ontology: A Knowledge Classification of Issues for Recommender Systems Researchers. *Information Systems Frontiers, 97*, 667. https://doi.org/10.1007/s10796-019-09935-9.
Burke, R., & Ramezani, M. (2011). Matching Recommendation Technologies and Domains. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 367–386). Boston, MA: Springer US.
Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28.
Chang, J.-H., Tsai, C.-E., & Chiang, J.-H. (2018). Using Heterogeneous Social Media as Auxiliary Information to Improve Hotel Recommendation Performance. *IEEE Access, 6*, 42647–42660. https://doi.org/10.1109/ACCESS.2018.2855690.
Chang, W.-L., & Jung, C.-F. (2017). A hybrid approach for personalized service staff recommendation. *Information Systems Frontiers, 19*, 149–163. https://doi.org/10.1007/s10796-015-9597-7.
Christen, P. (2012). *Data matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg.
Edmunds, A., & Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. *International Journal of Information Management, 20*, 17–28. https://doi.org/10.1016/S0268-4012(99)00051-1.
Enders, C. K. (2010). *Applied missing data analysis* (Methodology in the social sciences). New York: Guilford Press.
Feldman, M., Even, A., & Parmet, Y. (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.
Forbes, P., & Zhu, M. (2011). Content-boosted matrix factorization for recommender systems. In B. Mobasher, R. Burke, D. Jannach, & G. Adomavicius (Eds.), *The fifth ACM conference on Recommender systems, Chicago, Illinois, USA* (pp. 261–264). New York, NY: ACM. https://doi.org/10.1145/2043932.2043979.
Ge, M. (2009). *Information quality assessment and effects on inventory decision-making*. Doctoral dissertation. Dublin City University, Dublin.
GitHub. (2020). Procedure Completeness: Extending Item Content Data. https://github.com/ProcedureCompleteness/ExtendingItemContentDataSets. Accessed 14 September 2020.
Hasan, M. R., Jha, A. K., & Liu, Y. (2018). Excessive use of online video streaming services: Impact of recommender system use, psychological factors, and motives. *Computers in Human Behavior, 80*, 220–228. https://doi.org/10.1016/j.chb.2017.11.020.
Heinrich, B., Hopf, M., Lohninger, D., Schiller, A., & Szubartowicz, M. (2019). Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. *Electronic Markets, 23*, 169. https://doi.org/10.1007/s12525-019-00366-7.
Heinrich, B., Klier, M., Schiller, A., & Wagner, G. (2018). Assessing data quality – A probability-based metric for semantic consistency. *Decision Support Systems, 110*, 95–106. https://doi.org/10.1016/j.dss.2018.03.011.
Jannach, D., & Adomavicius, G. (2016). Recommendations with a Purpose. In S. Sen & W. Geyer (Eds.), *The 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA* (pp. 7–10). New York, NY, USA: Association for Computing Machinery.
Jannach, D., Zanker, M., Ge, M., & Gröning, M. (2012). Recommender Systems in Computer Science and Information Systems – A Landscape of Research. *E-Commerce and Web Technologies, 123*, 76–87. https://doi.org/10.1007/978-3-642-32273-0_7.
Jurek, A., Hong, J., Chi, Y., & Liu, W. (2017). A novel ensemble learning approach to unsupervised record linkage. *Information Systems, 71*, 40–54. https://doi.org/10.1016/j.is.2017.06.006.
Kamath, K. Y., Caverlee, J., Lee, K., & Cheng, Z. (2013). Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In D. Schwabe (Ed.), *The 22nd International Conference on the World Wide Web, Rio de Janeiro, Brazil* (pp. 667–678). New York, NY: ACM. https://doi.org/10.1145/2488388.2488447.

Kamis, A., Stern, T., & Ladik, D. M. (2010). A flow-based model of web site intentions when users customize products in business-to-consumer electronic commerce. *Information Systems Frontiers, 12*, 157–168. https://doi.org/10.1007/s10796-008-9135-y.

Karimova, F. (2016). A Survey of e-Commerce Recommender Systems. *European Scientific Journal, ESJ, 12*, 75. https://doi.org/10.19044/esj.2016.v12n34p75.

Karumur, R. P., Nguyen, T. T., & Konstan, J. A. (2018). Personality, User Preferences and Behavior in Recommender systems. *Information Systems Frontiers, 20*, 1241–1265. https://doi.org/10.1007/s10796-017-9800-0.

Kayaalp, M., Özyer, T., & Özyer, S. T. (2009). A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site. In N. Memon (Ed.), *International Conference on Advances in Social Networks Analysis and Mining, 2009, Athens, Greece* (pp. 113–118). Piscataway, NJ: IEEE. https://doi.org/10.1109/ASONAM.2009.41.

Kim, D., Park, C., Oh, J., Lee, S., & Yu, H. (2016). Convolutional Matrix Factorization for Document Context-Aware Recommendation. In S. Sen, W. Geyer, J. Freyne, & P. Castells (Eds.), *The 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA* (pp. 233–240). New York, New York, USA: ACM Press. https://doi.org/10.1145/2959100.2959165.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation, 81*, 1–10.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer, 42*, 30–37. https://doi.org/10.1109/MC.2009.263.

Lathia, N., Amatriain, X., & Pujol, J. M. (2009). Collaborative filtering with adaptive information sources. In S. S. Anand, B. Mobasher, A. Kobsa, & D. Jannach (Eds.), *7th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems, Pasadena, California, USA* (pp. 81–86, CEUR Workshop Proceedings (CEUR-WS.org), Vol. 528).

Levi, A., Mokryn, O., Diot, C., & Taft, N. (2012). Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In P. Cunningham, N. Hurley, I. Guy, & S. S. Anand (Eds.), *The sixth ACM conference on Recommender systems, Dublin, Ireland* (pp. 115–122). New York, NY: ACM. https://doi.org/10.1145/2365952.2365977.

Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. *Informing Science, 9*, 181–212.

Li, Y., Zhang, Z., Peng, Y., Yin, H., & Xu, Q. (2018). Matching user accounts based on user generated content across social networks. *Future Generation Computer Systems, 83*, 104–115. https://doi.org/10.1016/j.future.2018.01.041.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems, 74*, 12–32. https://doi.org/10.1016/j.dss.2015.03.008.

Manca, M., Boratto, L., & Carta, S. (2018). Behavioral data mining to produce novel and serendipitous friend recommendations in a social bookmarking system. *Information Systems Frontiers, 20*, 825–839. https://doi.org/10.1007/s10796-015-9600-3.

Mladenić, D., & Grobelnik, M. (2003). Feature selection on hierarchy of web documents. *Decision Support Systems, 35*, 45–87. https://doi.org/10.1016/S0167-9236(02)00097-0.

Molina, L. C., Belanche, L., & Nebot, À. (2002). Feature selection algorithms: a survey and experimental evaluation. In V. Kumar (Ed.), *IEEE International Conference on Data Mining, Maebashi City, Japan* (pp. 306–313). Los Alamitos, CA: IEEE Computer Society.

Naumann, F., Freytag, J.-C., & Leser, U. (2004). Completeness of integrated information sources. *Information Systems, 29*, 583–615. https://doi.org/10.1016/j.is.2003.12.005.

Nguyen, J., & Zhu, M. (2013). Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis and Data Mining, 6*, 286–301. https://doi.org/10.1002/sam.11184.

Nguyen, T. T., Maxwell Harper, F., Terveen, L., & Konstan, J. A. (2018). User Personality and User Satisfaction with Recommender Systems. *Information Systems Frontiers, 20*, 1173–1189. https://doi.org/10.1007/s10796-017-9782-y.

Ning, Y., Shi, Y., Hong, L., Rangwala, H., & Ramakrishnan, N. (2017). A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation. In P. Cremonesi, F. Ricci, S. Berkovsky, & A. Tuzhilin (Eds.), *The Eleventh ACM Conference on Recommender Systems, Como, Italy* (pp. 23–31). New York, New York, USA: ACM Press. https://doi.org/10.1145/3109859.3109909.

Ntoutsi, E., & Stefanidis, K. (2016). Recommendations beyond the ratings matrix. In Association for Computing Machinery (Ed.), *The Workshop on Data-Driven Innovation on the Web, Hannover, Germany* (pp. 1–5). New York, New York, USA: ACM Press. https://doi.org/10.1145/2911187.2914580.

Ozsoy, M. G., Polat, F., & Alhajj, R. (2016). Making recommendations by integrating information from multiple social networks. *Applied Intelligence, 45*, 1047–1065. https://doi.org/10.1007/s10489-016-0803-1.

Peska, L., & Vojtas, P. (2015). Using Implicit Preference Relations to Improve Content Based Recommending. *E-Commerce and Web Technologies, 239*, 3–16. https://doi.org/10.1007/978-3-319-27729-5_1.

Pessemier, T. de, Dooms, S., Deryckere, T., & Martens, L. (2010). Time dependency of data quality for collaborative filtering algorithms. In X. Amatriain, M. Torrens, P. Resnick, & M. Zanker (Eds.), *The fourth ACM conference on Recommender systems, Barcelona, Spain* (pp. 281–284). New York, NY: ACM. https://doi.org/10.1145/1864708.1864767.

Picault, J., Ribiere, M., Bonnefoy, D., & Mercer, K. (2011). How to get the Recommender out of the Lab? In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 333–365). Boston, MA: Springer US.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM, 45*, 211–218. https://doi.org/10.1145/505248.506010.

Porcel, C., & Herrera-Viedma, E. (2010). Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems, 23*(1), 32–39.

Power, D. J., Sharda, R., & Burstein, F. (2015). *Decision support systems*. Hoboken, New Jersey, USA: John Wiley & Sons, Ltd.

Raad, E., Chbeir, R., & Dipanda, A. (2010). User Profile Matching in Social Networks. In T. Enokido (Ed.), *13th International Conference on Network-Based Information Systems (NBIS), 2010* (pp. 297–304). Piscataway, NJ: IEEE Service Center.

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015a). *Recommender Systems Handbook*. Boston, MA: Springer US.

Ricci, F., Rokach, L., & Shapira, B. (2015b). Recommender Systems: Introduction and Challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–34). Boston, MA: Springer US.

Richthammer, C., & Pernul, G. (2018). Situation awareness for recommender systems. *Electronic Commerce Research, 37*, 85. https://doi.org/10.1007/s10660-018-9321-z.

Sar Shalom, O., Berkovsky, S., Ronen, R., Ziklik, E., & Amihood, A. (2015). Data Quality Matters in Recommender Systems. In H. Werthner, M. Zanker, J. Golbeck, & G. Semeraro (Eds.), *9th ACM Conference on Recommender Systems, Vienna, Austria* (pp. 257–260). New York, NY: ACM. https://doi.org/10.1145/2792838.2799670.

Scannapieco, M., & Batini, C. (2004). Completeness in the Relational Model: a Comprehensive Framework. In *International Conference on Information Quality, Cambridge, Massachusetts, USA* (pp. 333–345).

Scholz, M., Dorner, V., Schryen, G., & Benlian, A. (2017). A configuration-based recommender system for supporting e-commerce decisions. *European Journal of Operational Research, 259*(1), 205–215.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 257–297). Boston, MA: Springer US.

Smith, B., & Linden, G. (2017). Two decades of recommender systems at Amazon. com. *Ieee internet computing, 21*(3), 12–18.

Statista. (2019). Statistics and Market Data about E-commerce. https://www.statista.com/markets/413/e-commerce/. Accessed 3 June 2020.

Steorts, R. C., Ventura, S. L., Sadinle, M., & Fienberg, S. E. (2014). A Comparison of Blocking Methods for Record Linkage. In J. Domingo-Ferrer (Ed.), *Privacy in Statistical Databases* (Vol. 8744, pp. 253–268, Lecture Notes in Computer Science). Cham: Springer International Publishing.

Tang, H., Lee, C. B. P., & Choong, K. K. (2017). Consumer decision support systems for novice buyers – a design science approach. *Information Systems Frontiers, 19*, 881–897. https://doi.org/10.1007/s10796-016-9639-9.

Vanaja, R., & Mukherjee, S. (2019). Novel Wrapper-Based Feature Selection for Efficient Clinical Decision Support System. In L. Akoglu, E. Ferrara, M. Deivamani, R. Baeza-Yates, & P. Yogesh (Eds.), *Third International Conference on Intelligent Information Technologies, Chennai, India* (Vol. 941, pp. 113–129, Communications in Computer and Information Science, Vol. 941). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-3582-2_9.

Vargas-Govea, B., González-Serna, G., & Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. In B. Mobasher, R. Burke, D. Jannach, & G. Adomavicius (Eds.), *The fifth ACM conference on Recommender systems, Chicago, Illinois, USA* (pp. 592–596). New York, NY: ACM.

Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM, 39*, 86–95. https://doi.org/10.1145/240455.240479.

Wei, C., Khoury, R., & Fong, S. (2013). Web 2.0 Recommendation service by multi-collaborative filtering trust network algorithm. *Information Systems Frontiers, 15*, 533–551. https://doi.org/10.1007/s10796-012-9377-6.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods, Alexandria, Virginia*. Alexandria, Virginia, USA: American Statistical Association.

Woodall, P., Borek, A., Gao, J., Oberhofer, M., & Koronios, A. (2015). An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics. In R. Wang (Ed.), *19th International Conference on Information Quality, Xi'an, China* (pp. 24–33, Management and data quality). Red Hook, NY: Curran.

Xu, H., Liu, B., Shu, L., & Yu, P. (2019). BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In (pp. 2324–2335). Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1242.

Yan, X., Wang, J., & Chau, M. (2015). Customer revisit intention to restaurants: Evidence from online reviews. *Information Systems Frontiers, 17*, 645–657. https://doi.org/10.1007/s10796-013-9446-5.

Zhou, L. (2020). Product advertising recommendation in e-commerce based on deep learning and distributed expression. *Electronic Commerce Research, 20*, 321–342. https://doi.org/10.1007/s10660-020-09411-6.

2.2 Paper 2: Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems

Current Status	Full Citation
This paper is **accepted** and **published in** Volume 31, Issue 2 in the journal Electronic Markets.	Heinrich, Bernd; Hopf, Marcus; Lohninger, Daniel; Schiller, Alexander; Szubartowicz, Michael (2021): Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. In Electron Markets 23 (2–3), p. 169. DOI: 10.1007/s12525-019-00366-7.

Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems

Bernd Heinrich, University of Regensburg, Regensburg, Germany, bernd.heinrich@ur.de

Marcus Hopf, University of Regensburg, Regensburg, Germany, marcus.hopf@ur.de

Daniel Lohninger, University of Regensburg, Regensburg, Germany, daniel.lohninger@ur.de

Alexander Schiller, University of Regensburg, Regensburg, Germany, alexander.schiller@ur.de

Michael Szubartowicz, University of Regensburg, Regensburg, Germany, michael.szubartowicz@ur.de

Abstract:

Recommender systems strive to guide users, especially in the field of e-commerce, to their individually best choice when a large number of alternatives is available. In general, literature suggests that the quality of data which a recommender system is based on may have important impact on recommendation quality. In this paper, we focus on the data quality dimension completeness of item content data (i.e., features of items and their feature values) and investigate its impact on the prediction accuracy of recommender systems. In particular, we examine the increase in completeness per item, per user and per feature as moderators for this impact. To this end, we present a theoretical model based on the literature and derive ten hypotheses. We test these hypotheses on two real-world data sets, one from two leading web portals for restaurant reviews and another one from a movie review portal. The results strongly support that, in general, the prediction accuracy is positively influenced by increased completeness. However, the results also reveal, contrary to existing literature, that among others increasing completeness by adding features which differ significantly from already existing features (i.e., a high diversity) does not positively influence the prediction accuracy of recommender systems.

Introduction

Recommender systems strive to guide users to their individually best choice when a large number of alternatives is available. Due to a broad variety of interesting problem settings for scholars and a plethora of practical applications, recommender systems continue to be a topic widely discussed in literature (Adomavicius and Tuzhilin 2005; Bobadilla et al. 2013; Karatzoglou and Hidasi 2017). For example, in recent years, many of these practical applications have been in the field of e-commerce and electronic markets (Li and Karahanna 2015; Lu et al. 2015; Ricci et al. 2011). Thereby, recommender systems "have become one of the most powerful and popular tools" (Ricci et al. 2011), mainly because of the large amount of available data about items (e.g., songs or movies). Here, usually, a choice amongst an abundance of items needs to be made, which has inspired providers such as *Netflix* or *Spotify* to develop elaborate recommender systems (Bell et al. 2007; Gomez-Uribe and Hunt 2016; Song et al. 2013). Similarly, recommender systems can assist users in their choice of which restaurant to visit or in which hotel to stay (Levi et al. 2012; Vargas-Govea et al. 2011). In this context, several works suggest that the quality of the determined recommendations depends on the quality of the data which a recommender system is based on (Adomavicius and Zhang 2012; Felfernig et al. 2007; Konstan and Riedl 2012; Picault et al. 2011; Sar Shalom et al. 2015). As discussed by Jannach et al. (2016), these works mainly investigate the data quality of rating data (e.g., how to achieve the most accurate completion of the user-item matrix with rating predictions) and therefore, propose to leverage additional user data such as the user's context, the user's browsing history or the user's social graph. In contrast to these articles mainly discussing data quality of user or rating data (cf. Section "Background"), this paper focuses on data quality of *item content* data, which means, features of items such as *Genre* or *Actors* of movies and their feature values.

In general, data quality constitutes a multidimensional construct (Pipino et al. 2002; Wand and Wang 1996; Wang et al. 1995) comprising several dimensions such as correctness, completeness and currency of data (Batini and Scannapieco 2016; Heinrich et al. 2018b; Lee et al. 2002; Redman 1996). Some existing works investigate and assess the impact of data quality and its dimensions in decision making (Feldman et al. 2018; Heinrich and Hristova 2016). As recommender systems are an important category of decision support systems, especially in electronic markets, we aim to examine the impact of *item content* data and their quality on the determined recommendations. Here, capturing a more complete view of this item content data (i.e. more available features and feature values) is of particular relevance (Adomavicius and Tuzhilin 2005; Pazzani and Billsus 2007; Picault et al. 2011). After all, "some representations capture only certain aspects of the content, but there are many others that would influence a user's experience" (Lops et al. 2011). Hence, in this paper, we focus on the data quality dimension *completeness*.

Batini et al. (2009) summarize that completeness can be understood as the amount to which an available data view includes data describing the corresponding set of considered real-world objects (cf., e.g., also Ballou and Pazer 1985; Redman 1996). Following this definition, we aim to investigate the impact of completeness on recommendation quality, with completeness being the amount of available features and their feature values describing the set of items. For instance, the movie feature *Genre* has multiple feasible feature values such as

Comedy, *Drama*, *Thriller* and so forth, while the restaurant feature *Cuisine* has multiple feasible feature values such as *Italian*, *American* or *Mexican*. Providers covering such domains typically assign such feature values to items in order to describe and emphasize their (special) characteristics and thus, allow a more complete view on these items. Moreover, to assess the impact of completeness on recommendation quality, we examine the prediction accuracy, which is by far the most discussed quality measure in recommender systems literature (Shani and Gunawardana 2011). In this paper, prediction accuracy is assessed by the familiar evaluation measures Root Mean Squared Error (RMSE), Precision, Recall and F1-measure enabling a broad but also differentiated analysis of the results. To the best of our knowledge, no existing work analyzes the impact of the amount of available features or feature values (*completeness of item content data*) on prediction accuracy. Thus, we focus on the following two research questions:

RQa: *Does the amount of available item features influence the prediction accuracy of recommender systems?*

RQb: *Does the amount of filled up missing item feature values influence the prediction accuracy of recommender systems?*

We address these research questions by formulating ten hypotheses based on a theoretical model derived from the literature. Further, we test the statistical significance of these hypotheses by means of both a t-test and a moderated regression analysis concerning the impact of the amount of available item features and their feature values on prediction accuracy. The results show that completeness of the item content data generally has a significant positive impact on prediction accuracy. However, the results also reveal some findings which are contrary to statements in existing literature (Mitra et al. 2002; Tabakhi and Moradi 2015) stating that adding features with low diversity to a data set has less positive impact on prediction accuracy than adding features with high diversity.

Further, this research is also interesting for practitioners. For instance, the rapid development in e-commerce implies a swiftly increasing number of heavily competing web portals in electronic markets. Thus, increasing prediction accuracy by additional features and feature values may lead to competitive advantages for a portal. Furthermore, portals nowadays have their own individual data sets, which usually vary in their features and feature values for items, even for portals of the same domain (e.g., restaurants as items). Extending a data set with additional item content data from another data set (e.g., in case of a meta search portal) can be highly valuable for a recommender system as the two data sets may offer a differing and, when combined, more complete view of the items at hand. While portals offering a meta view exist (e.g., *trivago.com* compiles pricing data from various hotel portals), these portals usually simply juxtapose the data and do not use it to provide recommendations based on additional features and feature values. Analyzing the impact of increased completeness of item content data on prediction accuracy may reveal substantial unused potential in this context.

The remainder of the paper is organized as follows: In the next section, we discuss related work regarding data quality in the context of recommender systems, especially in terms of the dimension completeness, and outline the theoretical model which is used to substantiate the hypotheses presented in the following section. Thereafter, we discuss the used evaluation measures and testing methodology. In the evaluation section, we statistically test the significance of our hypotheses based on two different real-world data sets. Afterwards, we analyze and discuss the results and give some further practical implications. Finally, we summarize our work and point out limitations as well as directions for future research.

Background and Theoretical Model

This section consists of two subsections covering the literature background and the theoretical model for our research.

Background

In this subsection, we firstly analyze existing works related to our research questions. Thereafter, we identify the research gap which is addressed in this paper. Following the guidelines of standard approaches to prepare the related work (e.g., Levy and Ellis 2006), we performed a literature search on the databases ACM Digital Library, AIS Electronic Library, IEEE Xplore, ScienceDirect and Springer as well as the proceedings of the European and International Conference on Information Systems, the ACM Conference on Recommender Systems and the International Conference on Information Quality. The resulting papers were examined based on title, abstract and keywords, leading to thirteen remaining papers. We performed an additional forward and backward search on these papers, leading to a total of twenty-seven relevant papers. These papers were analyzed in detail and could be organized within three categories A, B and C. Works of category A discuss data quality issues in the context of recommender systems, whereas works of category B present recommender systems which deal with a data set extended by using web data sources. Works of category C investigate the impact of data characteristics such as the entropy of the distribution of rating data on recommendation quality. In the following, we discuss the relevant papers of each category.

The eight works in category A explicitly recognize the importance of data quality for recommender systems from a *general* perspective (Amatriain et al. 2009; Berkovsky et al. 2012; Burke and Ramezani 2011; Konstan and Riedl 2012; Lathia et al. 2009; Levi et al. 2012; Pessemier et al. 2010; Sar Shalom et al. 2015), including several approaches that deal with data quality issues. For instance, as data sparsity and inaccuracy have been identified to influence recommendation quality, Lathia et al. (2009) suggest to choose data sources for the application of a recommender system user-dependently. Sar Shalom et al. (2015) tackle sparsity and redundancy issues by deleting or omitting certain users or items while Pessemier et al. (2010) analyze consumption data such as ratings in regard to currency. Further, Levi et al. (2012) use text mining on user reviews from various sources to alleviate the cold start problem of new users by assigning them to so-called context groups.

The four works in category B (implicitly) investigate completeness in recommender systems (Abel et al. 2013; Bostandjiev et al. 2012; Kayaalp et al. 2009; Ozsoy et al. 2015). More precisely, these works propose to use data from additional web sources to gain an extended data set and to increase recommendation quality in this way. Abel et al. (2013) study user profiles based on aggregated data sets from the social web and show that recommendation quality is improved by user profiles extended through several cross-system user-modelling strategies. Ozsoy et al. (2015) argue that recommendations can be improved by consolidating user data from multiple sources. In their experiments, they show that using multiple user features from several social networks produces an enhanced perspective of user behavior and preferences, leading to improved recommendations. Kayaalp et al. (2009) present an event recommender system for users of a social network. This system collects heterogeneous event data from various web pages to achieve an extended data set and proposes event recommendations on this basis. A further approach is proposed by Bostandjiev et al. (2012). They suggest to use multiple data sources such as Twitter, Facebook and Wikipedia to apply an individual recommender system on each data source. Afterwards, the recommendation results are combined aiming to improve recommendation quality.

The fifteen works in category C examine the impact of data characteristics (so-called meta-features) on recommendation quality. In particular, these works investigate the impact of data characteristics of rating data (e.g., Adomavicius and Zhang 2016; Griffith et al. 2012; Matuszyk and Spiliopoulou 2014), content data (Fortes et al. 2017) and other data such as binary purchase data (Geuens et al. 2018), social network graph data (Olteanu et al. 2014) or folksonomy data (Doerfel et al. 2016) on different performance measures of recommender systems. For instance, Cunha et al. (2016), Ekstrand and Riedl (2012) and Huang and Zeng (2005) aim to select the best recommender algorithm depending on data characteristics such as the entropy of ratings. Furthermore, Adomavicius and Zhang (2012), Basaran et al. (2017) and Grčar et al. (2006) analyze the recommendation quality based on rating data specific meta-features such as the user-item ratio. As meta-features usually provide valuable information, for instance, Sergis and Sampson (2016) and Zapata et al. (2015) enhance hybrid recommender systems by including the meta-features directly as input to the recommender algorithm.

Given this discussion, none of the works above investigates the impact of completeness of item content data on recommendation quality. The works in category A focus on data quality issues in recommender systems, analyzing the impact of dimensions such as accuracy and currency on recommendation quality. We extend this category of works by contributing investigations for the impact of completeness on recommendation quality. The works in category B focus on completeness aspects in the context of recommender systems. Abel et al. (2013) and Ozsoy et al. (2015) aim to improve recommendation quality by using more complete *user* data. Kayaalp et al. (2009) and Bostandjiev et al. (2012) use multiple sources for data concerning items in the context of recommender systems. Here, Kayaalp et al. (2009) focus on the technical challenges arising from the integration of heterogeneous event data types for recommender systems and do not discuss the impact of completeness of item content data on recommendation quality. Bostandjiev et al. (2012) apply different recommender systems on each data source separately. Their resulting recommendation is the aggregation of the recommendations based on each single data source. Therefore, works in category B do not aim at an explanatory analysis or refer to a theoretical model to study whether recommendation quality is influenced by adding features and feature values. The works in category C focus on the impact of data characteristics on recommendation quality. While the majority of works study impact of data characteristics (meta-features) of rating data, only Fortes et al. (2017) investigate data characteristics in relation to item content data. They enhance the recommender system by including these data characteristics directly in the recommender algorithm as they aim for a predictive analysis. In contrast to the discussed works, which either focus on the consideration of *rating* data characteristics (e.g., entropy of rating distribution) or generate recommendations in a predictive analysis, we extend this category of works in two ways. Firstly, we explicitly investigate the impact of completeness of *item content* data on prediction accuracy. Secondly, we conduct an explanatory analysis based on causal hypotheses and a theoretical model, which strongly differs from predictive analytics (Shmueli and Koppius 2011). Both aspects have important implications in practice as the actual relevance of increasing the amount of available features and feature values for prediction accuracy is examined.

Theoretical Model

This subsection presents a theoretical model constituting a basis for the hypotheses discussed in the subsequent section. Research in the field of data quality shows an increasing tendency to study the impact of data quality of data views and data values (independent variable) on different evaluation criteria of decision support systems such as decision quality or data mining outcome (dependent variable) (e.g., Bharati and Chaudhury 2004; Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Woodall et al. 2015). More precisely, Blake and Mangiameli (2011) analyze the impact of the data quality dimensions accuracy, completeness, consistency and currency on data mining results in order to support decision-making. Woodall et al. (2015) investigate the impact of completeness on classification outcomes used for supporting users in their decision process. Bharati and Chaudhury (2004) examine the effects of accuracy, completeness and currency on the ability of an online analytical processing system to sustain decision-making. Ge (2009) focuses on accuracy, completeness and consistency and their impact on decision quality. Feldman et al. (2018) propose an analytical framework to investigate the impact of incomplete data sets on a binary classifier that serves for decision support.

The focus of these papers is to investigate in which way and to what extent the quality of data views and data values, especially the dimension completeness, influences evaluation criteria such as data mining outcome of particular decision support systems. Because recommender systems are a relevant category of decision support systems, especially in electronic markets, assisting users that face decision-making problems (Porcel and Herrera-Viedma 2010; Power et al. 2015), we derive the theoretical model from these works to examine the impact of completeness of item content data on prediction accuracy of recommender systems. Figure 1 presents this theoretical model.

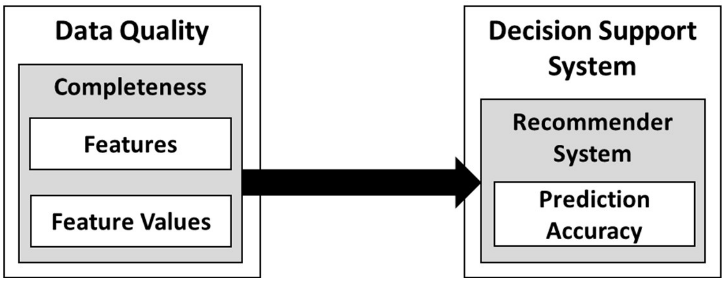

Figure 1. Theoretical Model

In the context of decision support systems, completeness is a frequently investigated dimension of data quality (Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Woodall et al. 2015). These works refer to completeness as the amount of available data views and data values. We take up this idea in the theoretical model and consider completeness by the amount of features and their feature values (cf. left side of Figure 1). As discussed above, features such as *Cuisine* can have multiple feasible feature values such as *Italian*, *American* or *Mexican*, which are assigned to items in order to describe and underline their characteristics enabling a more complete view on these items. Therefore, we focus on such features and their feature values when analyzing completeness. Similar to Bharati and Chaudhury (2004) and Ge (2009), the presented theoretical model in Figure 1 indicates a direct relation between data quality and evaluation criteria of decision support systems. In particular, the theoretical model suggests this relation between completeness of item content data and prediction accuracy of recommender systems (cf. right side of Figure 1). This model constitutes the foundation for the following hypotheses and is customized by different moderator variables to allow for a detailed analysis.

Hypotheses

Based on the theoretical model, we present ten hypotheses to address our research questions. Each hypothesis examines the impact of completeness of item content data on prediction accuracy from a different angle. Figure 2 at the end of this section shows an overview of all hypotheses.

Content-based and hybrid recommender systems, two major categories of recommender systems (Ning et al. 2015), operate on item content data to propose items to users that they are likely to be interested in (Lops et al. 2011). For this kind of data, increased completeness means that more features and/or more feature values are assigned to items (cf. Section “Theoretical Model”). Thus, increased completeness in this sense can be achieved in two ways: First, by adding features and their feature values to the feature set. For instance, a feature *Actors* can be added to the feature set for the movie domain. Second, by filling up missing feature values. For example, an already available feature *Parking Information* stating the parking options of a restaurant may have missing values for some restaurants which can be filled up. This can be done in various ways, for example by surveys, analyses or imputation (cf. Section “Description and Preparation of Data Sets”). Hence, all following hypotheses address both ways of increasing completeness in correspondence with our research questions *RQa* and *RQb*. Hypotheses labelled “a” focus on completeness increased by adding features and their feature values, whereas hypotheses labelled “b” focus on completeness increased only by filling up missing feature values. For both types of hypotheses, we test whether an increase in prediction accuracy can be observed.

This discussion leads to the following first two hypotheses:

> **H1a:** Adding features and their feature values leads to higher prediction accuracy.
>
> **H1b:** Filling up missing feature values leads to higher prediction accuracy.

Hypothesis H1a pursues the idea that the preferences of users can be analyzed in more detail when more item features and their feature values are available and suggests that the prediction accuracy (assessed by RMSE, Precision, Recall and F1-measure; cf. Section “Assessing Prediction Accuracy”) is thus higher. Hypothesis H1b follows the expectation that recommendations are more accurate when missing values of item features are filled up.

Depending on the analysis of Hypotheses H1a/b, it is further interesting whether the extent of increased completeness measured *per item*, *user* or *feature* influences the extent of increased prediction accuracy. Regarding items and users, this can be described more precisely as follows: Does the increase in the amount of additional features and feature values (type “a”) or the increase in the amount of filled up feature values (type “b”) positively moderate the impact of completeness on prediction accuracy for an item or a user?

Therefore, it is meaningful to examine moderator effects regarding users and items on the relationship between completeness and prediction accuracy. This discussion leads to further hypotheses, which consider the increase in the amount of additional features and feature values, respectively, the increase in the amount of filled up feature values, per item or per user. Beginning with items, we examine the following hypotheses:

> **H2a:** The increase in the amount of additional features and their feature values *for an item* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H2b:** The increase in the amount of filled up feature values *for an item* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Analogously, we formulate the hypotheses regarding the increase in completeness for users as follows:

> **H3a:** The increase in the amount of additional features and their feature values *regarding a user* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H3b:** The increase in the amount of filled up feature values *regarding a user* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Similar to items and users, it appears reasonable that the extent of increased completeness *per feature* also influences the extent of increase in prediction accuracy. Consequently, the following hypotheses examine the moderator effect regarding features on the relationship between completeness and prediction accuracy.

At first, we focus on a higher *amount* of values of added or filled up features, respectively, which leads to the following two hypotheses:

> **H4a:** The increase in the amount of feature values *for an additional feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

H4b: The increase in the amount of feature values *for a filled up feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Finally, we focus on increased completeness through higher *diversity* of added or filled up features. Additional features may have similar feature value assignments for items as already existing features. In particular, adding a feature, which has exactly the same feature values for items as an existing feature, may not influence the prediction accuracy at all, since such a feature does not add any further diversity to the item content data (Mitra et al. 2002; Tabakhi and Moradi 2015). In contrast, adding features that provide a high diversity to the item content data enhance the recommender system’s ability to differentiate items and users and thus may lead to a high increase in prediction accuracy. Therefore, we consider the following hypotheses expecting a moderator effect when adding or filling up features depending on their diversity:

H5a: The diversity *for an additional feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

H5b: The diversity *for a filled up feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Figure 2 customizes the theoretical model (cf. Figure 1) by incorporating moderator variables and the stated hypotheses. In general, it shows the expected impact of the data quality dimension completeness on the prediction accuracy as stated by Hypotheses H1a/b. Additionally, it illustrates the hypotheses examining a moderating effect for the increase in completeness per item (Hypotheses H2a/b), per user (Hypotheses H3a/b) and per feature (Hypotheses H4a/b, H5a/b).

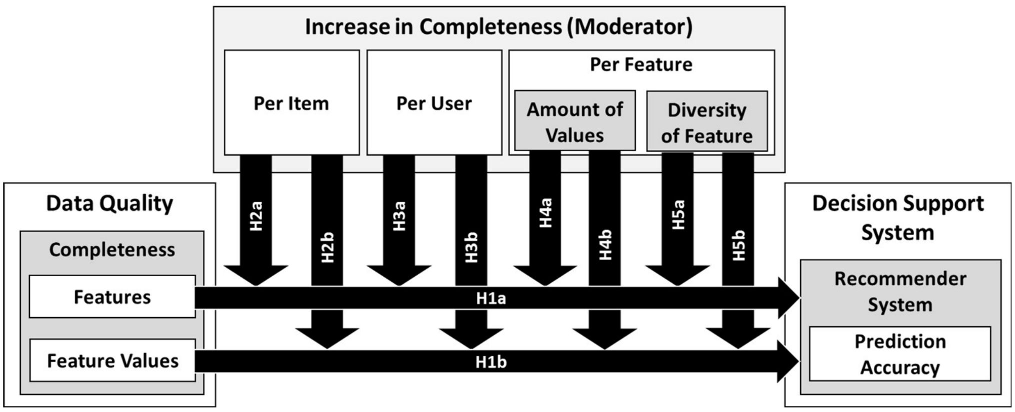

Figure 2. Overview for Hypotheses H1-H5

Methodology

In this section, we introduce the models used to test Hypotheses H1-H5. To do so and to assess prediction accuracy as the dependent variable, we first discuss selected measures which allow differentiated analyses and interpretations regarding the impact on prediction accuracy. Thereafter, we describe the testing methodology for Hypotheses H1a/b as well as the regression models for testing Hypotheses H2-H5.

Assessing Prediction Accuracy

To enable a detailed and careful analysis of the results of the Hypotheses H1-H5, we assessed prediction accuracy by means of different measures from literature, namely RMSE, Precision, Recall and F1-measure (Gunawardana and Shani 2015). RMSE as shown in Equation (1) is one of the most popular measures for assessing prediction accuracy (Gunawardana and Shani 2015) and is defined by the term

$$RMSE = \sqrt{\frac{1}{|T|} \cdot \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2}, \tag{1}$$

where T is a test set of user-item pairs (u, i) for which the ratings $\hat{r}_{ui}$ are predicted by the recommender system and the actual ratings r_{ui} are known. RMSE received special attention by the Netflix Prize Challenge in 2006 (Koren 2009). Its main characteristic is that higher errors (i.e., the difference between predicted and actual rating) are weighted stronger through its quadratic structure than lower errors. Further, usually the predicted ratings $\hat{r}_{ui}$ are continuous (real-valued) and the actual ratings r_{ui} are discrete (and ordered). Hence, minor RMSE value changes may not result in a different mapping (by rounding) of the continuous predicted rating $\hat{r}_{ui}$ to a discrete star rating $\widehat{dr}_{ui} \in \{1, \ldots, 5\}$. This means that the mapping to a discrete star rating may not change, even with an improved RMSE value. Therefore, it is also necessary to assess whether the mapping of continuous predicted ratings $\hat{r}_{ui}$ to discrete star ratings $\widehat{dr}_{ui}$ changes or improves with the increase in completeness and the expected increase in prediction accuracy. To evaluate this, Precision, Recall, and F1-measure are the most important measures. These measures assess whether or not the predicted rating level $\widehat{dr}_{ui}$ exactly coincides with the actual true rating level r_{ui} for each user-item pair (u, i) (Aggarwal 2014). Precision and Recall are calculated as the average of the Precision and Recall values for each star rating level $k \in \{1,2,3,4,5\}$, which are given by the following terms.

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{2}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{3}$$

Here, TP_k is the number of user-item pairs (u, i) with $r_{ui} = k$ and $\widehat{dr}_{ui} = k$ (“true positives”), FP_k as shown in Equation (2) is the number of user-item pairs (u, i) with $r_{ui} \neq k$ and $\widehat{dr}_{ui} = k$ (“false positives”), and FN_k as shown in Equation (3) is the number of user-item pairs (u, i) with $r_{ui} = k$ and $\widehat{dr}_{ui} \neq k$ (“false negatives”). F1-measure as shown in Equation (4) is then given by the harmonic mean of Precision and Recall

$$F1 = 2 \cdot \frac{Precision \ \cdot \ Recall}{Precision \ + \ Recall}. \tag{4}$$

The main difference in interpretation of these measures is that the Precision, Recall and F1-measure focus on correct or incorrect mappings of predicted and actual star ratings while ignoring the (real-valued) error size, which is in the focus of RMSE.

Model for Hypotheses H1a/b

Each of the Hypotheses H1a and H1b focuses on a comparison of the prediction accuracy of two item content data sets, one data set without increased completeness and the other data set with increased completeness (cf. Figure 3).

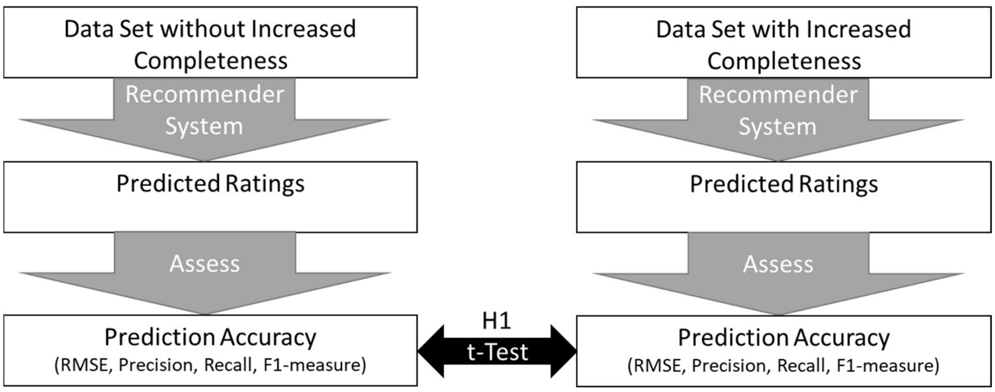

Figure 3. Testing Hypotheses H1a/b

In both cases, we initially do not consider any moderator variable. To test the significance of both hypotheses, we used the paired Student’s t-test, a broadly applied test in the evaluation of recommender systems to compare the results of two different settings, while in both settings the considered set of user ratings remains the same (Shani and Gunawardana 2011). More precisely, the t-test was used to compare each of the measures RMSE, Precision, Recall and the F1-measure (and thus the prediction accuracy) based on the data set with increased completeness (i.e., when adding features and their feature values or when filling up missing feature values) and based on the data set without increased completeness.

Model for Hypotheses H2-H5

The Hypotheses H2-H5 analyze whether the increase in completeness per item, user or feature moderates the impact of completeness on the increase in prediction accuracy caused by adding features and their feature values (hypotheses of type “a”) or by filling up missing feature values (hypotheses of type “b”). This means that the tests of the Hypotheses H2-H5 are organized in a similar way. Therefore, we describe the general structure for all of these tests in the following.

To test moderator effects on the impact of completeness on increased prediction accuracy, we chose moderated regression analysis (cf., e.g., Cohen et al. 2003; Dawson 2014; Hayes 2013; Helm and Mark 2012) as it is a widespread statistical tool to test whether the relationship between two variables is dependent on a third variable (the moderator). The underlying regression model is represented by the equation

$$y = b_0 + b_1 \cdot x + b_2 \cdot z + b_3 \cdot x \cdot z. \tag{5}$$

Here, y is the dependent or endogenous variable (criterion), x is the independent or exogenous variable (predictor) and z is the moderator variable. Regarding Hypotheses H2-H5, the endogenous variable y constitutes the (expected) increase in prediction accuracy measured by RMSE, Precision, Recall and F1-measure while the exogenous variable x indicates whether the data set with increased completeness or the data set without increased completeness is used. The moderator variable z constitutes the increase in completeness. More precisely, for H2a, H3a and H4a, the variable z represents the increase in additional features and feature values and for H2b, H3b and H4b, the variable z represents the increase in filled up feature values. Similar, for H5a, the variable z represents the diversity of added features, and for H5b, the variable z represents the diversity of filled up features.

Besides the common interpretation of the coefficient b_0 as well as the coefficients b_1 and b_2 (*first order effects* of the regression model), the product term $x \cdot z$ and its coefficient b_3 are of special interest. This term represents the interaction (moderation) of two variables. More precisely, the coefficient b_3 estimates how much the slope of x changes as z changes. This represents how much the impact of increased completeness on prediction accuracy is influenced by the (different) values of the moderator variable. Therefore, a hypothesis proposing a moderator effect can be supported, if there is evidence that b_3 is different from zero with a certain level of significance.

In case of a moderator effect, the strength of this effect can be assessed by Cohen's f^2. Here, the coefficient of determination regarding the regression model depicted in Equation (5) is compared to the coefficient of determination of the regression model without the interaction term, which means,

$$y = b_0 + b_1 \cdot x + b_2 \cdot z. \tag{6}$$

Denoting the coefficient of determination R^2 according to each Equation (5) and (6) (i.e., R_1^2 and R_2^2), Cohen's f^2 is given by the term

$$f^2 = \frac{R_1^2 - R_2^2}{1 - R_1^2}. \tag{7}$$

Cohen's f^2 measures the relative increase in the explained variance of y when adding the interaction term to Equation (6) as shown in Equation (5). In Cohen (1988) the values 0.02, 0.15 and 0.35 are suggested for f^2 to indicate small, medium or large moderator effect sizes, which is critically discussed in scientific literature (Aguinis et al. 2005; Gignac and Szodorai 2016; Helm and Mark 2012). For instance, Aguinis et al. (2005) conducted a review of 261 articles published in several journals (maintaining high methodological standards) in order to analyze the size of moderating effects. They found that the mean of Cohen's f^2 was about 0.009 (with a standard deviation of 0.025), and the median about 0.002 with a positively skewed distribution (skewness = 6.52). This indicates that – regarding the suggested values of Cohen (1988) – a medium or strong moderator effect can be rarely attained. In their discussion, they encourage researchers to "plan future research designs based on smaller (and more realistic) targeted effect sizes" (Aguinis et al. 2005) as long as the observed effect has a meaningful impact and interpretation for science and practice.

Evaluation

In this section, we outline the test procedure and results of our empirical evaluation. Initially, we describe both used real-world data sets. Afterwards, we introduce the recommender system which was applied to these data sets and outline in detail how we tested each hypothesis. We conclude the section by presenting the results of these tests.

Description and Preparation of Data Sets

For testing our hypotheses, we prepared two real-world data sets. While the first data set contains a large number of user-generated ratings about restaurants and was retrieved from two leading advertising web portals, the second data set is based on the non-commercial movielens data set containing approximately one million ratings (Harper and Konstan 2015). In both data sets, the ratings are assessments of items by users and hence, each rating corresponds to exactly one user and one item. Further, the rating values are given on an ordinal, five-tier scale of stars, ranging from 1 star to 5 stars.

Restaurant Data Set

In the first data set, one portal (Portal 1) focuses on local businesses such as bars or restaurants and provided over 100 million ratings by 2018. The second portal (Portal 2) specializes on travel opportunities and businesses such as restaurants providing over 400 million ratings by 2018. Since each web portal provided a vast amount of data, we focused on an excerpt and chose rating data of restaurants from the area of New York City, USA, because the high number of restaurants in this area allows for testing each hypothesis on a sufficiently high number of items, users or features, respectively. This led to a data set with more than 2.2 million ratings provided by over 550,000 users on more than 18,500 restaurants from Portal 1 and more than 720,000 ratings from about 375,000 users for more than 8,600 restaurants from Portal 2. Table 1 describes the restaurant data set.

	Portal 1	**Portal 2**
# of Users	556,462	374,960
# of Restaurants	18,507	8,631
# of Ratings	2,252,224	721,416

Table 1. Description of the Restaurant Data Set

Both web portals provide features such as *Cuisine* with multiple feasible feature values such as *Italian*, *American* or *Mexican*. In both portals, these feature values are assigned to an item. Other features of restaurants are *Special Diets* with feature values such as *Vegetarian, Vegan* or *Gluten-free* and *Type of Establishment* with feature values such as *Café, Bistro* or *Bar*. With this in mind, the knowledge about feature value assignments is especially relevant for each item in this data set. In the case that a feature value is unknown, we indicated the missing feature value by the value *N/A* (not available).

From Portal 1 we retrieved an item content data set with 13 different features, denoted by P1, while Portal 2 provided an item content data set with 12 different features, denoted by P2. As only Portal 1 yielded features containing missing values, we split up P1 into an item content data set P1.1, containing only the seven features without missing values, and an item content data set P1.2, containing only the six features with missing values. More precisely, 44% of all possible 425,661 feature values for the six features of P1.2 were not available for the 18,507 restaurants of Portal 1. Table 2 illustrates the features and feature values per portal.

	Portal 1		Portal 2
Item Content Data Set	**P1**		**P2**
	P1.1	**P1.2**	
# of Features	7	6	12
# of Missing Feature Values	0 (0%)	189,164 (44%)	0 (0%)

Table 2. Features and Feature Values provided by the two Web Portals of the Restaurant Data Set

Data sets for hypotheses of type "a"

To prepare the data set for testing the hypotheses of type "a", we focused on the features from P1.1 from Portal 1 and P2 from Portal 2 that did not contain any missing data. This was important in order to carefully separate hypotheses of type "a" and of type "b". To obtain the joint feature set for a restaurant from the item content data sets P1.1 and P2, it was necessary to match restaurants between both portals. We thus conducted record linkage, which is the task of identifying records that refer to the same entity across different data sources (Christen 2012). To do so, we used a common rule-based classification model. The model was built using manually labelled training data and evaluated by quality measures. The classification resulted in 5,367 restaurants matching across the two portals with a false discovery rate below 1% on manually labelled test data. This means that less than 1% of these restaurants were incorrectly classified as matching. We exclusively focused on such matching restaurants to test the hypotheses of type "a" because these restaurants had added features compared to the features in each single portal. Furthermore, for each portal, we considered users with more than 30 ratings in order to only evaluate users with a substantial number of ratings (Sarwar et al. 2002). To increase completeness, features from Portal 2 were added to the feature set of Portal 1 and vice versa. This resulted in two cases used for testing the hypotheses of type "a": The data for the first case originated from Portal 1, consisted of 5,367 items with 367,182 ratings of 8,138 users and was evaluated using the item content data sets P1.1 as baseline and P2 as set of additional features and their feature values. The data for the second case originated from Portal 2, comprised the same 5,367 items with 20,659 ratings of 505 users and was evaluated using the item content data sets P2 as baseline and P1.1 as set of additional features (cf. Table 3).

Data sets for hypotheses of type "b"

To prepare data for testing the hypotheses of type "b", we focused on the first portal, as the second portal did not provide any features with missing values. In this case, to fill up missing feature values in the item content data set P1.2 containing six features, we used the common nearest neighbor imputation technique (Enders 2010). Similar to above, this imputation was evaluated by means of training and test data as well as quality measures. Missing values were imputed with a mean absolute error of only 0.299 for the test data. Again, we considered users with more than 30 ratings. This led to the data for testing the hypotheses of type "b" consisting of 18,507 restaurants with 731,395 ratings of 10,556 users, which was evaluated comparing the item content data sets P1.2 as baseline (consisting of 236,497 feature values) and P1.2' as set of baseline features with filled up feature values (consisting of 425,661 feature values including the 189,164 filled up feature values) (cf. Table 3).

	Hypotheses of Type "a" originating from Portal 1		**Hypotheses of Type "a"** originating from Portal 2		**Hypotheses of Type "b"** originating from Portal 1	
Item Content Data Set	**P1.1** (Baseline)	**P1.1&P2** (Baseline & add. features)	**P2** (Baseline)	**P1.1&P2** (Baseline & add. features)	**P1.2** (Baseline)	**P1.2'** (Baseline & filled up feature values)
# of Features/ # of Feature Values	7	19	12	19	236,497	425,661
# of Items	5,367		5,367		18,507	
# of Ratings	367,182		20,659		731,395	
# of Users	8,138		505		10,556	

Table 3. Description of the Data Bases for Evaluating Hypotheses H1a/b-H5a/b on the Restaurant Data Set

Movie Data Set

The second data set focuses on movies and originates from the research lab grouplens, which provides data sets with up to 20 million ratings from the non-commercial web portal movielens by 2016. Since the movielens data sets have been updated since 1998, new features and feature values have been added in new versions. To enable an evaluation based on a larger amount of ratings, we consider the data set from 2003 with only one feature and its most recent version from 2016 with five additional features and their feature values. The old version (OldV) of the movielens data set from 2003 contains over one million ratings provided by over 6,000 users on approximately 3,900 movies, while the new version (NewV) consists of over 20 million ratings from about 140,000 users for more than 27,000 movies.

	OldV	**NewV**
# of Users	6,040	138,493
# of Movies	3,883	27,278
# of Ratings	1,000,209	20,000,263

Table 4. Description of the Movie Data Set

Table 4 describes the movie data set. Similar to the restaurant data set, both versions of the movielens data set provide the feature *Genre* with multiple feasible feature values such as *Comedy*, *Drama* or *Thriller*, while the new version provides additional features and their feature values such as *Actors* and *Country of Origin* each with according feature values. For example, the additional feature *Actors* in the version NewV indicates the top billed actors of the movie cast. Both versions do not yield features containing missing values, which means that only hypotheses of type “a” could be tested on the movie data set. Table 5 illustrates the features and feature values per version.

Item Content Data Set	**OldV**	**NewV**
# of Features	1	6
# of Missing Feature Values	0 (0%)	0 (0%)

Table 5. Features and Feature Values provided by the two Versions of the Movie Data Set

Data set for hypotheses of type “a”

Since the movie data set consists of an old and a new version, it is clear that the baseline item content data set is given by the old version and the item content data set with increased completeness is given by the union of both versions. Similar to the restaurant data set, the joint feature set for a movie was obtained by matching movies between both versions. As the movielens identifiers of the movies did not change between both versions (except from 24 movies, which were removed), record linkage was easy to conduct. Furthermore, the 6,040 users in both versions had at least 20 ratings, enabling a substantial number of ratings for the evaluation. Since 24 movies and their corresponding 2,175 ratings had been removed in the new version NewV, this resulted in content data sets consisting of 3,859 items with 998,034 ratings of 6,040 users and was evaluated using the item content data sets OldV as baseline and NewV as set of additional features and their feature values (cf. Table 6).

	Hypotheses of Type “a” originating from the old version of the movielens data set	
Item Content Data Set	**OldV** (Baseline)	**OldV&NewV** (Baseline & add. features)
# of Features	1	6
# of Items	3,859	
# of Ratings	998,034	
# of Users	6,040	

Table 6. Description of the Data Bases for Evaluating Hypotheses H1a-H5a on the Movie Data Set

Used Recommender System

For our evaluation, we used the hybrid recommender system approach *Content-Boosted Matrix Factorization* (CBMF) as presented by Forbes and Zhu (2011) and Nguyen and Zhu (2013). Matrix factorization approaches became very popular by the contest on the Netflix Grand Prize, which started 2006 and ended 2009 (Koren et al.

2009; Koren 2009). They are now state-of-the-art models in the research of recommender systems (Kim et al. 2016; Ning et al. 2017; Symeonidis 2016). CBMF is able to utilize both non-content data (ratings) and, in particular, content data (features and feature values of items). Like all matrix factorization models, CBMF models are learned by optimization and therefore, preliminary steps such as feature weighting or feature selection are not necessary for CBMF (Koren et al. 2009; Nguyen and Zhu 2013).

CBMF learns a d-dimensional vector of latent factors $p_u \in \mathbb{R}^d$ for each user u and a d-dimensional vector of latent factors $a_f \in \mathbb{R}^d$ for each feature f, such that the actual rating r_{ui} for a user-item pair (u, i) is approximated by the predicted star rating $\hat{r}_{ui} = p_u^T q_i$, with $q_i = \sum_{f \in F_i} a_f$ and F_i being the set of features that are assigned to item i. In our evaluation, we used the default configuration for CBMF as described, for instance, by Nguyen and Zhu (2013). Excepting this default configuration concerns the regularization penalty factor λ, which has to be adjusted depending on the data set (Koren et al. 2009). Thus, to determine this factor we conducted cross-validation tests as described by Koren et al. (2009). For instance, the value $\lambda = 10 \times 10^{-6}$ (cf. Figure 4) yielded the best results on test data from Portal 1 regarding the RMSE. All other parameter configurations were adopted from Nguyen and Zhu (2013).

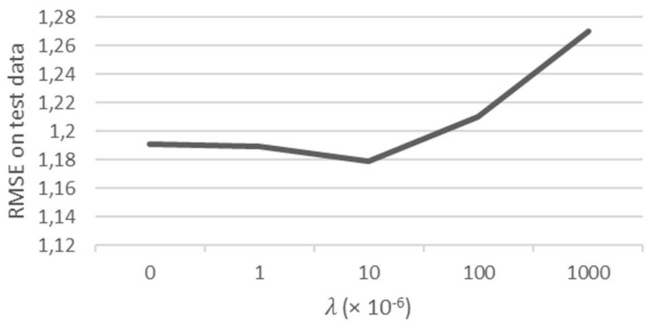

Figure 4. RMSE on the Test Data Depending on the Regularization Penalty Factor λ

Test Procedure and Results

For our evaluation, we split ratings into 50% training data for learning the CBMF model and 50% test data for assessing the prediction accuracy. On the one hand, dividing the data in half at random allowed to obtain a large *test* set (cf. also Nguyen and Zhu 2013), which is important for meaningful results when testing hypotheses. On the other hand, because of the large real-world data sets, 50% *training* data allowed us to learn the CBMF model.

After that, we utilized the recommender system for each pair of item content data sets (with and without increased completeness) to predict ratings and assess the corresponding prediction accuracy. The increase in prediction accuracy assessed separately by Precision, Recall and F1-measure was determined by subtracting the prediction accuracy based on the baseline content from the prediction accuracy based on the content with increased completeness. As lower RMSE values indicate more accurate predictions, the negative difference was used in this case, accordingly.

A requirement for evaluating Hypotheses H1a/b using Student's t-test is that sample groups should be normally distributed. Because of the large sample size in our evaluation, this requirement is obviously met (Boneau 1960). For evaluating moderator effects in Hypotheses H2-H5, we examined whether the selection of the linear regression model is appropriate or whether non-linear, for instance, quadratic regression models should be preferred (i.e., a curvilinear moderator effect is expected). Therefore, to test for potential non-linear moderator effects, we compared the fitness of the quadratic (non-linear) model and the linear model relying on the frequently discussed and used Bayesian Information Criterion (BIC) for model selection (cf. Schwarz 1978), for which smaller BIC values indicate the preferred model. These tests yielded almost the same BIC values for both models. For instance, for the first Hypothesis H2a the BIC value for the linear model was -5,464 and for the quadratic model -5,446 (e.g., regarding the measure Precision) and for the last Hypothesis H5b the BIC value for the linear model was -155 and for the quadratic model -148. Since the quadratic model did not or hardly improve the BIC values, the linear model was used because of its lower complexity, as suggested by literature (Cohen et al. 2003; MacCallum and Mar 1995).

The moderator variable for Hypotheses H2a/b was operationalized by the number of added or filled up feature value assignments *per item* (cf. Blake and Mangiameli 2011) relative to the number of feature value assignments *per item* in the baseline content data set. For Hypotheses H3a/b, the mean of the aforementioned operationalization across all rated items of *a user* was used as the moderator variable. In a similar way, the moderator variable for Hypothesis H4a was operationalized by the number of added feature value assignments for *a feature* relative to the number of feature value assignments in the baseline content data set. Hypothesis H4b was operationalized by the number of filled up feature value assignment for *a feature* relative to the number of feature value assignments for *this feature* in the baseline data set. The moderator for Hypotheses H5a/b was assessed by the mean cosine distance between the added/filled up features and the baseline features (Mitra et al. 2002; Tabakhi and Moradi 2015). Summing up the above, each operationalization of the moderator variables shares a similar concept as it was determined as the increase in completeness relative to the baseline content. Furthermore, we used the two standard levels of significance 0.01 (indicated by '**') and 0.05 (indicated by '*') for the tests of all hypotheses (e.g., Shani and Gunawardana 2011).

In the following, we outline the evaluation results. In particular, we present the impact on prediction accuracy for all tests, which means, the values for each measure (RMSE, Precision, Recall and F1-measure), their relative increase in prediction accuracy and the significance of the t-values in case of H1a/b and the significance of the regression coefficients together with the effect sizes in case of H2-H5. Table 7 shows the results of our evaluation for the first two hypotheses: Hypotheses H1a and H1b can be supported with positive t-values and statistical significance by p-values below 0.01. This means that both adding features and their feature values as well as filling up missing feature values lead to significantly higher prediction accuracy as indicated by each of the evaluation measures in Table 7.

Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Prediction Accuracy** (RMSE/Precision/ Recall/F1) (Without Increased Completeness)	**Prediction Accuracy** (RMSE/Precision/ Recall/F1) (Increased Completeness)	**Relative Increase in Prediction Accuracy** (RMSE/Precision/ Recall/F1)	**Corresponding t-Values** (*:p-value<0.05; **:p-value<0.01)	**Hypo-thesis can be supported**
H1a (Portal 1)	P1.1 vs. P1.1&P2	1.57/0.216/ 0.218/0.217	1.18/0.246/ 0.231/0.238	25%/14%/ 6%/10%	164**/63**/ 63**/63**	**Yes** (by all)
H1a (Portal 2)	P2 vs. P1.1&P2	1.29/0.236/ 0.235/0.235	1.20/0.249/ 0.246/0.247	7%/6%/ 5%/5%	17**/5**/ 5**/5**	**Yes** (by all)
H1a (movie-lens)	OldV vs. OldV&NewV	1.67/0.226/ 0.228/0.227	0.95/0.443/ 0.315/0.368	43%/96%/ 38%/62%	413**/185**/ 185**/185**	**Yes** (by all)
H1b (Portal 1)	P1.2 vs. P1.2’	1.60/0.227/ 0.221/0.224	1.04/0.332/ 0.225/0.268	35%/46%/ 2%/20%	269**/112**/ 112**/112**	**Yes** (by all)

Table 7. Results for Hypotheses H1a/b

The results of the hypotheses with regard to items and users are given in Table 8: Hypotheses H2a and H2b can also be supported with statistical significance by p-values below 0.01. This means that for items both the amount of additional features and their feature values and the amount of filled up feature values are positive moderators. In other words, items that obtain a stronger increase in completeness can then be recommended at a significant higher level of accuracy than before (cf. Table 8). For Hypotheses H3a and H3b, focusing on users instead of items, the test results were as follows: Hypothesis H3a in the case of Portal 1 and movielens as well as Hypothesis H3b can be supported with statistical significance by p-values below 0.01 (except for the case of the measure Precision for H3b, where the p-value was between 0.01 and 0.05). The test of Hypothesis H3a in the case of Portal 2 yielded a p-value below 0.01 only for the measure RMSE, but p-values above 0.05 for the measures Precision, Recall and F1-measure. Hence, Hypothesis H3a cannot be supported for all measures in the case of Portal 2. Except from that, Hypothesis H3 can be supported in the case of Portal 1 and movielens with statistical significance at the level 0.05. Therefore, it can be concluded that both the amount of additional features and their feature values and the amount of filled up feature values each measured per user are also positive moderators of the impact of completeness on prediction accuracy assessed by RMSE and, except H3a (Portal 2), on prediction accuracy assessed by Precision, Recall and F1-measure. This means that users, whose rated items obtain a stronger increase in completeness, benefit the most and that recommendations for these users are significantly more accurate than before.

Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Interaction Coefficients b_3 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1** (*:p-value<0.05; **:p-value<0.01)	**Cohen's f^2 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1**	**Hypothesis can be supported**
H2a (Portal 1)	P1.1 vs. P1.1&P2	0.06**/0.02**/ 0.01**/0.01**	0.024/0.014/ 0.002/0.008	**Yes** (by all)
H2a (Portal 2)	P2 vs. P1.1&P2	0.12**/0.02**/ 0.02**/0.02**	0.042/0.001/ 0.001/0.001	**Yes** (by all)
H2a (movielens)	OldV vs. OldV& NewV	0.03**/0.01**/ 0.001**/0.004**	0.032/0.015/ 0.001/0.011	**Yes** (by all)
H2b (Portal 1)	P1.2 vs. P1.2'	0.24**/0.03**/ 0.02**/0.02**	0.436/0.019/ 0.009/0.015	**Yes** (by all)
H3a (Portal 1)	P1.1 vs. P1.1&P2	0.08**/0.02**/ 0.01**/0.01**	0.013/0.002/ 0.001/0.001	**Yes** (by all)
H3a (Portal 2)	P2 vs. P1.1&P2	0.18**/-0.01/ 0.00/-0.01	0.015/-/ -/-	**Only for RMSE measure**
H3a (movielens)	OldV vs. OldV& NewV	0.02**/0.004**/ 0.004**/0.004**	0.018/0.003/ 0.003/0.005	**Yes** (by all)
H3b (Portal 1)	P1.2 vs. P1.2'	0.39**/0.01*/ 0.02**/0.01**	0.094/0.0003/ 0.001/0.001	**Yes** (by all)

Table 8. Results for Hypotheses H2a/b and H3a/b

The results of Hypotheses H4a/b and H5a/b are given in Table 9. Hypothesis H4a can be supported with statistical significance by p-values below 0.01, whereas Hypothesis H4b cannot be supported indicated by negative coefficients b_3. In other words, only the amount of additional features and their feature values is a positive moderator of the impact on prediction accuracy (H4a), but not the amount of filled up feature values (H4b). Hypotheses H5a/b cannot be supported as indicated by negative coefficients or by p-values above 0.05. This suggests that the diversity for an additional feature or for a filled up feature is not a positive moderator.

Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Interaction Coefficients b_3 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1** (*:p-value<0.05; **:p-value<0.01)	**Cohen's f^2 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1**	**Hypothesis can be supported**
H4a (Portal 1)	P1.1 vs. P1.1&P2	0.40**/0.02**/ 0.02**/0.02**	1.221/0.611/ 0.628/0.645	**Yes** (by all)
H4a (Portal 2)	P2 vs. P1.1&P2	0.27**/0.09**/ 0.08**/0.09**	0.363/0.236/ 0.162/0.198	**Yes** (by all)
H4a (movielens)	OldV vs. OldV&NewV	0.70**/0.10**/ 0.04**/0.07**	1.657/0.665/ 0.233/0.575	**Yes** (by all)
H4b (Portal 1)	P1.2 vs. P1.2'	-0.01/0.00/ 0.00/0.00	-/-/ -/-	**No** (by all)
H5a (Portal 1)	P1.1 vs. P1.1&P2	-1.94**/-0.11**/ -0.10**/-0.11**	0.487/0.297/ 0.367/0.338	**No** (by all)
H5a (Portal 2)	P2 vs. P1.1&P2	-0.02**/-0.01**/ -0.01**/-0.01**	0.040/0.051/ 0.038/0.044	**No** (by all)
H5a (movielens)	OldV vs. OldV&NewV	-0.76**/-0.12**/ -0.07**/-0.09**	0.352/0.280/ 0.280/0.367	**No** (by all)
H5b (Portal 1)	P1.2 vs. P1.2'	-0.43*/-0.05/ 0.00/-0.02	0.155/-/ -/-	**No** (by all)

Table 9. Results for Hypotheses H4a/b and H5a/b

Discussion and Implications

In general, the results support the theoretical model serving as foundation of the tested hypotheses, which means, the completeness of item content data has a significant positive impact on the prediction accuracy of recommendations. More precisely, adding features and their feature values (Hypothesis H1a) or filling up missing feature values (Hypothesis H1b) leads to higher prediction accuracy. Besides this general finding, we also examined moderator effects on the impact of completeness on prediction accuracy (Hypotheses H2-H5). Thereby, the results reveal some interesting findings. While the increase in completeness per item and per user are positive moderators of the impact of completeness on prediction accuracy (Hypotheses H2a/b and H3a/b, except for Hypothesis H3a and Portal 2, which will be discussed below), the same cannot always be examined for the increase in completeness per feature. In particular, adding features with a high amount of additional feature values leads to a higher increase in prediction accuracy (Hypothesis H4a). However, filling up missing feature values with a high amount of additional feature values does not lead to a higher increase in prediction accuracy (Hypothesis H4b). In addition, neither adding features (Hypothesis H5a) nor filling up missing values of features (Hypothesis 5b) with a high diversity leads to a higher increase in prediction accuracy, which constitutes a further interesting finding. In the following, we discuss each result in detail.

Both Hypotheses H1a and H1b are supported as indicated by t-values with positive sign and with p-values below 0.01. This means, as illustrated in Table 7, both adding features and their feature values as well as filling up missing feature values led to a considerable increase in prediction accuracy. After increasing completeness, the RMSE was between 7% and 43% lower than the RMSE before increasing completeness (corresponding to absolute decreases of RMSE between 0.09 and 0.72). Precision was between 6% and 96% higher, Recall was between 2% and 38% higher and F1-measure was between 5% and 62% higher. By a detailed consideration of the results for Hypotheses H1a/b, two interesting observations can be made. First, the relative increase in prediction accuracy is lower for H1a in the case of Portal 2 compared to all other cases of H1a. This may be due to the fact that the additional features only constitute less than 40% of all features of the item content data set with increased completeness in case of Portal 2 (7 of 19 features). In the other cases of H1a, the additional features constitute at minimum 60% of the features of the data set with increased completeness (12 of 19 features or 5 of 6 features). Second, the increase in prediction accuracy measured by RMSE and Precision is in almost all cases (considerably) higher than measured by Recall and F1-measure. In contrast to the discrete nature of the measures Precision, Recall and the F1-measure, the higher increase in prediction accuracy measured by RMSE may be reasoned by the fact that RMSE uses the predicted ratings as determined by the recommender system (i.e., as a continuous variable). Therefore, the errors between predicted and actual ratings are assessed by an interval-scaled difference. To analyze the high increases in Precision, we examined the results of H1a (movielens) in more detail, which shows the highest increase of Precision (+96%). Here, we found that, on the one hand, the *decreases* in the number of incorrect predictions (i.e., false positives) was the largest for the rating levels 1 star (-96%), 2 stars (-76%) and 5 stars (-60%). On the other hand, the largest *increases* in correct predictions (true positives) was achieved for the ratings levels 3 stars (+31%) and 4 stars (+207%). This means that by increasing completeness the used recommender system was less likely to incorrectly predict “extreme” ratings (i.e., very high or very low ratings) while mostly improving the correct prediction of “mainstream” ratings (the mean overall rating is 3.6). Hence, the Precision of most classes achieved a much higher increase than the Recall or F1-measure. In total, the results of Hypotheses H1a/b show that recommendations based on item content data sets with increased completeness are more accurate, which is valuable for achieving a high user satisfaction (Koren et al. 2009; Ricci et al. 2015). At this point, we want to emphasize that the increase in prediction accuracy is provided only by increasing data quality and not by enhancing the recommender algorithm. Nowadays, the aim of numerous works in the research field of recommender systems is to develop very sophisticated recommender algorithms in order to increase prediction accuracy (partly to a small extent). One seminal example is the winning solution of the Netflix Grand Prize, which decreased the RMSE by 10% through a very elaborate and complex enhancement and combination of multiple recommender algorithms (Koren 2009). Instead, our results show that devoting more importance to maintaining high data quality for recommender systems is also highly promising and may inspire further research.

For Hypotheses H2-H5 we focus on the coefficients b_3 regarding the moderated regression (cf. Equation (5)) as well as the corresponding effect sizes indicated by Cohen's f^2 (cf. Equation (7)). Here, in general, the absolute values of the coefficients b_3 are consistently higher when evaluating the RMSE compared to the other measures. This is due to the higher values for the RMSE as seen in Table 7, where the values for RMSE range from 0.95 to 1.67 while Precision, Recall and F1-measure take values between 0.216 and 0.443. Considering the results for Hypothesis H2b, for instance, the coefficient for the RMSE signifies that the RMSE based on increased completeness is lowered by 0.24 when the moderator variable is increased by one. In the same setting, the Precision would increase by only 0.03.

The evaluation results support Hypotheses H2a/b. As illustrated in Table 8, all coefficients b_3 of our evaluation were positive (ranging from 0.001 to 0.24) and significant (p-value<0.01). This finding shows that the amount of additional features and their feature values and the amount of filled up feature values *per item* has a significant moderator effect. The effect size indicated by Cohen's f^2 ranges from 0.001 to 0.436 (cf. Section "Model for Hypotheses H2-H5" for the interpretation of Cohen's f^2). By a detailed consideration of the results for Hypotheses H2a/b, three observations can be made. First, the evaluation measure RMSE showed the largest effect sizes. This is in accordance with the finding discussed above that prediction accuracy measured by RMSE shows the highest increase in general due to its continuous nature. Second, the effect sizes for Precision, Recall and F1-measure, especially for H2a (Portal 2), are small. This may be reasoned by similar arguments as the first observation and by the fact, that the additional features only constitute less than 40% of all features of the item content data set, as discussed above for H1a (Portal 2). Third, the effect size for Hypothesis H2b is relatively high. An analysis of the data indicated that items, which have many missing feature values, receive highly incorrect rating predictions based on the data set without increased completeness (i.e., the baseline prediction accuracy is low). Therefore, these items benefit considerably from increased completeness in terms of prediction accuracy. The findings above should encourage web portals and business owners to increase and maintain the completeness of item content data. In addition, the results of Hypotheses H2a/b can be used to balance the cost and benefit of data quality improvement measures, a topic discussed in recent literature (Heinrich et al. 2018a). For instance, only items (e.g., products offered by a web portal) with a higher profit margin can be extended with additional content in a selective manner, avoiding a potentially expensive large-scale extension of the whole data set. This opens up an effective option to manage the item content data in an affordable manner, which can be a crucial factor for web portals.

Hypotheses H3a/b can be also supported except in the case of Portal 2 regarding the measures Precision, Recall and the F1-measure. In all other cases of H3a/b, our evaluation yields significant coefficients b_3 ranging from 0.004 to 0.39. This means that the amount of additional features and their feature values and the amount of filled up feature values *per user* show moderator effects. The effect size indicated by Cohen's f^2 ranges from 0.0003 to 0.094. By a detailed consideration of the results for Hypotheses H3a/b, two interesting observations can be made. First, similarly to the discussions above, RMSE shows the largest effect sizes. Second, in the case of H3a (Portal 2) the p-values of the coefficient b_3 were above the significance level of 0.05 for the measures Precision, Recall and F1-measure. This may be reasoned by the lower additional item content (7 of 19 features) as well as the lower number of users (505 users) in this particular evaluation. Thus, according to the results of H3 users with a stronger increase in the amount of additional features and their feature values or in the amount of filled up feature values are suggested to have a significantly higher increase in prediction accuracy. This means that web portals – similar to the discussion above – can manage and increase the prediction accuracy for specific users (e.g., users with low versus high sales volumes) by extending the content of items, which have been rated by these users or which may be interesting and recommended for them in the future. In addition, another promising option would be to give providers as well as users, which mainly rate items with a lower number of available features, an incentive to provide additional data for these items. In return, the user community would benefit in this way from more appropriate item recommendations.

The results of Hypotheses H4a/b and H5a/b indicate that the *amount* and *diversity* of additional item content does *in general* not moderate the increase in prediction accuracy as intuition might suggest. Although Hypothesis H4a can be supported by our evaluation with relatively high moderator effects indicated by Cohen's f^2 ranging from 0.162 to 1.657 and with positive significant coefficients b_3 (ranging from 0.02 to 0.70), Hypothesis H4b cannot be supported. This means that portals aiming to extend item content data should primarily focus on (selected) additional features with a high amount of feature values, but filling up features with a high amount of additional feature values does not lead to a higher increase in prediction accuracy in general. At first sight, this result is counterintuitive, as one would have expected that more filled up feature values would lead to a higher increase in prediction accuracy. A reason why filling up individual features with a high amount of missing values does not result in a higher increase in prediction accuracy – indicated by p-values above 0.05 of the coefficients b_3 for all four evaluation measures – could be that the additional content was inferred by a deficient imputation method. However, this can be rebutted as a significant increase in prediction accuracy was achieved in H1b, which would be also caused by the inferred feature values and thus by the chosen imputation technique. Instead, it is necessary to consider the importance of features in this context. For example, the feature *Special Needs* with values such as *Dog Allowed* and *Good For Dancing* has more missing feature values (i.e., less available feature values) than the feature *Parking Information* with values such as *Bike Parking* and *Private Parking Lot*. Therefore, filling up missing values for *Special Needs* leads to a higher increase in completeness compared to *Parking Information*. However, as transportation (e.g., by bike, car or subway) is an important aspect for restaurant visitors in New York City, features such as *Parking Information* seem to be more important for the majority of users (and thus, may be better maintained by those users) than features such as *Special Needs*. In our evaluation, this importance is indicated by a higher increase in prediction accuracy when filling up feature values, for instance, for the feature *Parking Information* compared to filling up the feature *Special Needs*. This shows that the result of H4b may be caused by important features having potentially less missing data values in the baseline data set. The results regarding Hypothesis H4a can be reasoned in a similar way. Compared to all other hypotheses, effect sizes regarding Hypothesis H4a are the largest. Here, an analysis of the data of H4a (Portal 1) shows that adding features with a high amount of feature value assignments such as *Special Services* yield a high increase in prediction accuracy. This is reasonable, since the feature *Special Services* has the feature values *Cheap Eats*, *Delivery* and *Take Out* and therefore, *Special Services* seems to constitute an important feature for the user ratings for restaurants in general. This further indicates that important features for users are those features with a high amount of available feature value assignments with regard to Hypothesis H4a. Therefore, it is reasonable, that the effect sizes for the moderator in H4a are the largest. Overall, the results do not indicate that the amount of additional feature values by itself is a positive moderator, but a high amount of available feature value assignments in a data set may be an indicator for the importance of features and its impact on prediction accuracy (cf. H4a).

Hypotheses H5a/b cannot be supported as indicated by coefficients b_3 with negative sign or with p-values above the 0.05 level of significance. This means that a higher diversity of added or filled up features does not yield a higher increase in prediction accuracy. In general, adding a feature with exactly the same feature value assignments as an existing feature to the data set should not yield any increase in prediction accuracy, as stated by the literature (Mitra et al. 2002; Tabakhi and Moradi 2015). Hence, the increase in prediction accuracy caused by adding features to a data set is expected to decrease with the similarity of these additional features to the existing features. Therefore, we would have anticipated that adding and filling up features with a high diversity would enable the recommender system to differentiate items in more details, thus leading to more accurate recommendations to users. However, an analysis shows that even features with a high diversity can be of low importance to users and thus, result in a low increase in prediction accuracy. For example, the additional feature *Production Company* with feature values such as *Paramount Pictures* or *Twentieth Century Fox* brings high diversity to the baseline feature *Genre*, as indicated by a mean cosine distance of 0.96 between the features *Production Company* and *Genre*. Nevertheless, adding only this feature has low impact on the increase in prediction accuracy (e.g., the RMSE decreased only by 0.002). This seems reasonable, as production companies produce diverse movies with different actors, directors and of different genres and therefore, the feature *Production Company* is usually of low importance for the majority of users. This underlines that features exist which have diverse feature value assignments, but their importance is low for users. Contrary to works such as (Mitra et al. 2002; Tabakhi and Moradi 2015), which propose to sort out features with high similarity (i.e., low diversity), this shows that the diversity or similarity of features may only be a subordinate factor for the impact of completeness on the prediction accuracy. In total, the increase in completeness by the amount of additional feature values (H4) as well as by the diversity of added/filled up features (H5) does not constitute a positive moderator of the impact of completeness on prediction accuracy.

Based on these findings and the above discussion, the contribution of our work to the existing body of knowledge can be outlined. Blake and Mangiameli (2011), Feldman et al. (2018) and Woodall et al. (2015) proposed and substantiated that completeness – in the sense of the amount of available feature values – has a significant impact on evaluation criteria such as decision quality of specific considered decision support systems. Complementary to these works, our results show that not only a higher amount of available feature values, but also adding new features to the feature set can have a significant impact on evaluation criteria of decision support systems and in particular recommender systems. Furthermore, so far, the impact of data quality was validated for different evaluation criteria. The works of Bharati and Chaudhury (2004) and Ge (2009) supported the impact on the evaluation criteria decision-making satisfaction and decision quality. Blake and Mangiameli (2011), Feldman et al. (2018) and Woodall et al. (2015) demonstrated the impact on data mining outcome. Supplementing these findings, our work is the first to analyze the impact of data quality – in particular completeness – on the evaluation criterion prediction accuracy. Moreover, our results show that the impact of data quality can be significantly influenced by moderators. While our findings support the so far not examined statement that the impact on prediction accuracy is moderated by the increase in completeness per item and per user, they show that the amount of additional feature values is not a positive moderator in this regard. Moreover, our findings do not support the intuitive concept that the diversity of features is a positive moderator of the impact of completeness on prediction accuracy.

Following this discussion, notable implications can be concluded for applications in practice. Expanding the discussion above, it is crucial for business owners to provide a large(r) number of features for their businesses and to check whether additional important features are available. The resulting increase in completeness leads to more accurate recommendations of these businesses, which better fit the users' preferences. Similarly, the acquisition of additional data is highly advantageous for web portals. It allows improved recommendations and enhances the efficacy of the web portal. Moreover, our findings should encourage meta portals, which already make use of data from different web sources, to further collect additional features and feature values and, in this way, to provide high quality recommendations. Currently, many meta portals (such as *trivago.com*) mainly focus on the integration of user ratings and reviews from different sources and mostly ignore the impact of an extended item content data set. By recommending items based on data with increased completeness, meta portals can exploit a much higher potential of making high quality product recommendations for customers. In case of limitations in acquiring additional features or feature values, it is important to focus on important additional features, which may be indicated by a high amount of available feature values. In contrast, a high diversity of additional features is not required.

Conclusions, Limitations and Directions for Future Work

We investigate the impact of the data quality dimension completeness of item content data on prediction accuracy. Based on a theoretical model derived from literature, hypotheses are formulated and substantiated. These hypotheses focus on the impact of adding features and filling up missing feature values on the prediction accuracy of recommendations, which was assessed by the measures RMSE, Precision, Recall and the F1-measure. The hypotheses are evaluated on two real-world data sets, one from the domain of restaurants and another one from the domain of movies. Our results yield that rating predictions are significantly more accurate when more features and feature values are available. Moreover, this impact of completeness on the increase in prediction accuracy is moderated by the amount of additional features and their feature values or the amount of filled up feature values per items and per users. In contrast, this statement does not hold for features. While adding features with a high amount of feature values leads to a higher increase in prediction accuracy, filling up a high amount of feature values or adding features to the existing content with a high diversity does not lead to a higher increase in prediction accuracy. Here, our results suggest that the importance of features to users is an essential factor for the increase in prediction accuracy. Our findings are not only valuable from a scientific perspective but also in practice for business owners as well as for web portals and meta portals.

Our work also has some limitations, which could be starting points for future research. In this paper, we increased completeness by adding features from other web portals as well as by imputing missing feature values. Nevertheless, other approaches to increase completeness are possible. For example, a feature set could be extended with features based on user-generated item tags as proposed by Zhang et al. (2010). Similarly, feature values could be filled up by analyzing additional textual data using text mining to extract non-available feature values (Ghani et al. 2006). Another limitation are the costs of data preparation and computation caused by adding features and their feature values or by filling up missing feature values. In our evaluation settings, the necessary additional time and costs are reasonable: For example, the computation time of CBMF for training and evaluating the model for Hypothesis H1a (Portal 1) was raised from 285 seconds to 488 seconds for all users/items, for Hypothesis H1a (Portal 2) from 10 seconds to 24 seconds. However, these costs might indeed be relevant for applications with a vast amount of additional item content data. Furthermore, it would be highly interesting to test the impact of other data quality dimensions such as currency on recommendation quality. Additionally, in this paper we focus on different metrics for prediction accuracy as the most important quality measures for recommender systems (Shani and Gunawardana 2011). However, as the goals of a recommender system can be very diverse (e.g., introducing customers to the full product spectrum) further metrics can be of particular interest for other application scenarios (Jannach et al. 2016). Thus, further research on the impact of data quality assessed by other quality measures such as coverage, serendipity or scalability (Herlocker et al. 2004; Shani and Gunawardana 2011) would also be relevant. Finally, in the future, tests similar to ours could also be conducted using data sets from further domains, such as recommendations for music songs.

References

Abel, Fabian; Herder, Eelco; Houben, Geert-Jan; Henze, Nicola; Krause, Daniel (2013): Cross-system user modeling and personalization on the Social Web. In *User Modeling and User-Adapted Interaction* 23 (2-3), pp. 169–209. DOI: 10.1007/s11257-012-9131-2.

Adomavicius, G.; Tuzhilin, A. (2005): Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions. In *IEEE Trans. Knowl. Data Eng.* 17 (6), pp. 734–749. DOI: 10.1109/TKDE.2005.99.

Adomavicius, Gediminas; Zhang, Jingjing (2012): Impact of data characteristics on recommender systems performance. In *ACM Trans. Manage. Inf. Syst.* 3 (1), pp. 1–17. DOI: 10.1145/2151163.2151166.

Adomavicius, Gediminas; Zhang, Jingjing (2016): Classification, Ranking, and Top-K Stability of Recommendation Algorithms. In *INFORMS Journal on Computing* 28 (1), pp. 129–147. DOI: 10.1287/ijoc.2015.0662.

Aggarwal, Charu C. (2014): Data Classification: Chapman and Hall/CRC.

Aguinis, Herman; Beaty, James C.; Boik, Robert J.; Pierce, Charles A. (2005): Effect size and power in assessing moderating effects of categorical variables using multiple regression: a 30-year review. In *The Journal of applied psychology* 90 (1), pp. 94–107. DOI: 10.1037/0021-9010.90.1.94.

Amatriain, Xavier; Pujol, Josep M.; Tintarev, Nava; Oliver, Nuria (2009): Rate it again. In Lawrence Bergman, Alex Tuzhilin, Robin Burke, Alexander Felfernig, Lars Schmidt-Thieme (Eds.): Proceedings of the third ACM conference on Recommender systems. New York, New York, USA. ACM Special Interest Group on Computer-Human Interaction. New York, NY: ACM, pp. 173–180.

Ballou, Donald P.; Pazer, Harold L. (1985): Modeling data and process quality in multi-input, multi-output information systems. In *Management Science* 31 (2), pp. 150–162.

Basaran, Daniel; Ntoutsi, Eirini; Zimek, Arthur (2017): Redundancies in Data and their Effect on the Evaluation of Recommendation Systems: A Case Study on the Amazon Reviews Datasets. In Nitesh Chawla, Wei Wang (Eds.): Proceedings of the 2017 SIAM International Conference on Data Mining. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 390–398.

Batini, Carlo; Cappiello, Cinzia; Francalanci, Chiara; Maurino, Andrea (2009): Methodologies for data quality assessment and improvement. In *ACM Comput. Surv.* 41 (3), pp. 1–52. DOI: 10.1145/1541880.1541883.

Batini, Carlo; Scannapieco, Monica (2016): Data and Information Quality. Cham: Springer International Publishing.

Bell, Robert M.; Koren, Yehuda; Volinsky, Chris (2007): The BellKor solution to the Netflix prize.

Berkovsky, Shlomo; Kuflik, Tsvi; Ricci, Francesco (2012): The impact of data obfuscation on the accuracy of collaborative filtering. In *Expert Systems with Applications* 39 (5), pp. 5033–5042. DOI: 10.1016/j.eswa.2011.11.037.

Bharati, Pratyush; Chaudhury, Abhijit (2004): An empirical investigation of decision-making satisfaction in web-based decision support systems. In *Decision Support Systems* 37 (2), pp. 187–197. DOI: 10.1016/S0167-9236(03)00006-X.

Blake, Roger; Mangiameli, Paul (2011): The Effects and Interactions of Data Quality and Problem Complexity on Classification. In *J. Data and Information Quality* 2 (2), pp. 1–28. DOI: 10.1145/1891879.1891881.

Bobadilla, Jesús; Ortega, Fernando; Hernando, Antonio; Gutiérrez, Abraham (2013): Recommender systems survey. In *Knowledge-Based Systems* 46, pp. 109–132.

Boneau, C. Alan (1960): The effects of violations of assumptions underlying the t test. In *Psychological Bulletin* 57 (1), pp. 49–64. DOI: 10.1037/h0041412.

Bostandjiev, Svetlin; O'Donovan, John; Höllerer, Tobias (2012): TasteWeights: a visual interactive hybrid recommender system. In Pádraig Cunningham, Neil Hurley, Ido Guy, Sarabjot Singh Anand (Eds.): Proceedings of the sixth ACM conference on Recommender systems. Dublin, Ireland. ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human Interaction; ACM Special Interest Group on Hypertext, Hypermedia, and Web; ACM Special Interest Group on Information Retrieval. New York, NY: ACM, pp. 35–42.

Burke, Robin; Ramezani, Maryam (2011): Matching Recommendation Technologies and Domains. In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 367–386.

Christen, Peter (2012): Data matching. Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Berlin, Heidelberg: Springer Berlin Heidelberg.

Cohen, Jacob (1988): Statistical Power Analysis for the Behavioral Sciences. 2nd ed. Hillsdale, NJ: Erlbaum. Available online at http://gbv.eblib.com/patron/FullRecord.aspx?p=1192162.

Cohen, Jacob; Cohen, Patricia; West, Stephen G.; Aiken, Leona S. (2003): Applied multiple regression/correlation analysis for the behavioral sciences. third edition. New York, London, Mahwah, NJ: Routledge Taylor & Francis Group. Available online at http://www.loc.gov/catdir/enhancements/fy0634/2002072068-d.html.

Cunha, Tiago; Soares, Carlos; Carvalho, André C. P. L. F. de (2016): Selecting Collaborative Filtering Algorithms Using Metalearning. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, Jilles Vreeken (Eds.): Machine Learning and Knowledge Discovery in Databases. European Conference, Ecml Pkdd 2016, Riva

Del Garda, Italy, September 19-23, 2016, Proceedings, vol. 9852. Cham: Springer-Verlag New York Inc (LNCS Sublibrary: SL7 - Artificial Intelligence, 9851-9853), pp. 393–409.

Dawson, Jeremy F. (2014): Moderation in Management Research: What, Why, When, and How. In *J Bus Psychol* 29 (1), pp. 1–19. DOI: 10.1007/s10869-013-9308-7.

Doerfel, Stephan; Jäschke, Robert; Stumme, Gerd (2016): The Role of Cores in Recommender Benchmarking for Social Bookmarking Systems. In *ACM Trans. Intell. Syst. Technol.* 7 (3), pp. 1–33. DOI: 10.1145/2700485.

Ekstrand, Michael; Riedl, John (2012): When recommenders fail. In Pádraig Cunningham (Ed.): Proceedings of the sixth ACM conference on Recommender systems. the sixth ACM conference. Dublin, Ireland, 9/9/2012 - 9/13/2012. New York, NY: ACM (ACM Digital Library), p. 233.

Enders, Craig K. (2010): Applied missing data analysis. New York: Guilford Press (Methodology in the social sciences). Available online at http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10389908.

Feldman, Michael; Even, Adir; Parmet, Yisrael (2018): A methodology for quantifying the effect of missing data on decision quality in classification problems. In *Communications in Statistics–Theory and Methods* 47 (11), pp. 2643–2663.

Felfernig, Alexander; Friedrich, Gerhard; Schmidt-Thieme, Lars (2007): Recommender systems. In *IEEE Intelligent Systems* 22 (3).

Forbes, Peter; Zhu, Mu (2011): Content-boosted matrix factorization for recommender systems. In Bamshad Mobasher, Robin Burke, Dietmar Jannach, Gediminas Adomavicius (Eds.): Proceedings of the fifth ACM conference on Recommender systems. Proceedings of the fifth ACM conference on Recommender systems. Chicago, Illinois, USA. New York, NY: ACM, pp. 261–264.

Fortes, Reinaldo Silva; Freitas, Alan R. R. de; Gonçalves, Marcos André (2017): A Multicriteria Evaluation of Hybrid Recommender Systems: On the Usefulness of Input Data Characteristics.

Ge, Mouzhi (2009): Information quality assessment and effects on inventory decision-making. Doctoral dissertation. Dublin City University, Dublin City University.

Geuens, Stijn; Coussement, Kristof; Bock, Koen W. de (2018): A framework for configuring collaborative filtering-based recommendations derived from purchase data. In *European Journal of Operational Research* 265 (1), pp. 208–218. DOI: 10.1016/j.ejor.2017.07.005.

Ghani, Rayid; Probst, Katharina; Liu, Yan; Krema, Marko; Fano, Andrew (2006): Text mining for product attribute extraction. In *ACM SIGKDD Explorations Newsletter* 8 (1), pp. 41–48. DOI: 10.1145/1147234.1147241.

Gignac, Gilles E.; Szodorai, Eva T. (2016): Effect size guidelines for individual differences researchers. In *Personality and Individual Differences* 102, pp. 74–78. DOI: 10.1016/j.paid.2016.06.069.

Gomez-Uribe, Carlos A.; Hunt, Neil (2016): The Netflix recommender system: Algorithms, business value, and innovation. In *ACM Transactions on Management Information Systems (TMIS)* 6 (4, Article 13).

Grčar, Miha; Mladenič, Dunja; Fortuna, Blaž; Grobelnik, Marko (2006): Data Sparsity Issues in the Collaborative Filtering Framework. In Olfa Nasraoui (Ed.): Advances in web mining and web usage analysis. 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005 : Chicago, IL, USA, August 21, 2005 : revised papers, vol. 4198. Berlin: Springer (Lecture Notes in Computer Science, 4198), pp. 58–76.

Griffith, Josephine; O'Riordan, Colm; Sorensen, Humphrey (2012): Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In Sascha Ossowski, Paola Lecca (Eds.): Proceedings of the 27th annual ACM symposium on applied computing 2012. Symposium on Applied Computing : Riva del Garda, Trento, Italy, March 26-30, 2012. the 27th Annual ACM Symposium. Trento, Italy, 3/26/2012 - 3/30/2012. New York, N.Y.: ACM Press; Association for Computing Machinery, p. 937.

Gunawardana, Asela; Shani, Guy (2015): Evaluating Recommender Systems. In Francesco Ricci, Lior Rokach, Bracha Shapira (Eds.): Recommender Systems Handbook, vol. 12. Boston, MA: Springer US, pp. 265–308.

Harper, F. Maxwell; Konstan, Joseph A. (2015): The MovieLens Datasets. In *ACM Trans. Interact. Intell. Syst.* 5 (4), pp. 1–19. DOI: 10.1145/2827872.

Hayes, Andrew F. (2013): Introduction to mediation, moderation, and conditional process analysis. A regression-based approach. New York, NY: Guilford Press (Methodology in the social sciences). Available online at http://lib.myilibrary.com/detail.asp?id=480011.

Heinrich, Bernd; Hristova, Diana (2016): A quantitative approach for modelling the influence of currency of information on decision-making under uncertainty. In *Journal of Decision Systems* 25 (1), pp. 16–41. DOI: 10.1080/12460125.2015.1080494.

Heinrich, Bernd; Hristova, Diana; Klier, Mathias; Schiller, Alexander; Szubartowicz, Michael (2018a): Requirements for Data Quality Metrics. In *J. Data and Information Quality* 9 (2), pp. 1–32. DOI: 10.1145/3148238.

Heinrich, Bernd; Klier, Mathias; Schiller, Alexander; Wagner, Gerit (2018b): Assessing data quality – A probability-based metric for semantic consistency. In *Decision Support Systems* 110, pp. 95–106. DOI: 10.1016/j.dss.2018.03.011.

Helm, Roland; Mark, Antje (2012): Analysis and evaluation of moderator effects in regression models: state of art, alternatives and empirical example. In *Rev Manag Sci* 6 (4), pp. 307–332. DOI: 10.1007/s11846-010-0057-y.

Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.; Riedl, John T. (2004): Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems (TOIS)* 22 (1), pp. 5–53.

Huang, Zan; Zeng, Daniel D. (2005): Why Does Collaborative Filtering Work? Recommendation Model Validation and Selection By Analyzing Bipartite Random Graphs. In *SSRN Journal*. DOI: 10.2139/ssrn.894029.

Jannach, Dietmar; Resnick, Paul; Tuzhilin, Alexander; Zanker, Markus (2016): Recommender Systems - Beyond Matrix Completion. In *Commun. ACM* 59 (11), pp. 94–102. DOI: 10.1145/2891406.

Karatzoglou, Alexandros; Hidasi, Balázs (2017): Deep Learning for Recommender Systems. In Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, Alexander Tuzhilin (Eds.): Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17. the Eleventh ACM Conference. Como, Italy, 27.08.2017 - 31.08.2017. New York, New York, USA: ACM Press, pp. 396–397.

Kayaalp, Mehmet; Özyer, Tansel; Özyer, Sibel Tariyan (2009): A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site. In Nasrullah Memon (Ed.): International Conference on Advances in Social Networks Analysis and Mining, 2009. Piscataway, NJ: IEEE, pp. 113–118.

Kim, Donghyun; Park, Chanyoung; Oh, Jinoh; Lee, Sungyoung; Yu, Hwanjo (2016): Convolutional Matrix Factorization for Document Context-Aware Recommendation. In Shilad Sen, Werner Geyer, Jill Freyne, Pablo Castells (Eds.): Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16. the 10th ACM Conference. Boston, Massachusetts, USA, 15.09.2016 - 19.09.2016. New York, New York, USA: ACM Press, pp. 233–240.

Konstan, Joseph A.; Riedl, John (2012): Recommender systems. From algorithms to user experience. In *User Model User-Adap Inter* 22 (1-2), pp. 101–123. DOI: 10.1007/s11257-011-9112-x.

Koren, Yehuda (2009): The bellkor solution to the netflix grand prize. In *Netflix prize documentation* 81, pp. 1–10.

Koren, Yehuda; Bell, Robert; Volinsky, Chris (2009): Matrix Factorization Techniques for Recommender Systems. In *Computer* 42 (8), pp. 30–37. DOI: 10.1109/MC.2009.263.

Lathia, Neal; Amatriain, Xavier; Pujol, Josep M. (2009): Collaborative filtering with adaptive information sources. In Sarabjot Singh Anand, Bamshad Mobasher, Alfred Kobsa, Dietmar Jannach (Eds.): Proceedings of the 7th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems (ITWP'09). Intelligent Techniques for Web Personalization & Recommender Systems -- ITWP'09. Pasadena, California, USA, July 11-17. CEUR-WS. org (CEUR Workshop Proceedings (CEUR-WS.org), 528), pp. 81–86.

Lee, Yang W.; Strong, Diane M.; Kahn, Beverly K.; Wang, Richard Y. (2002): AIMQ: a methodology for information quality assessment. In *Information & Management* 40 (2), pp. 133–146. DOI: 10.1016/S0378-7206(02)00043-5.

Levi, Asher; Mokryn, Osnat; Diot, Christophe; Taft, Nina (2012): Finding a needle in a haystack of reviews. cold start context-based hotel recommender system. In Pádraig Cunningham, Neil Hurley, Ido Guy, Sarabjot Singh Anand (Eds.): Proceedings of the sixth ACM conference on Recommender systems. Dublin, Ireland. ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human Interaction; ACM Special Interest Group on Hypertext, Hypermedia, and Web; ACM Special Interest Group on Information Retrieval. New York, NY: ACM, pp. 115–122.

Levy, Yair; Ellis, Timothy J. (2006): A systems approach to conduct an effective literature review in support of information systems research. In *Informing Science* 9, pp. 181–212.

Li, Seth Siyuan; Karahanna, Elena (2015): Online recommendation systems in a B2C E-commerce context: a review and future directions. In *Journal of the Association for Information Systems* 16 (2), pp. 72–107.

Lops, Pasquale; Gemmis, Marco de; Semeraro, Giovanni (2011): Content-based recommender systems. State of the art and trends. In : Recommender systems handbook: Springer, pp. 73–105.

Lu, Jie; Wu, Dianshuang; Mao, Mingsong; Wang, Wei; Zhang, Guangquan (2015): Recommender system application developments: A survey. In *Decision Support Systems* 74, pp. 12–32. DOI: 10.1016/j.dss.2015.03.008.

MacCallum, Robert C.; Mar, Corinne M. (1995): Distinguishing between moderator and quadratic effects in multiple regression. In *Psychological Bulletin* 118 (3), pp. 405–421. DOI: 10.1037/0033-2909.118.3.405.

Matuszyk, Pawel; Spiliopoulou, Myra (2014): Predicting the Performance of Collaborative Filtering Algorithms. In Rajendra Akerkar, Nick Bassiliades, John Davies, Vadim Ermolayev (Eds.): WIMS '14 : 4th International Conference on Web Intelligence, Mining and Semantics. the 4th International Conference. Thessaloniki, Greece, 6/2/2014 - 6/4/2014. New York, New York, USA: ACM Press, pp. 1–6.

Mitra, P.; Murthy, C. A.; Pal, S. K. (2002): Unsupervised feature selection using feature similarity. In *IEEE Trans. Pattern Anal. Machine Intell.* 24 (3), pp. 301–312. DOI: 10.1109/34.990133.

Nguyen, Jennifer; Zhu, Mu (2013): Content-boosted matrix factorization techniques for recommender systems. In *Statistical Analy Data Mining* 6 (4), pp. 286–301. DOI: 10.1002/sam.11184.

Ning, Xia; Desrosiers, Christian; Karypis, George (2015): A comprehensive survey of neighborhood-based recommendation methods. In : Recommender systems handbook: Springer, pp. 37–76.

Ning, Yue; Shi, Yue; Hong, Liangjie; Rangwala, Huzefa; Ramakrishnan, Naren (2017): A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation. In Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, Alexander Tuzhilin (Eds.): Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17. the Eleventh ACM Conference. Como, Italy, 27.08.2017 - 31.08.2017. New York, New York, USA: ACM Press, pp. 23–31.

Olteanu, Alexandra; Kermarrec, Anne-Marie; Aberer, Karl (2014): Comparing the Predictive Capability of Social and Interest Affinity for Recommendations. In Boualem Benatallah, Azer Bestavros, Yannis Manolopoulos, Athena Vakali, Yanchun Zhang (Eds.): Web information systems engineering - WISE 2014. 15th International Conference, Thessaloniki, Greece, October 12-14, 2014 : proceedings, vol. 8786. Cham: Springer (LNCS sublibrary. SL 3, Information systems and application, incl. Internet/Web and HCI, 8786-8787), pp. 276–292.

Ozsoy, Makbule Gulcin; Polat, Faruk; Alhajj, Reda (2015): Modeling Individuals and Making Recommendations Using Multiple Social Networks. In Jian Pei, Fabrizio Silvestri, Jie Tang (Eds.): Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Piscataway, NJ, New York, NY: IEEE; ACM, pp. 1184–1191.

Pazzani, Michael J.; Billsus, Daniel (2007): Content-based recommendation systems. In : The adaptive web: Springer, pp. 325–341.

Pessemier, Toon de; Dooms, Simon; Deryckere, Tom; Martens, Luc (2010): Time dependency of data quality for collaborative filtering algorithms. In Xavier Amatriain, Marc Torrens, Paul Resnick, Markus Zanker (Eds.): Proceedings of the fourth ACM conference on Recommender systems. Barcelona, Spain. ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human Interaction; ACM Special Interest Group on Information Retrieval; ACM Special Interest Group on Hypertext, Hypermedia, and Web. New York, NY: ACM, pp. 281–284.

Picault, Jérome; Ribiere, Myriam; Bonnefoy, David; Mercer, Kevin (2011): How to get the Recommender out of the Lab? In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 333–365.

Pipino, Leo L.; Lee, Yang W.; Wang, Richard Y. (2002): Data quality assessment. In *Commun. ACM* 45 (4), pp. 211–218. DOI: 10.1145/505248.506010.

Porcel, Carlos; Herrera-Viedma, Enrique (2010): Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. In *Knowledge-Based Systems* 23 (1), pp. 32–39.

Power, Daniel J.; Sharda, Ramesh; Burstein, Frada (2015): Decision support systems: John Wiley & Sons, Ltd.

Redman, Thomas C. (1996): Data quality for the information age. Boston, MA: Artech House (The Artech House computer science library).

Ricci, Francesco; Rokach, Lior; Shapira, Bracha (2015): Recommender Systems: Introduction and Challenges. In Francesco Ricci, Lior Rokach, Bracha Shapira (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 1–34.

Ricci, Francesco; Rokach, Lior; Shapira, Bracha; Kantor, Paul B. (Eds.) (2011): Recommender Systems Handbook. Boston, MA: Springer US.

Sar Shalom, Oren; Berkovsky, Shlomo; Ronen, Royi; Ziklik, Elad; Amihood, Amir (2015): Data Quality Matters in Recommender Systems. In Hannes Werthner, Markus Zanker, Jennifer Golbeck, Giovanni Semeraro (Eds.): Proceedings of the 9th ACM Conference on Recommender Systems. Vienna, Austria. RecSys; Association for Computing Machinery; ACM Conference on Recommender Systems; ACM Recommender Systems Conference. New York, NY: ACM, pp. 257–260.

Sarwar, Badrul M.; Karypis, George; Konstan, Joseph; Riedl, John (2002): Recommender systems for large-scale e-commerce. Scalable neighborhood formation using clustering. In : Proceedings of the fifth international conference on computer and information technology, vol. 1, pp. 291–324.

Schwarz, Gideon (1978): Estimating the Dimension of a Model. In *Ann. Statist.* 6 (2), pp. 461–464. DOI: 10.1214/aos/1176344136.

Sergis, Stylianos; Sampson, Demetrios G. (2016): Learning Object Recommendations for Teachers Based On Elicited ICT Competence Profiles. In *IEEE Trans. Learning Technol.* 9 (1), pp. 67–80. DOI: 10.1109/TLT.2015.2434824.

Shani, Guy; Gunawardana, Asela (2011): Evaluating recommendation systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 257–297.

Shmueli; Koppius (2011): Predictive Analytics in Information Systems Research. In *MIS Quarterly* 35 (3), pp. 553–572. DOI: 10.2307/23042796.

Song, Yading; Dixon, Simon; Pearce, Marcus (2013): A survey of music recommendation systems and future perspectives. In Mitsuko Aramaki, Mathieu Barthet, Richard Kronland-Martinet, Sølvi Ystad (Eds.): From sounds to music and emotions. CMMR; International Symposium on Computer Music Modeling and Retrieval; CMMR "Music & Emotions". Berlin: Springer (Lecture Notes in Computer Science, 7900).

Symeonidis, Panagiotis (2016): Matrix and Tensor Decomposition in Recommender Systems. In Shilad Sen, Werner Geyer, Jill Freyne, Pablo Castells (Eds.): Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16. the 10th ACM Conference. Boston, Massachusetts, USA, 15.09.2016 - 19.09.2016. New York, New York, USA: ACM Press, pp. 429–430.

Tabakhi, Sina; Moradi, Parham (2015): Relevance–redundancy feature selection based on ant colony optimization. In *Pattern Recognition* 48 (9), pp. 2798–2811. DOI: 10.1016/j.patcog.2015.03.020.

Vargas-Govea, Blanca; González-Serna, Gabriel; Ponce-Medellın, Rafael (2011): Effects of relevant contextual features in the performance of a restaurant recommender system. In Bamshad Mobasher, Robin Burke, Dietmar Jannach, Gediminas Adomavicius (Eds.): Proceedings of the fifth ACM conference on Recommender systems. Proceedings of the fifth ACM conference on Recommender systems. Chicago, Illinois, USA. New York, NY: ACM, pp. 592–596.

Wand, Yair; Wang, Richard Y. (1996): Anchoring data quality dimensions in ontological foundations. In *Commun. ACM* 39 (11), pp. 86–95. DOI: 10.1145/240455.240479.

Wang, Richard Y.; Storey, Veda C.; Firth, Christopher P. (1995): A framework for analysis of data quality research. In *IEEE transactions on knowledge and data engineering* 7 (4), pp. 623–640.

Woodall, Philip; Borek, Alexander; Gao, Jing; Oberhofer, Martin; Koronios, Andy (2015): An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics. In Richard Wang (Ed.): Big data. Management and data quality ; 19th International Conference on Information Quality (ICIQ 2014), Xi'an, China, 1 - 3 August 2014. International Conference on Information Quality; ICIQ. Red Hook, NY: Curran, pp. 24–33.

Zapata, Alfredo; Menéndez, Víctor H.; Prieto, Manuel E.; Romero, Cristóbal (2015): Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. In *International Journal of Human-Computer Studies* 76, pp. 22–39. DOI: 10.1016/j.ijhcs.2014.12.002.

Zhang, Zi-Ke; Zhou, Tao; Zhang, Yi-Cheng (2010): Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. In *Physica A: Statistical Mechanics and its Applications* 389 (1), pp. 179–186. DOI: 10.1016/j.physa.2009.08.036.

3 Disruptive Events in Service Systems

3.1 Paper 3: User-based Event Handling Strategy for Multi User Context-Aware Service Systems in Mobile Environments

Current Status	Full Citation
This paper **is under review** at the 19th Internationale Tagung Wirtschaftsinformatik.	Lohninger, Daniel (2024): User-based Event Handling Strategy for Multi User Context-Aware Service Systems in Mobile Environments. Working Paper, University of Regensburg.

User-based Event Handling Strategy for Multi User Context-Aware Service Systems in Mobile Environments

Daniel Lohninger

University of Regensburg, Information Systems, Regensburg, Germany
Daniel.Lohninger@informatik.uni-regensburg.de

Abstract. Modern service systems enable the execution of individual multi user context-aware processes in domains such as tourism or healthcare. The dynamic nature of mobile environments leads to a variety of disruptive events at runtime, requiring re-selection of the initially planned service composition for multiple users. In this respect, re-selection for each event is not practical due to the high complexity of the underlying service selection problem. Consequently, this paper presents an event handling strategy for service systems based on *user preferences* for dependencies in regard to *context information* and *multiple users*. In particular, we surveyed 201 participants as part of an empirical study and derived innovative user-based rules and mechanisms in order to prevent disruptive events and to reduce the number of re-selections due to systematic processing of events.

1 Introduction

Processes that use context information (e.g., location information) to provide personalized value to multiple users are becoming increasingly important (Allied Market Research 2021). These processes can be found in several domains such as tourism, disaster relief or healthcare (Gavalas et al. 2014; Kartiwi and Gunawan 2013; Ventola 2014). Consequently, this development also supports the *engineering* of multi user context-aware *service systems* (Alter 2012). Hence, a service composition for a process at planning time is determined for each user by selecting from a large number of available candidate services for each task of the process (Bortlik et al. 2018). Accordingly, mobile environments are an important driver of *service system innovation* (Demirkan et al. 2015) and are steadily gaining in importance in the form of mobile applications and mobile business (Statista 2019a; Statista 2019b). For example, a mobile application for multi user context-aware service systems from the tourism domain called *Culture Trip* supports processes such as city day trips. Specifically, guided joint tours, the creation of individual plans or the search for interesting spots are offered.

However, due to the dynamic nature of service systems in particular in mobile environments (Sheng et al. 2014), a high number of *disruptive events* may occur at runtime (Bobek and Nalepa 2017; Wang et al. 2017). This means, for example, the quality of selected candidate services or context information changes (cf., e.g., Canfora et al. 2008; Sbai et al. 2020; Sheng et al. 2014; Zheng et al. 2014). In this regard, we define a disruptive event as a *change* of *planned values* of candidate services at runtime (Bearzotti et al. 2012). This can be illustrated by an example from the tourism domain for context changes: The city day trip would change for one or more users if the weather changed from the planned value "sunny" to "rainy". Thus, these changes lead to a re-selection and thus to an adaptation of the service composition. Due to the NP-hardness of the underlying service selection problem (Abu-Khzam et al. 2015; Alrifai et al. 2012), re-selecting a service composition on the occurrence of each disruptive event at runtime is not practical. Thus, this paper focuses on the following research question:

How to develop an event handling strategy for multi user context-aware service systems based on user preferences?

According to literature, customer participation is crucial for service systems engineering and hence positively influences the performance of mobile applications (Böhmann et al. 2014; Demirkan et al. 2015; Ye and Kankanhalli 2020). In this regard and to address our research question, we surveyed 201 participants as part of an empirical study and present a user-based event handling strategy. Thus, we are the first *event handling strategy* for multi user context-aware service systems that explicitly derives rules and mechanisms based on *user preferences*, especially for *context information* and *multiple user*s, to systematically prevent and process disruptive events.

The remainder of the paper is structured as follows. In the next section, the theoretical background, related work and research gap as well as a real-world scenario from the tourism domain are discussed. In the third section, we present the findings of the empirical study conducted. In the fourth section, we propose a user-based event handling strategy for multi user context-aware service systems. The final section summarizes the work and discusses practical implications as well as limitations.

2 Background

2.1 Theoretical Background

Research in the field of service systems engineering can be divided into the two main topics: *Engineering service architecture* and *Engineering service systems interaction* (Böhmann et al. 2014). A structure of the topics and the mapping to the application context of this paper (cf. Figure 1) can be derived from the service science literature (cf., e.g., Alter 2012; Böhmann et al. 2014; Cristea et al. 2011; Melo and Aquino 2021; Michelson 2006). The *first topic*, which describes advanced models, methods and tools for service architecture engineering can already be found in the literature for multi user context-aware service systems (i.e., e.g., service selection algorithms; cf., e.g., Bortlik et al. 2018). The *second topic* mainly describes information intensive interactions with(in) service systems mainly focusing on the recovery after a service failure. In this regard, disruptive events are processed by event handling strategies and, if necessary, an alternative service composition is selected at runtime by service re-selection algorithms (cf. Bortlik et al. 2023). In the following, we will focus on event handling strategies as these have not yet been examined in more detail in the literature for multi user context-aware service systems (cf. Section 2.2). One further perspective for service systems engineering is the *environment* of a service system, which directly or indirectly influences the architecture or interactions of a service system (Alter 2012) and are primarily driven by including *context information* (Chandler and Vargo 2011). Furthermore, the increasing demand for *personalization* (Demirkan et al. 2015) and *customer participation* (Ye and Kankanhalli 2020) also influences service systems engineering.

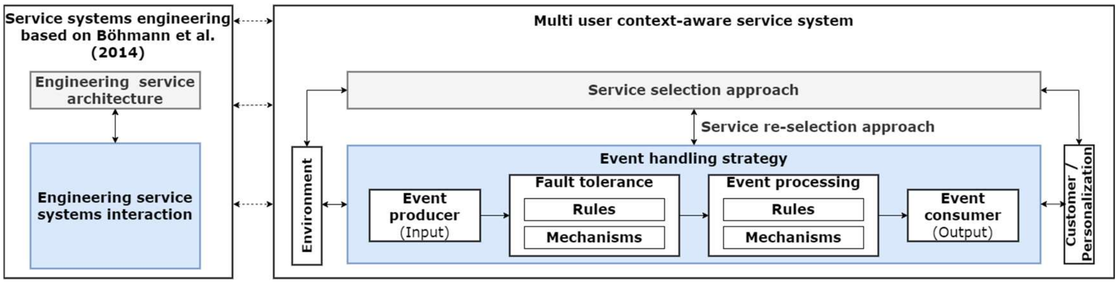

Figure 4. Components of service systems engineering

Event handling strategies are a well-known topic in the literature especially for service systems interaction (Cristea et al. 2011; Melo and Aquino 2021; Michelson 2006) to cope with the environment, personalization and customer participation. These strategies enable the definition of *rules* and *mechanisms* in order to respond adequately to disruptive events at runtime (Michelson 2006). Figure 1 illustrates the four components of event handling strategies in the service systems literature: *Event producer*; *Fault tolerance*; *Event processing*; *Event consumer*. The component *event producer* is the trigger of an event. In mobile environments, for example, triggers may be changes in the contextual environment (e.g., mobile location sensors (GPS), cf. Cristea et al. 2011). *Fault tolerance* enables the regular operation of a service system, even in the presence of disruptions, errors or failures, and therefore reduces disruptive events with support of rules and mechanisms (Melo and Aquino 2021). Furthermore, with *event processing*, disruptive events are collected as well as evaluated against rules and consequently mechanisms are initiated (Michelson 2006). Finally, the component *event consumer* is the application or business process of the service system, which receives the events from the event processing and reacts accordingly to these events (Cristea et al. 2011).

2.2 Related Work and Research Gap

In the following, we discuss existing event handling strategies for service systems from the literature. The discussion is based on a literature search of related work in ACM, aisnet.org, IEEE Xplore, INFORMS, ScienceDirect, Springer and Web of Science, which was performed using 30 keywords (i.e., e.g., *event handling*, *service system*, *fault tolerance*, *event processing*, *multiple user*, *context information*) resulting in 102 papers. Furthermore, we also conducted a backward and forward search based on these papers, resulting in 25 further articles. After a more detailed text analysis of the relevant articles and only considering papers, which includes *fault tolerance* and/or *event processing*, dependencies between *multiple users* and/or *context information* as well as *user preferences*, eleven relevant approaches remain that reflect our research topic in general. A short summary of the systematized related work can be found in Table 1.

Table 1. Overview of existing event handling strategies for service systems

Event handling strategy for service systems	**Dependencies**		**User prefer-ences**	**Event handling**	
	Con-text	Multiple user		Fault toler-ance	Event pro-cessing
Angarita et al. (2013), Angarita et al. (2014), Buys et al. (2011), Fekih et al. (2019), Shen and Yang (2011), Zheng and Lyu (2009)	**X**	-/-	-/-	**X**	-/-
Adams et al. (2007), Ayed et al. (2013), Flouris et al. (2017), Kum (2020), Wang et al. (2017)	**X**	-/-	-/-	-/-	**X**

The first group of event handling strategies mainly deal with *fault tolerance*. Therefore, existing literature suggests dynamic fault tolerance and recovery strategies with the goal to prevent a failure of a web service including the execution environment (Angarita et al. 2013; Angarita et al. 2014; Fekih et al. 2019). Furthermore, Zheng and Lyu (2009) tries to pass on experiences from web service users to future web service users with its fault tolerance strategy based on context information. In addition, Buys et al. (2011) develops metrics to better predict the likelihood of web service failures in the face of contextual changes. Finally, Shen and Yang (2011) describes an approach to prevent a failed negotiation process in the execution environment of a composite service.

The second group of service system approaches contributes to *event processing*.
There are several works, which use event processing to handle events in service systems (Ayed et al. 2013; Flouris et al. 2017; Kum 2020). Therefore, rules and event patterns are created for the processing of events in order to control faulty services. In this regard, context information is described as environmental context, in which, for example, the abstraction hierarchy of events is mapped (e.g., Kum 2020). In further research, dynamic self-adaptation approaches enable a self-contained and automatic repair of the exception situation in service systems on the basis of event patterns, in which context is defined as the valid conditions at runtime (Adams et al. 2007; Wang et al. 2017).

To sum up, all discussed approaches contribute to context-awareness, but do not deal with multiple users. Thus, we are the first event handling strategy for service systems (i.e., engineering service systems interaction, cf. Section 2.1), which directly derives rules and mechanisms based on *user preferences* for dependencies in regard to *context information* and *multiple users*. Moreover, a multi user context-aware service system that copes with all main event handling components (i.e., fault tolerance and event processing) is – to the best of our knowledge – missing so far.

2.3 Real-world Scenario

In the following, we introduce a scenario of a mobile-enabled multi user context-aware process from the tourism domain (i.e., a group is planning a city day trip, cf. Figure 2).

Therefore, our process consists of a number of temporal ordered tasks (e.g., visiting restaurant) and each task is connected by a transport (e.g., car). Moreover, a single task is conducted simultaneously by multiple users (i.e., user 1, user 2 and user 3) based on the user's specifications (e.g., favorite cuisine). Each task consists of a set of functionally equivalent candidate services (i.e., a special restaurant, e.g., *Da Guido 365*), that only differ in their non-functional properties (e.g., price). Thus, a single candidate service is described by attributes such as *name*, *duration*, *opening hours*, *price, location* or *rating*. Furthermore, world states are used to deal with context information (Ghallab et al. 2008). Therefore, context information such as *time* or *location* for each task of an individual user is defined by world states (e.g., user 1 is able to visit a restaurant at 12 p. m. based on the actual location of the user, i.e., world state ws_{20}). The aim of the service system in our scenario is to determine the optimal service composition (i.e., maximizing the utility) for all users in the process, considering specifications and constraints (e.g., maximum budget) per user (cf. Heinrich and Mayer 2018).

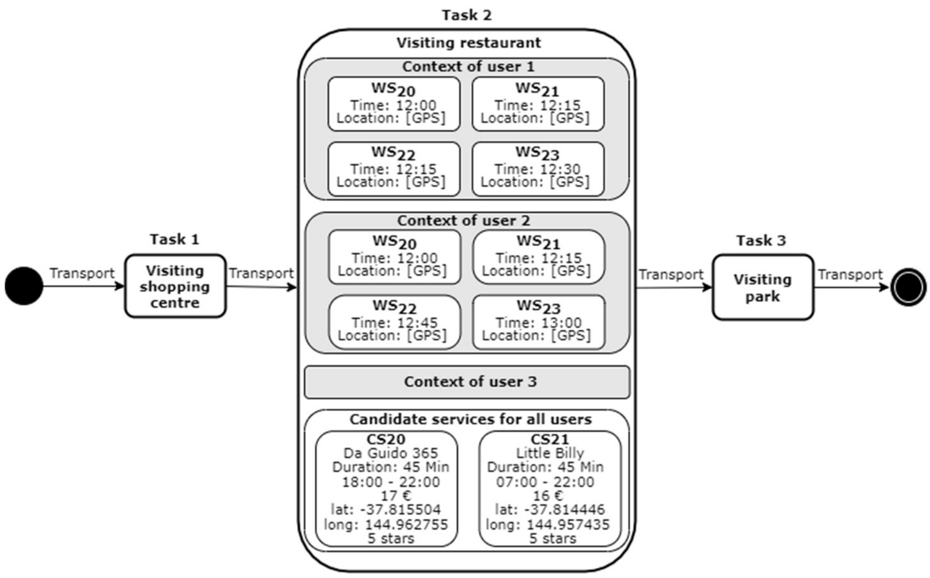

Figure 2. Multi user context-aware process in the tourism domain

At runtime several disruptive events can occur in multi user context-aware processes. An event consequently leads to planned values of candidate services (e.g., for context information or quality of service) changing at runtime and thus a service composition is no longer optimal (i.e., constraints not exceeded) or no longer valid (i.e., constraints exceeded). These changes potentially leading to a re-selection and therefore to adaptations of the originally planned service composition. In this work, we assume that events can occur at runtime with regard to the following types: Changes in *quality of service* (e.g., increase in prices of candidate services, worsening of the assumed rating of candidate services) or *context information* (e.g., negative change in the weather, delays in the execution of process tasks or the transport, extensions in the distance e.g. from the candidate service to the parking lot) because these are main issues in service systems in which planned values can change (Mo et al. 2011; Zeng et al. 2007).

3 Survey for Event Handling in Multi User Context-Aware Processes in Mobile Environments

Goals & questionnaire: In order to develop an event handling strategy for multi user context-aware service systems based on user preferences, we conducted an online survey. In particular, the goal of our survey was to derive user-based rules and mechanisms for an event handling approach with focus on *fault tolerance* and *event processing*, as well as *multiple users* and *context-awareness*. In this regard, the main structure of the standardized questionnaire is based on Atteslander and Cromm (2010) as well as Schnell (2019) and was created and made available online for *students* over a period of four weeks. In particular, mainly closed questions were used but the respondent also had the option of textual annotations. In a total, 35 questions in the following seven thematic blocks were surveyed in order to get the preferences of the participants: 1) *General* 2) *Events and solutions* 3) *Event types* 4) *Constraints* 5) *Context information* 6) *Multiple users* 7) *Demographic questions*. The blocks and also the individual questions were defined and cross-validated in advance through qualitative interviews with two students. All questions are primarily aimed at whether and how different types of events in the tourism domain should be prevented or processed. One question with regard to multiple users for illustration was: "*Assume your group in the city day trip has ten members. At what proportion of users affected by an event should an event recovering for all users take place?*". Overall, the tourism domain was chosen for the survey as this is a valid and comprehensible scenario in service systems engineering and mobile environments (Femenia-Serra et al. 2019).

Participants: The survey was made available to a large proportion of business informatics students at the University of Regensburg. A large majority of the participants is familiar with service systems and also with the use of mobile applications and therefore the target group was clearly defined. No prior knowledge of event handling, multiple users and processes in the tourism domain was required. In total, 201 people participated in the online survey. Almost 62% of the participants are from the age group between 18 and 30 years, which was the main target group in this questionnaire due to the necessary affinity for mobile applications (Statista 2023). Furthermore, with 35%, single people account for the largest share, while around 34% live in a partnership without children and just under 27% in a partnership with children. To sum up, the broad distribution of participants in terms of age and marital status is elementary to consider the user needs for an event handling strategy for multi user context-aware service systems.

Findings: In this section, we present the results of our empirical study to determine the most important findings for our approach (cf. Section 4). Therefore, important results from the empirical study are summarized in Table 2. In this regard, the user preferences from the online survey of each question were analyzed, generalized and the most significant findings were derived and assigned to the corresponding categories in Table 2.

(A): The first group of findings describes general user preferences with impact to various components of event handling. Thus, 90% of the participants want to decide independently whether or not the service system should perform a re-selection of the originally planned service composition in case of an event. Accompanying, 72% of users would like to be informed immediately after an event. Furthermore, 88% prefer a tolerable limit below which no event processing is performed even though the original defined constraints are exceeded. In this context, the level of the tolerable limit for which a change is accepted depends on the initial value of the constraint (e.g., budget). Specifically, the lower the level of the constraint, the higher the relative tolerable limit (e.g., with a budget of 5 €, a 100% overshoot is tolerated, while with a budget of 80 €, the majority would only tolerate a maximum overshoot of 10%). Moreover, when multiple events occur, 45% would like to process the events collectively according to the severity and 29% according to the chronological order of appearance in the process. Finally, 86% of participants prefer an event processing if the event leads to significant changes (with minimal changes, 86% prefer to keep the original service composition).

(B): The second group describes user preferences for event handling strategies related to context information. The results show that the *event type* is decisive whether an event is considered for processing. Thus, 38% aim to avoid the event in case of a negative change in the weather (i.e., context information) whereas only 8.5% in case of a worsening of the assumed rating of candidate services (i.e., quality of service; 51.2% would tolerate a deterioration of a maximum of one star). Furthermore, the tolerance for time delays depends on the *type of task* in the process. Thus, participants are more likely to accept time delays in tasks that are not time-bounded (e.g., visiting park) than tasks that are more time-bounded (e.g., visiting restaurant). Finally, the consideration of events depends on the *context type*. For example, 86% would like the event to be processed if the weather worsens, while only 29% would like the event to be processed if distances are extended due to a shortage of parking space.

(C): The third group describes user preferences related to multiple users. Therefore, 91% tolerate constraint exceedances (e.g., budget) for a task if the overall budget of the process (i.e., all tasks) for all users is not exceeded. Furthermore, the exceeding of constraints for a single user does not lead to an event processing for almost 60% of the participants. In this regard, whether an event leads to processing depends on the proportion of users affected compared to the total number of users in the process. Thus, 61% of users would refrain from re-selection for an event if less than 40% of the users in the group were affected by a constraint violation.

Table 2. Findings for a user-based event handling strategy from our online survey

Nr.		Finding from online survey	User preferences considering			Event handling		
			Context	Multiple user	General	Fault tolerance	Event processing	General
A	1	Users decide independently how to deal with events	-/-	-/-	**X**	-/-	-/-	**X**
	2	Users want to be informed immediately about events	-/-	-/-	**X**	-/-	-/-	**X**
	3	Tolerable limits are suggested, below which an event is not processed	-/-	-/-	**X**	**X**	-/-	-/-
	4	The tolerable exceedance of defined constraints depends on the absolute value of the constraint (e.g., budget)	-/-	-/-	**X**	**X**	-/-	-/-
	5	Event processing is preferred according to severity or time of appearance	-/-	-/-	**X**	-/-	**X**	-/-
	6	Execution of event processing only in the case of significant changes	-/-	-/-	**X**	**X**	**X**	-/-
B	7	Processing depends on the event type (e.g., context vs. quality of service)	**X**	-/-	**X**	**X**	-/-	-/-
	8	Tolerable limit for context changes (e.g., time delays) depend on the type of task in the process (e.g., visiting restaurant vs. visiting park)	**X**	-/-	-/-	**X**	-/-	-/-
	9	Processing depends on the context type (e.g., weather vs. distance)	**X**	-/-	-/-	**X**	-/-	-/-
C	10	Budget overrun for individual task is accepted if total budget of all users for all tasks in the process is not exceeded	-/-	**X**	-/-	**X**	-/-	-/-
	11	Event should not be handled upon if it affects only one person in the overall group	-/-	**X**	-/-	**X**	-/-	-/-
	12	The proportion of people affected by an event in a group is decisive for the relevance of events	-/-	**X**	-/-	**X**	-/-	-/-

4 Approach for a User-based Event Handling Strategy in Multi User Context-Aware Service Systems

The empirical study in this work further shows that 86% of the participants want to avoid re-selection in general in case of disruptive events. This is also confirmed by the fact that 80% of the respondents tolerate adaptations of maximum 60% of all tasks in a process (e.g., in a process of five tasks, a maximum of three tasks may deviate from the original service composition in regard to selected candidate services). To reduce the number of processed events and thus the number of adaptations, an event handling strategy is required that checks and consequently processes events in such a way that they can be avoided. Thus, in this section, we present our user-based event handling strategy for multi user context-aware service systems (cf. Figure 3).

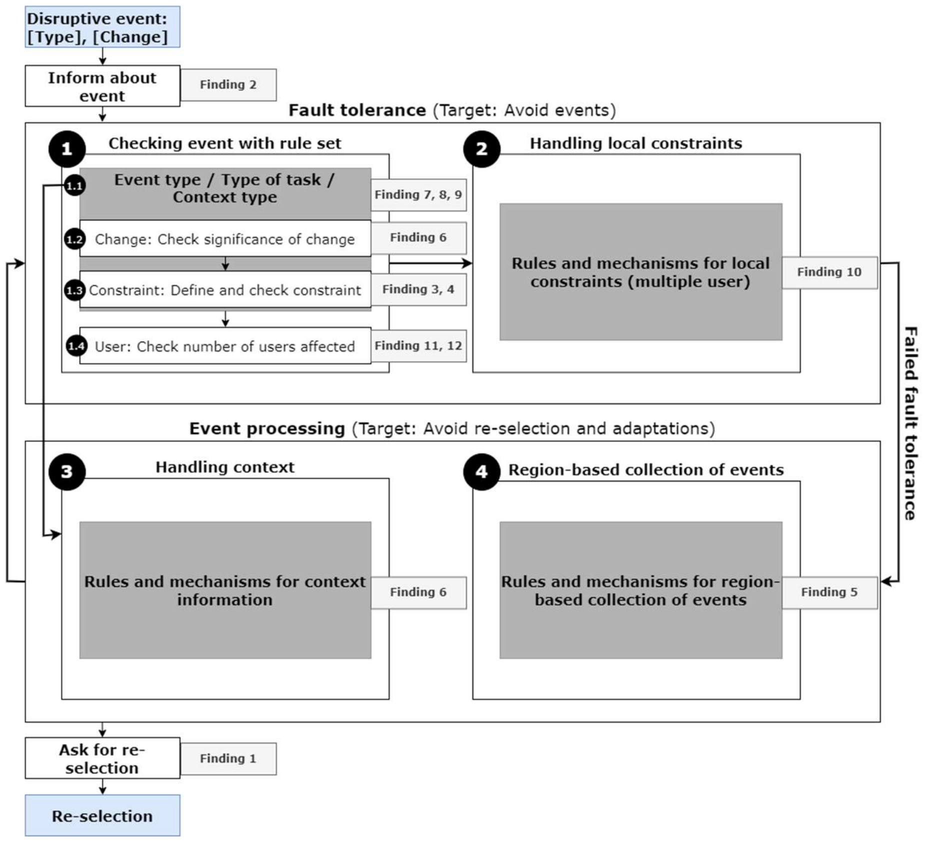

Figure 3. Approach for user-based event handling in multi user context-aware service systems

In this regard, Figure 3 illustrates the main components of our event handling strategy:

Inform about event, *Fault tolerance*, *Event processing* and *Ask for re-selection*. In particular, rules and mechanisms were derived from the generalized findings of our empirical study (cf. Section 3 - Table 2) in order to integrate user-based preferences in our *two-step procedure*. In particular, the individual steps were derived generically and the strategy can therefore be used in a broad context of service systems. In the first step, we try to evaluate the changes with rules and avoid events with mechanisms (i.e., fault tolerance) to minimize the number of processed events. Second, events that cannot be avoided are systematically processed (i.e., event processing) in such a way that adaptations of the originally planned service composition are minimized. The steps of our event handling strategy are described in detail in the following.

Inform about event: In general, there are changes for services at runtime due to an event. The type of event (e.g., change in quality of service) and the amount of change (e.g., ten percent increase in price) are known at runtime. These changes can lead to a re-selection of the planned service composition. In order to provide users with information about the event, all users affected by the change must be informed immediately.

Fault tolerance: Rules and mechanisms of fault tolerance aim to avoid the event, and therefore prevents the event from being processed. In the following, we discuss the two main components of our fault tolerance strategy.

❶ *Checking event with rule set*: A rule set is used to check if an event is a candidate for further processing or whether an event can be avoided by default. These rules are defined individually per user. First, a rule checks whether the constraints of the affected attribute for the specific task (i.e., candidate service) were exceeded due to the change:

- If the constraints were *not exceeded*, a significance level of the change per attribute is defined, which must be exceeded if the event is to be processed (e.g., processing only from a minimum change of 10%). In the case of minor deviations, processing can be dispensed and the event can be avoided per default (cf. Figure 3 – 1.2).
- If the constraints were *exceeded* a rule checks for a tolerable limit. For each attribute, a percentage limit is set for each user, up to which a constraint overshoot is tolerated and thus an event processing is not necessary. In addition, the tolerable limit per user is defined depending on the basic value of the constraint (e.g., the higher the budget, the lower the relative tolerable overshoot of the constraint, cf. Figure 3 – 1.3).

Any change caused by an event must be considered in the context of the *event type*, the *context type* and the *affected task* (cf. Figure 3 – 1.1). For each event type, a clear definition is necessary whether event processing is targeted by the user or the event can be avoided per default. This rule can be further refined, especially for contextual events, by processing an event only for specific events with regard to context information (e.g., only events related to delays in the execution of the process are processed). Furthermore, the type of task (e.g., visiting restaurant vs. visiting park) is also decisive for the level of tolerance. Thus, another rule defines the level of the tolerance limit for certain attributes depending on the affected task. Finally, a rule is required for all users of the process, which specifies the minimum number of users affected by an event. Only if this number of users is exceeded, the event will be processed, even if the event leads to constraint violations for single users (cf. Figure 3 – 1.4).

❷ *Handling local constraints*: For certain types of events (i.e., change in quality of service, e.g., violation of budget constraint), we propose to prevent events from being processed by distributing the value of the constraint exceedance. For example, if there is an overshoot of the budget for a task (e.g., the tolerable limit is 10 € and now visiting a restaurant costs 16 €), this event would have to be processed in general. Our approach tries to distribute the amount of the overshoot (i.e., 6 €) - for each user whose constraints are exceeded - equally among all remaining tasks that have not yet been conducted (e.g., if three tasks have not yet been conducted, 2 € would be transferred to each of these three tasks). Afterwards, the mechanism checks for each user whether all of these tasks still falls below the tolerable limit. When the tolerable limit is undercut, event processing is not necessary for this user. Even if this is not successful for all users, the event may not have to be processed when the number of affected users falls below the required minimum number of users for an event processing (cf. Figure 3 - 1.4).

Event processing: In case an event cannot be avoided in the first step (i.e., fault tolerance), events are processed systematically in order to avoid the number of re-selections and consequently adaptations. In the following, two mechanisms are presented.

❸ *Handling context*: For each event that involves significant context changes (cf. Figure 3 – 1.1/1.2), a service composition is no longer optimal or a candidate service can no longer be conducted in a task. Thus, a re-selection of the service composition at runtime is recommended or even inevitable. However, we extend the event processing with a check of world states in order to reduce the number of adaptations. In particular, for the corresponding task containing the failed candidate service, world states are analyzed whether an alternative candidate service exists, which is valid for the same context information then the previously selected candidate service (i.e., time and location). In case that an alternative candidate service can be found, re-selection can be avoided over the entire process and only one adaptation is necessary (i.e., failed candidate service is exchanged against alternative candidate service in the corresponding task, cf. Bortlik et al. 2023).

❹ *Region-based collection of events*: The user-preferred processing of events is by severity followed by time of appearance (cf. Table 2 – Finding A5). The severity of events has already been considered in the mechanisms for fault tolerance (cf. Figure 3 - 1.2). In our approach, we propose event processing by time of appearance using *regions* (i.e., each task represents a region) to reduce the number of adaptations. In particular, all events (classified by type, e.g., all changes of the quality of service attribute *price*) are collected and processed per task, starting with the first task not executed, and processed sequentially up to the last task of the process. In this regard, an alternative candidate service is searched for in a task, which fulfills all constraints under consideration of all event types. Only if this step fails, a re-selection has to be done on the entire process to get a valid service composition. In this regard, it is checked for each task whether an event is still relevant. When a re-selection has been performed for the entire process, it is possible that adaptations where necessary in various tasks (e.g., replacement of a candidate service) and that events are therefore no longer relevant. Furthermore, events of the type *change in context information* can become irrelevant over time. For example, the weather may change for the better or delays may be reduced in the process. In the course of event processing, all rules of multiple users must be considered (i.e., check number of users affected, cf. Figure 3 - 1.4) to avoid processing events unnecessarily. Finally, the processing according to regions must be repeated continuously, starting from the beginning, since new events can occur regularly.

Ask for re-selection: In the case that the two-step procedure of our approach (i.e., fault tolerance and event processing) cannot prevent a re-selection for one or multiple users due to the occurrence of disruptive events, the entire group must be actively asked whether a re-selection should be performed or the event should be avoided per default.

5 Conclusion, Practical Implications and Future Research

In this paper, we present a user-based event handling strategy for multi user context-aware service systems in mobile environments. In particular, we propose an innovative approach in order to systematically prevent and process disruptive events at runtime with the aim of minimizing the number of processed events and adaptations of already planned service compositions. In the literature, there exists no event handling strategy for service systems that explicitly derives rules and mechanisms based on *user preferences*, especially for *context information* and *multiple user*s. To address this research gap in the area of service systems engineering, we surveyed 201 participants to derive rules and mechanisms for a user-based event handling strategy in mobile environments and also developed an approach for the distribution of local constraints for multiple users (cf. Section 4 – ❷).

For practitioners, our strategy also provides some important benefits. Thereby, our approach can be applied in mobile applications that assist multi user context-aware processes such as a city day trip in the tourism domain. Existing mobile applications (e.g., *Culture Trip*) are not able to control event processing through active customer participation at runtime. Furthermore, by implementing our approach, real-time data (e.g., context information such as publicly available information about the weather) can be used to proactively inform the user about potential future events that have not yet occurred. Finally, service providers (e.g., restaurants) can also benefit from our approach as it enables the active integration of service providers into the event processing mechanism. In particular, service providers can provide possible discounts in a mobile application, which can be suggested to the user depending on their maximum constraints if, for example, the price exceed the tolerable limit (i.e., disruptive event) and the algorithm would select an alternative service provider (i.e., re-selection).

Furthermore, there are also limitations which provide directions for future research. First, further surveys should be conducted, possibly also from other domains in mobile environments, to derive user-based rules and mechanisms even more generic. Second, a rule for tolerable limits was proposed in our approach. However, the limits should be further validated by experiments to evaluate the impact of different limits on the number of processed events. Third, we are the first event handling approach that takes dependencies between multiple users into account. However, our approach is a first idea and needs to be analyzed in more detail. Finally, the event handling strategy should be implemented and its performance evaluated against re-selection approaches that handle each event independently without an event handling strategy (cf. Bortlik et al. 2023).

References

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., Sikora, F. (2015). On the Complexity of QoS-Aware Service Selection Problem. In: A. Barros, D. Grigori, N. C. Narendra et al. (Eds.). Service-Oriented Computing. Berlin, Heidelberg, Springer Berlin Heidelberg, 345–352.

Adams, M., Ter Hofstede, A. H. M., Van Der Aalst, Wil MP, Edmond, D. (2007). Dynamic, extensible and context-aware exception handling for workflows. In: OTM Confederated International Conferences: On the Move to Meaningful Internet Systems, 95–112.

Allied Market Research (2021). Location Based Services Market. Available online at https://www.alliedmarket-research.com/location-based-services-market (accessed 3/24/2024).

Alrifai, M., Risse, T., Nejdl, W. (2012). A hybrid approach for efficient Web service composition with end-to-end QoS constraints. ACM Transactions on the Web 6 (2), 1–31. https://doi.org/10.1145/2180861.2180864.

Alter, S. (2012). Metamodel for service analysis and design based on an operational view of service and service systems. Service Science 4 (3), 218–235.

Angarita, R., Cardinale, Y., Rukoz, M. (2013). Dynamic recovery decision during composite web services execution. In: L. Ladid, A. Montes, P. A. Bruck et al. (Eds.). Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems - MEDES '13, 5th International Conference on Management of Emergent Digital EcoSystems, Luxembourg, Luxembourg, 28.10.2013 - 31.10.2013. New York, New York, USA, ACM Press, 187–194.

Angarita, R., Cardinale, Y., Rukoz, M. (2014). Reliable composite web services execution: towards a dynamic recovery decision. Electronic Notes in Theoretical Computer Science 302 (0), 5–28.

Atteslander, P., Cromm, J. (2010). Methoden der empirischen Sozialforschung. 13., neu bearb. und erw. Aufl. Berlin: Schmidt.

Ayed, D., Asim, M., Llewellyn-Jones, D. (2013). An event processing approach for threats monitoring of service compositions. In: 2013 International Conference on Risks and Security of Internet and Systems (CRiSIS), International Conference on Risks and Security of Internet and Systems (CRiSIS), La Rochelle, France, 1–10.

Bearzotti, L. A., Salomone, E., Chiotti, O. J. (2012). An autonomous multi-agent approach to supply chain event management. International Journal of Production Economics 135 (1), 468–478. https://doi.org/10.1016/j.ijpe.2011.08.023.

Bobek, S., Nalepa, G. J. (2017). Uncertain context data management in dynamic mobile environments. Future Generation Computer Systems 66, 110–124. https://doi.org/10.1016/j.future.2016.06.007.

Böhmann, T., Leimeister, J. M., Möslein, K. (2014). Service systems engineering. Business & Information Systems Engineering 6 (2), 73–79.

Bortlik, M., Heinrich, B., Lohninger, D. (2023). Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime. Information Systems Frontiers 6 (3), 1. https://doi.org/10.1007/s10796-023-10392-8.

Bortlik, M., Heinrich, B., Mayer, M. (2018). Multi User Context-Aware Service Selection for Mobile Environments. Business & Information Systems Engineering 60 (5), 415–430. https://doi.org/10.1007/s12599-018-0552-2.

Buys, J., Florio, V. de, Blondia, C. (2011). Towards context-aware adaptive fault tolerance in SOA applications. In: O. Etzion, A. Gal, S. Zdonik et al. (Eds.). Proceedings of the 5th ACM international conference on Distributed event-based system - DEBS '11, the 5th ACM international conference, New York, USA, 11.07.2011 - 15.07.2011. New York, USA, ACM Press, 63.

Canfora, G., Di Penta, M., Esposito, R., Villani, M. L. (2008). A framework for QoS-aware binding and re-binding of composite web services. Journal of Systems and Software 81 (10), 1754–1769. https://doi.org/10.1016/j.jss.2007.12.792.

Chandler, J. D., Vargo, S. L. (2011). Contextualization and value-in-context: How context frames exchange. Marketing Theory 11 (1), 35–49. https://doi.org/10.1177/1470593110393713.

Cristea, V., Pop, F., Dobre, C., Costan, A. (2011). Distributed Architectures for Event-Based Systems. In: J. Kacprzyk, S. Helmer, A. Poulovassilis et al. (Eds.). Reasoning in Event-Based Distributed Systems. Berlin, Heidelberg, Springer Berlin Heidelberg, 11–45.

Demirkan, H., Bess, C., Spohrer, J., Rayes, A., Allen, D., Moghaddam, Y. (2015). Innovations with Smart Service Systems: Analytics, Big Data, Cognitive Assistance, and the Internet of Everything. Communications of the Association for Information Systems 37. https://doi.org/10.17705/1CAIS.03735.

Fekih, H., Mtibaa, S., Bouamama, S. (2019). The dynamic reconfiguration approach for fault-tolerance web service composition based on multi-level VCSOP. Procedia Computer Science 159 (1), 1527–1536. https://doi.org/10.1016/j.procs.2019.09.323.

Femenia-Serra, F., Neuhofer, B., Ivars-Baidal, J. A. (2019). Towards a conceptualisation of smart tourists and their role within the smart destination scenario. The Service Industries Journal 39 (2), 109–133. https://doi.org/10.1080/02642069.2018.1508458.

Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., Mock, M. (2017). Issues in complex event processing: Status and prospects in the big data era. Journal of Systems and Software 127, 217–236.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G. (2014). A survey on algorithmic approaches for solving tourist trip design problems. Journal of Heuristics 20 (3), 291–328.

Ghallab, M., Nau, D., Traverso, P. (2008). Automated planning. Theory and practice. Amsterdam, Elsevier.

Heinrich, B., Mayer, M. (2018). Service selection in mobile environments: considering multiple users and context-awareness. Journal of decision systems 27 (2), 92–122. https://doi.org/10.1080/12460125.2018.1513223.

Kartiwi, M., Gunawan, T. S. (2013). Framework for Logistics Coordination and Distribution Mobile Application in Disaster Management. In: 2013 International Conference on Advanced Computer Science Applications and Technologies, 2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuching, Malaysia. IEEE, 461–465.

Kum, D. K. (2020). Extended CEP Model for Effective Enterprise Systems Service Monitoring. KSII Transactions on Internet and Information Systems (TIIS) 14 (2), 807–825.

Melo, M., Aquino, G. (2021). FaTEMa: A Framework for Multi-Layer Fault Tolerance in IoT Systems. Sensors (Basel, Switzerland) 21 (21). https://doi.org/10.3390/s21217181.

Michelson, B. M. (2006). Event-driven architecture overview. Patricia Seybold Group 2 (12), 10–1571.

Mo, T., Li, W., Chu, W., Wu, Z. (2011). An Event Driven Model for Context-Aware Service. In: 2011 IEEE International Conference on Web Services, 2011 IEEE International Conference on Web Services (ICWS), Washington, DC, USA. IEEE, 740–741.

Sbai, M., Taktak, H., Moussa, F. (2020). Management of Context in Mobile Environments. In: 2020 20th International Conference on Computational Science and Its Applications (ICCSA), 2020 20th International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy. IEEE, 95–98.

Schnell, R. (2019). Survey-Interviews. Springer.

Shen, Y.-h., Yang, X.-h. (2011). A self-optimizing QoS-aware service composition approach in a context sensitive environment. Journal of Zhejiang University SCIENCE C 12 (3), 221–238. https://doi.org/10.1631/jzus.C1000031.

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., Xu, X. (2014). Web services composition: A decade's overview. Information Sciences 280, 218–238. https://doi.org/10.1016/j.ins.2014.04.054.

Statista (2019a). Number of available apps in the Apple App Store from July 2008 to January 2017. Available online at https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/ (accessed 3/24/2024).

Statista (2019b). Size of mobile professional services market by segment worldwide in 2014 and 2018. Available online at https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/ (accessed 3/24/2024).

Statista (2023). Generationen in Deutschland nach Zustimmung zu Aussagen über Technikaffinität und Technikkenntnisse im Jahr 2021. Available online at https://de.statista.com/statistik/daten/studie/1133513/umfrage/umfrage-zu-technikaffinitaet-und-technikkenntnissen-nach-generationen/#:~:text=Im%20Jahr%202021%20stimmten%20rund,X)%20traf%20dies%20ebenfalls%20zu. (accessed 3/24/2024).

Ventola, C. L. (2014). Mobile devices and apps for health care professionals: uses and benefits. Pharmacy and Therapeutics 39 (5), 356.

Wang, X., Feng, Z., Huang, K., Tan, W. (2017). An automatic self-adaptation framework for service-based process based on exception handling. Concurrency and Computation: Practice and Experience 29 (5), e3984.

Ye, H., Kankanhalli, A. (2020). Value Cocreation for Service Innovation: Examining the Relationships between Service Innovativeness, Customer Participation, and Mobile App Performance. Journal of the Association for Information Systems, 294–312. https://doi.org/10.17705/1jais.00602.

Zeng, L., Lingenfelder, C., Lei, H., Chang, H. (2007). Event-Driven Quality of Service Prediction. In: D. Hutchison, T. Kanade, J. Kittler et al. (Eds.). Service-Oriented Computing – ICSOC 2007. Berlin, Heidelberg, Springer Berlin Heidelberg, 147–161.

Zheng, Z., Lyu, M. R. (2009). A QoS-aware fault tolerant middleware for dependable service composition. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, Networks (DSN), Lisbon, Portugal. IEEE, 239–248.

Zheng, Z., Zhang, Y., Lyu, M. R. (2014). Investigating QoS of Real-World Web Services. IEEE Transactions on Services Computing 7 (1), 32–39. https://doi.org/10.1109/TSC.2012.34.

3.2 Paper 4: Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime

Current Status	Full Citation
This paper is **accepted** and **published in** the journal Information Systems Frontiers.	Bortlik, Michael; Heinrich, Bernd; Lohninger, Daniel (2023): Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime. In Inf Syst Front. DOI: 10.1007/s10796-023-10392-8.

Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime

Michael Bortlik ▪ Bernd Heinrich ▪ Daniel Lohninger

Information Systems ▪ University of Regensburg ▪ Universitätsstraße 31, 93053 Regensburg, Germany

Abstract

Modern service-based processes in mobile environments are highly complex due to the necessary spatial-temporal coordination between multiple participating users and the consideration of context information. Due to the dynamic nature of mobile environments, disruptive events occur at runtime, which require a re-selection of the planned service compositions respecting multiple users and context-awareness. Thereby, when re-selecting services the features performance, solution quality, solution robustness and alternative solutions are essential and contribute to the efficacy of service systems. This paper presents an optimization-based heuristic technique based on a stateful representation that uses a region-based approach to re-select services considering multiple users, context information and in particular disruptive events at runtime. The evaluation results, which are based on a real-world scenario from the tourism domain, show that the proposed heuristic is superior compared to competing artifacts.

1. Introduction

The development of mobile environments in the form of mobile technologies (such as Smartphones, IoT-Devices) and mobile business is steadily increasing (Muhammad et al. 2018; Statista 2019a, 2019b). Services that use context information (e.g., sensory capabilities of mobile devices) to provide individual solutions for users gain in importance, which can be seen, for instance, by the increasing market value of 11.99 billion $ in 2015 to 44.95 billion $ in 2021 for location based services (Allied Market Research 2021). Besides location information, a further important dimension of context is the participating user (user context), which includes the interaction among users (Baccari and Neji 2016; Grotherr et al. 2018; Ma et al. 2015; Roda et al. 2018; Shen and Yang 2011; Tung et al. 2014; Weinert et al. 2020; Zhang et al. 2009). Processes in mobile environments, which include multiple participating users can be found in several domains such as healthcare or disaster relief assistance, field work in companies, everyday efficiency and planning, roadside or in tourism (cf. Gavalas et al. 2014; Neville et al. 2016; Ventola 2014; Zhang et al. 2009). This development supports the construction of *multi user context-aware* service systems, in which mobile technologies enable the realization and support of individual processes.

Determining and realizing processes by an individual service composition for each user at *planning time*, including the selection within a high number of available candidate services, context information (e.g., location or time of day) and the coordination of multiple users, is a known problem in the literature (e.g., Bortlik et al. 2018; Heinrich and Mayer 2018). However, due to the dynamic nature of mobile environments (cf., e.g., Nagarajan and Thirunavukarasu 2020; Sheng et al. 2014), several so-called *disruptive events* may occur *at runtime* (Bobek and Nalepa 2017). Guided by Bearzotti et al. (2012), we define a disruptive event as a real-world event at runtime that significantly change planned values of service candidates, user constraints etc. Thus, disruptive events occurring at runtime can affect all users resulting in service compositions selected at planning time that are no longer optimal if not even no longer feasible. For instance, the disruptive event that a business – selected as service at planning time – is closed and thus become no longer available (cf. Sheng et al. 2014; Zhang et al. 2018; Zheng et al. 2014), results in a need for re-selection at runtime. As the computational effort for determining an optimal solution for such re-selection tasks – considering all possible disruptive events and their changes of planned values – would be very high (the selection problem is known to be NP-hard (non-deterministic polynomial-time hardness); cf. Abu-Khzam et al. 2015; Alrifai et al. 2012), we propose an optimization-based heuristic technique. Thus, our work refers to the following research question:

How to design an optimization-based heuristic technique for re-selecting services under consideration of multiple users, context information and in particular disruptive events at runtime?

To address this research question, we present a novel heuristic technique for service re-selection at runtime for multi user context-aware service systems. Our approach is based on the meta-heuristic local selection (e.g., Gendreau and Potvin 2010) and a stateful representation (cf., e.g., Heinrich and Lewerenz 2015). To enable a performant service re-selection at runtime considering multiple users, context-awareness and in particular disruptive events, the presented approach carefully divides the underlying process into regions and efficiently select a service composition by using feasibility checks and a measure based on the stateful representation. Furthermore, the concept of a stateful representation is substantially extended by a dynamic re-structuring of states at runtime to deal with disruptive events. The evaluation results show that for the presented heuristic technique the computation time increases proportionally in almost all settings, while competing approaches in general have an exponential increase in runtime. More precisely, our technique requires on average only about 1.6% of the computation time compared to existing approaches although achieving a solution quality similar to these approaches (88.8% compared to an optimal solution). This reveals a high performance while retaining a comparable high solution quality. Moreover, our technique achieves an average (solution) robustness of 92%, which describes the proportion of the predetermined services at planning time that can be retained during re-selection at runtime. This represents a significant improvement over existing approaches. Finally, in contrast to existing approaches, our technique is able to present several alternative feasible solutions in near-time especially for multiple participating users. The evaluation results show that an alternative solution can be selected within 6.1 seconds on average for multiple users in case a disruptive event occurs.

The remainder of the paper is organized as follows: In the next section, the background (i.e., Methodical Foundations and Related Work) as well as the research gap are discussed. In the third section, we present our research methodology followed by the introduction of a real-world scenario from the tourism domain in section four. In the fifth section, we propose our model setup and the main components of our heuristic technique, in particular an optimization model and an algorithm for multi user context-aware service re-selection. In the evaluation section, we analyze different features of our approach compared to competing artifacts based on the real-world scenario and a simulation experiment. Subsequently, the results and implications are discussed. Finally, we conclude our paper with a short summary and an outline of limitations and further research.

2. Background and Research Gap

2.1 Methodical Foundations

The selection of several services, thus determining a service composition, leads to a problem which in its basic form (Quality of Service-aware service selection) is related to the selection of exactly one item for each of n available sets of items without violating a family of constraints, while maximizing the overall utility (Caserta and Voß 2019). Thus, the underlying decision problem can be characterized as the well-known MMKP (Multi-Choice Multi-Dimensional Knapsack Problem). The MMKP is an advanced form of the general Knapsack Problem with an additional *multi-choice* property, i.e., the selection of items is performed on sets of items instead of a single item set and further, the resources within a MMKP are *multi-dimensional* since there are multiple resource constraints for the knapsack (Ardagna and Pernici 2006). In our problem context, the multi-choice property is given by the selection of items (=candidate services) grouped in multiple sets (=tasks), for which each item is characterized by a specific value (=utility). Furthermore, the multi-dimensionality is determined by multiple resources (e.g., duration of a candidate service), which constrain the knapsack (=user's service composition). When considering multiple users, the selected candidate services of a user are dependent on the selected candidate services of the other users in terms of optimality (i.e., utility) and feasibility (i.e., constraints). Thus, the considered MMKP is of higher complexity since user-based dependencies have to be additionally modelled when determining each user's service composition.

There are different meta-heuristics in the literature (Gendreau and Potvin 2010) that can be used as a basis for solving the MMKP in the context of multi user context-aware service re-selection. Especially the concepts of the meta-heuristic *Tabu search* are promising for solving this problem, because (1) Tabu search is basically a *local search* strategy (Gendreau und Potvin 2005), which searches the whole neighborhood *deterministically* allowing for a higher degree of solution robustness. Furthermore (2), the existing *memory concept* is highly promising (Gendreau and Potvin 2010) in order to provide access to relevant (existing) information during the iterations of the re-selection and thus enables to efficiently and effectively explore the search space by learning from previous solutions (Blum and Roli 2003; Boussaïd et al. 2013; Gendreau and Potvin 2010; Gogna and Tayal 2013). In particular (2a), the concept of *short-term memory* (Tabu list) facilitates to keep track of the most recently visited solutions and forbids (or allows) specific moves towards them (Blum and Roli 2003). Thus, the neighborhood of the current solution is restricted to the solutions that do not belong to the Tabu list (Blum and Roli 2003). In addition (2b), the concept of *long-term memory* contains a pool of previously generated solutions, which can be used to learn and restart the search (Gendreau and Potvin 2005). This allows in analogy to buffer states of already examined solutions to enable fast decision support at runtime even for problems which are NP-hard. The considered MMKP faces a high degree of complexity which is discussed in literature and caused by dependencies referring to multiple users, context-awareness and disruptive events at runtime (e.g., Heinrich and Mayer 2018):

1. User-based dependencies: Including *multiple users* cause the MMKP to be solved considering each users' service composition (i.e., characterized by multiple tasks of candidate services) in compliance with the user's preferences and constraints. Such preferences and constraints of participating users comprise also temporal-based dependencies arise in case (a subset of) users want to simultaneously select particular candidate services each assigned to a task leading to dependencies between various service compositions. Furthermore, by considering temporal-based dependencies certain candidate services which can (potentially) be added to a service composition are available or not (e.g., depending on temporal availability of a candidate service). Following, changing the candidate services within a service composition of one user affect other service compositions of (subsets of) related users. This requires the inclusion of temporal-based dependencies between multiple users in the optimization model, which in turn leads to additional user-dependent decision variables for each candidate service and thus increases the complexity of the MMKP.

2. Context-based dependencies: Context-based dependencies result from the consideration of *context information* (e.g., location of a candidate service or a user) in a MMKP. Context is usually addressed by the concept of states, which represent the contextual dependencies between candidate services of consecutive tasks. More precisely, the selection of a candidate service within a task can lead to context information, which directly affects the selection of candidate services (for other users) in subsequent tasks (cf. Heinrich and Lewerenz 2015) and thus increase the complexity of the MMKP (Mostofa Akbar et al. 2006; Sbihi 2007). Furthermore, the quantified values for subsequent candidate services (e.g., the distance between the locations of two candidate services), used to calculate the utility, depends on the specific context information.

3. Event-based disruption: As *disruptive events* at runtime can cause a re-selection of one or more service compositions from one or more users, the complexity of the corresponding MMKP increases. In particular, the re-selection and therefore the service compositions are dependent on the number and the type of the occurring disruptive events as well as the affected candidate services, which are unknown at planning time. For example, the disruptive event that a business is actually closed at runtime, which was selected as service at planning time and thus become no longer available, can affect a wide range in the planned service compositions such as exceeding or falling short of a threshold, time conflicts, capacity conflicts, changing context information, changing preferences, candidate service failures or delays. Due to the dynamic nature of disruptive events at runtime and the variety of event types, typically different affected candidate services must be re-selected.

Given these dependencies referring to multiple users, context-awareness and disruptive events, in the literature of service science and re-selection are multiple works, which discuss central features and goals to assess the re-selection solution (Ardagna and Pernici 2007; Cao et al. 2007; Caserta and Voß 2019; Di Napoli et al. 2021; Khan et al. 2002; Mostofa Akbar et al. 2006; Wang et al. 2019; Yu and Lin 2007). The following describes the features to be addressed in more detail:

1. **Performance:** In case of disruptive events, support of the participating users in near-time is needed as waiting times in mobile applications have a negative impact on the users' satisfaction. In particular, a fast and interactive support is a key indicator for usability and thus a key feature for success of mobile applications. Thus, users in mobile environments require near-time support when interacting with mobile applications (Galletta et al. 2004; Hoxmeier and DiCesare 2000; Li and Chen 2019). Therefore, finding a solution at runtime with high performance (near-time optimization) is vital for the efficacy of service systems in mobile environments (Harrison et al. 2013; Saleh et al. 2017; Seffah et al. 2006; Tan et al. 2013).
2. **Solution quality:** Users expect a high solution quality when re-selecting services. On the one hand, finding an optimal service composition in appropriate time is not realistic at runtime due to the high complexity of the underlying decision problem (Moghaddam and Davis 2014; Zhang et al. 2012b). On the other hand, finding only a feasible service composition is necessary but not sufficient. Therefore, a near-optimal solution should be aimed for.
3. **Solution robustness:** Considering re-selection at runtime, fundamental changes to the predetermined service composition at planning time have a negative impact on the users' satisfaction (Barber and Salido 2015; Rahmani and Ramezanian 2016; Seffah et al. 2006; Tan et al. 2013). Therefore, users expect a largely robust service composition at runtime, where in case of disruptive events a small number of changes occur to the predetermined service composition (Barber and Salido 2015; Rahmani and Ramezanian 2016).
4. **Alternative solutions:** The provided service composition at runtime may not meet the users expectations due to, for example, imprecise or incomplete context information or changing user preferences (David et al. 2014). Thus, determining and presenting alternative feasible solutions in near-time is important. Based on such alternative solutions, proactive user interaction at runtime is possible (Evers et al. 2014) to give the user the opportunity to change the service composition proposed in the, for instance, mobile environment.

As discussed in literature, these features contribute to the efficacy of multi user context-aware service systems since the response to disruptive events and the corresponding adaptation of service compositions represent a key role in the design of these systems (Alter 2017; Faieq et al. 2021; Hidri et al. 2019). However, existing approaches for multi user context-aware service systems primarily focus on the (adaptation) requirements of the service provider. As a result, the perspective of the users (i.e., service consumers, cf. Faieq et al. 2021) and their required features (e.g., high performance, cf. Fkaier et al. 2017) are mostly not included in the design of service systems. Therefore, addressing these features is highly important for research and practice, especially for the further development of multi user context-aware service systems (Fkaier et al. 2017).

2.2 Related Work

In the following, we discuss existing approaches from literature, which can be used in general for service re-selection. Therefore, we applied a literature search process consisting of three phases (cf. Fig. 1). The discussion is based on a literature search of related work (Phase 1) conducted in aisnet.org, Web of Science, ACM, IEEE Xplore, INFORMS, ScienceDirect and Springer, which was performed using 36 keywords resulting in 156 papers (Task I). Moreover, in order to identify further relevant works, we also conducted a backward and forward search based on these papers resulting in 35 further articles and 191 articles in sum (Task II). After a more detailed analysis of the relevant articles (Phase 2) by screening title, keywords and abstracts, 69 papers remained (Task III). Based on reading introduction and summary, we only considered works that contain dependencies between multiple users and context information resulting in 53 articles (Task IV). A detailed text analysis resulted in 38 relevant approaches that are meet our research topic in general (Task V). Thereupon, we structure the literature (Phase 3) firstly on the basis whether the approaches consider multiple users, context-awareness and in particular disruptive events and secondly based on the four central features discussed in the methodical foundations (cf. Section 2.1). A summary of the systematized related work can be found in Table 1 (Task VI), which is explicitly created based on the dependencies and features introduced in Section 2.1.

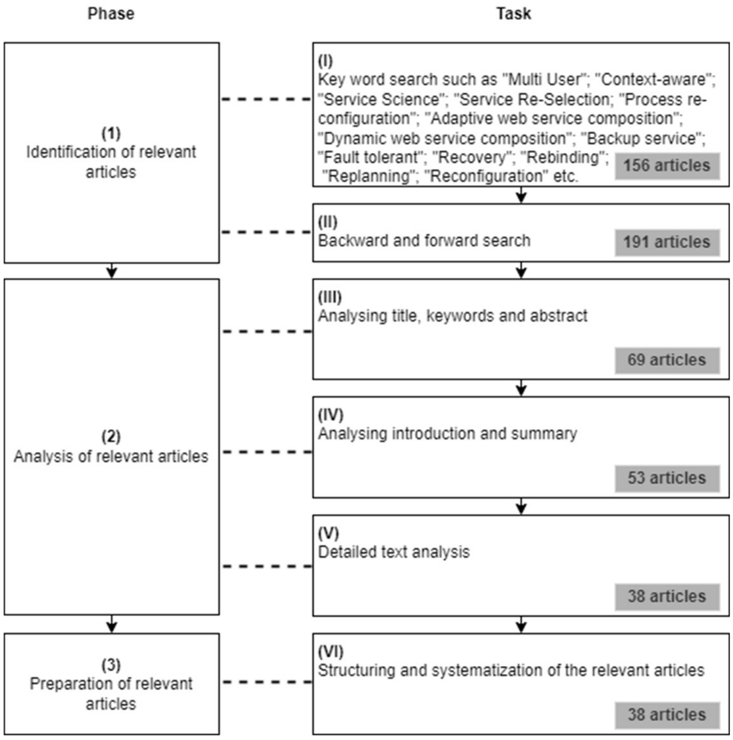

Fig. 5 *Literature search process*

(A): The first group of service system approaches mainly deal with *user-based dependencies* (e.g., Mayer 2017; Wanchun et al. 2011). With regard to multiple users, there are several approaches that allow the same candidate services to be used by multiple users (Heinrich et al. 2015; Mayer 2017; Wanchun et al. 2011; Wang et al. 2010; Wang et al. 2014), while other approaches address capacity limits of services in order to provide multi user support (He et al. 2012; Jin et al. 2012; Kurdija et al. 2019; Pang et al. 2020; Zhu et al. 2017). However, only Mayer (2017) presents an approach for service re-selection in order to cope with disruptive events occurring at runtime but do not deal with context-based dependencies at all. An analysis of the features shows, that some of the works aim at *performance* and therefore He et al. (2012), Heinrich et al. (2015), Jin et al. (2012), Kurdija et al. (2019), Mayer (2017), Wang et al. (2010) and Zhu et al. (2017) can in general provide results in near-time. In regard to *solution quality*, all of the approaches address as a foremost goal the selection of feasible solutions (e.g., Pang et al. 2020). Additionally, these works can be divided in exact approaches (i.e., providing an optimal solution, e.g., Wang et al. 2014) and heuristic techniques (i.e., providing a near-optimal solution, e.g., Kurdija et al. 2019). For the re-selection at runtime the use of heuristics is clearly favored, because they can enable near-optimal solutions while reducing the computational effort. Nevertheless, Mayer (2017) uses an exact approach at runtime to obtain the optimal solution. Finally, none of the existing works contributes to the features *solution robustness* as well as *alternative solutions*.

(B): Furthermore, there is a group of several service system approaches that deal with *context-awareness* (e.g., Faieq et al. 2019; Lewerenz 2015; Shen et al. 2012) and thus address context dependencies between various candidate services based on available context information such as price or distance. Existing literature that deals with context-awareness can be further divided into selection and re-selection works (including *disruptive events* at runtime). In the service systems literature with regard to context-awareness various types of re-selection approaches have been proposed such as fault tolerant strategies (Angarita et al. 2013, 2014; Fekih et al. 2019b; Zheng and Lyu 2009), process reconfiguration (Shen and Yang 2011; Zhang et al. 2012a) and adaptive web service compositions (Aouatef et al. 2008; Ardagna and Pernici 2007; Buys et al. 2011; Cao et al. 2015; Cherif et al. 2019; Ma et al. 2015; Sandionigi et al. 2013; Sedighiani et al. 2021; Tretola and Zimeo 2019). In a large part of these approaches, *performance* is an important feature to assess the solution (e.g., Bortlik et al. 2018; Xu and Jennings 2010). In contrast, near-time optimization in order to enable support at runtime is not addressed by any of the presented works. The analysis of the *solution quality* shows that all approaches enable a feasible solution. Furthermore, only a few articles can provide an optimal solution (i.e., Heinrich and Lewerenz 2015; Sandionigi et al. 2013; Shen et al. 2012; Xu and Jennings 2010; Zheng and Lyu 2009), while most of the approaches describe a heuristic technique in order to achieve a near-optimal solution (e.g., Fekih et al. 2019a). Besides that, Angarita et al. (2013), Angarita et al. (2014), Ardagna and Pernici (2007), Sandionigi et al. (2013), Shen and Yang (2011), Zhang et al. (2012a) and Zheng and Lyu (2009) focus on the feature *solution robustness*. However, just Zhang et al. (2012a) describes explicitly the reduction of changing services as goal. All other approaches only implicitly deal with the feature solution robustness without explicitly presenting techniques to improve the robustness of solutions. Finally, none of the considered approaches contributes to the feature *alternative solutions*.

(C): Finally, there is only a limited amount of articles that deal with *user-based as well as context-based dependencies* (Bortlik et al. 2018; Heinrich and Mayer 2018). Both approaches allow the same candidate services to be used by multiple users while respecting context dependencies between these candidate services. Nevertheless, none of these approaches discusses event-based disruptions. Considering the feature *performance,* it becomes apparent that both approaches cannot enable near-time optimization. Furthermore, in regard to *solution quality*, Heinrich and Mayer (2018) presents a technique to obtain an optimal solution, whereas Bortlik et al. (2018) focus on a heuristic technique. Finally, neither approach contributes to the features *solution robustness* and *alternative solutions*. A detailed comparison of our approach with the approach by Bortlik et al. (2018) showing significant differences can be found in the appendix of this work.

In summary, none of the identified works provides an approach that can select solutions with fast support, high quality and robustness while respecting multiple users, context-awareness and in particular disruptive events.

2.3 Research Gap

As discussed in the related work, there are several service system approaches that deal with multiple users, context-awareness or disruptive events. However, a multi user context-aware re-selection approach that copes with all these concepts, especially addressing the feature performance and near-time optimization at runtime, is – to the best of our knowledge – missing so far. At runtime, there are high user demands on the solution of a re-selection, in particular solutions in near-time (i.e., performance), solution quality, solution robustness and alternative solutions. Existing service system approaches from the literature indeed refer to the complexity of user-based and context-based dependencies but cannot address high-performance and robust solutions at runtime while maintaining a high solution quality. However, as discussed in the methodical foundations, these features are important for designing service systems. In particular, features contribute to the efficacy of multi user context-aware service systems since the response to disruptive events and the corresponding adaptation of service compositions represent a fundamental part in the design of these service systems (Alter 2017; Faieq et al. 2021; Hidri et al. 2019). Addressing these features is highly important in research as well as in practice, especially for the further development of multi user context-aware service systems (Fkaier et al. 2017). To deal with these features, we propose an optimization-based heuristic technique that uses a sophisticated region-based approach to re-select service compositions considering multiple users, context-awareness and in particular disruptive events at runtime. In this regard, we present the first re-selection approach that apply a heuristic on a stateful representation. Thus, the represented context information within these states (i.e., state space) enable our heuristic for near-time optimization as well as feasibility checks and a state space measure. Furthermore, the concept of a state space is significantly extended by a dynamic state space re-structuring at runtime to deal with disruptive events.

Table 1. Overview of existing approaches

	Service system approach	Approaches considering			Features of the solution						
					1. Solution quality			2. Solution robustness	3. Performance		4. Alternative solutions
		User-based dependencies	Context- based dependencies	Event-based disruption	Optimal solution	Near-optimal solution	Feasible solution		Perfor-mance	Near-time optimization	
A	Wanchun et al. (2011), Wang et al. (2014)	X	-/-	-/-	X	-/-	X	-/-	-/-	-/-	-/-
	He et al. (2012), Heinrich et al. (2015)	X	-/-	-/-	X	-/-	X	-/-	X	X	-/-
	Jin et al. (2012), Kurdija et al. (2019), Wang et al. (2010), Zhu et al. (2017)	X	-/-	-/-	-/-	X	X	-/-	X	X	-/-
	Pang et al. (2020)	X	-/-	-/-	-/-	X	X	-/-	-/-	-/-	-/-
	Mayer (2017)	X	-/-	X	X	-/-	X	-/-	X	X	-/-
B	Heinrich and Lewerenz (2015), Shen et al. (2012), Xu and Jennings (2010),	-/-	X	-/-	X	-/-	X	-/-	X	-/-	-/-
	Ai and Tang (2008), Deng et al. (2016), Faieq et al. (2019), Fekih et al. (2019a), Lewerenz (2015), Yu and Reiff-Marganiec (2009), Zhang et al. (2013a)	-/-	X	-/-	-/-	X	X	-/-	X	-/-	-/-
	Zhang et al. (2013b)	-/-	X	-/-	-/-	X	X	-/-	-/-	-/-	-/-
	Sandionigi et al. (2013), Zheng and Lyu (2009)	-/-	X	X	X	-/-	X	X	X	-/-	-/-
	Cao et al. (2015)	-/-	X	X	-/-	X	X	-/-	-/-	-/-	-/-
	Fekih et al. (2019b)	-/-	X	X	-/-	X	X	-/-	X	-/-	-/-
	Ardagna and Pernici (2007), Shen and Yang (2011)	-/-	X	X	-/-	X	X	X	X	-/-	-/-
	Aouatef et al. (2008), Buys et al. (2011), Ma et al. (2015), Sedighiani et al. (2021), Tretola and Zimeo (2019)	-/-	X	X	-/-	-/-	X	-/-	-/-	-/-	-/-
	Zhang et al. (2012a)	-/-	X	X	-/-	-/-	X	X	-/-	-/-	-/-
	Cherif et al. (2019)	-/-	X	X	-/-	-/-	X	-/-	X	-/-	-/-
	Angarita et al. (2013), Angarita et al. (2014)	-/-	X	X	-/-	-/-	X	X	X	-/-	-/-
C	Heinrich and Mayer (2018)	X	X	-/-	X	-/-	X	-/-	X	-/-	-/-
	Bortlik et al. (2018)	X	X	-/-	-/-	X	X	-/-	X	-/-	-/-

3 Research Methodology

Our work is found in general on the quantitative research paradigm (cf. Meredith et al. 1989; Will M. Bertrand and Fransoo 2002). In the following, we describe each step of this approach with regard to the work at hand in detail (cf. Fig. 2):

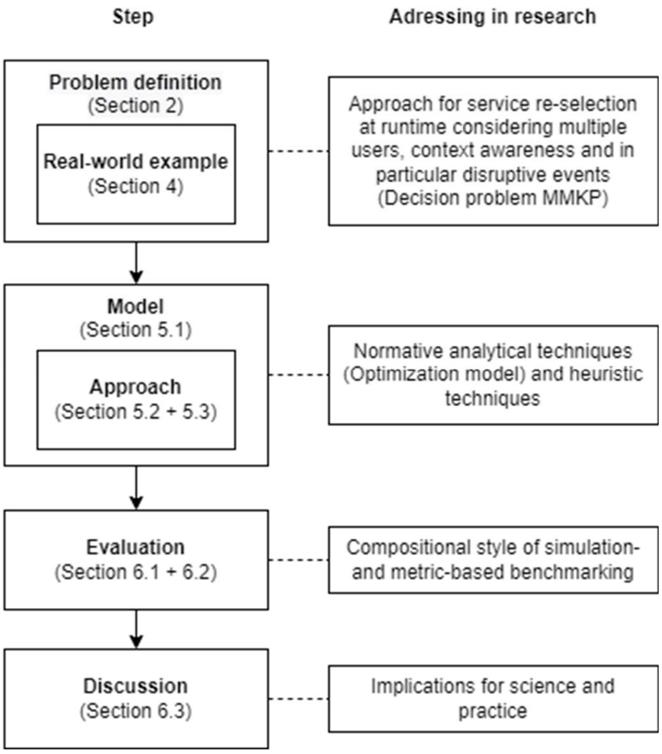

Fig. 6 Quantitative research approach

The first step *problem definition* includes the description and discussion of the subject and topic of the study with respect to existing knowledge bases and foundations (cf. Will M. Bertrand and Fransoo 2002). Therefore, we ground our research on *multi user context-aware service re-/selection* in the literature under the well-known decision problem MMKP including user-based as well as context-based dependencies and in particular disruptive events at runtime (cf. Section 2.1). Moreover, the scope and the differences of our work have to be discussed compared to existing works that study a related problem (cf. Section 2.2), resulting in a research gap (cf. Section 2.3). To address the illustration of the problem definition (cf. Will M. Bertrand and Fransoo 2002), we use a mobile-enabled process from the tourism sector, which represents a valuable part in the further development of service design (cf. Section 4). Based upon this problem definition, we introduce the basic *model notation* as foundation in Section 5.1. In this regard, an optimization model is established, which enables a solution for the underlying decision problem (cf. Section 5.2). Next, we present an algorithm to solve the optimization model efficiently with focus on central features (cf. Section 5.3). More precisely, normative analytical techniques (i.e., optimization model; cf. Meredith et al. 1989) and algorithms (i.e., heuristic techniques; cf. Will M. Bertrand and Fransoo 2002) are used for the multi user context-aware service re-selection in order to select a near-optimal solution at runtime with high performance. In the evaluation it is assessed in detail how well the proposed algorithm supports to solve the represented mathematical model (Tedeschi 2006). This evaluation can be done by means of mathematical techniques, simulation, benchmarking etc. and the results can be compared to actual measured results from other approaches (Prat et al. 2015). Thus, the design of our evaluation follows the compositional styles demonstration as well as simulation- and metric-based benchmarking of artifacts (Prat et al. 2015), in which the efficacy, performance or robustness of the artifact is measured and compared with those resulting from other approaches (cf. Section 6.1 and 6.2). Finally, the discussion of the solution, its effectiveness and importance must be presented appropriately to researchers and other relevant audiences such as practitioners. Therefore, Section 6.3 of this paper discusses the results and implications for science and practitioners.

4 Real-world Scenario

The importance of service systems and the associated support of mobile-enabled processes is continuously increasing (Deng et al. 2016) and in this regard, the literature shows that the tourism sector including mobile-enabled processes are gaining in importance, too (Femenia-Serra et al. 2019). Thus, tourism represents a valuable part of the further development of service systems and, in particular, of service design (Koskela-Huotari et al. 2021). To illustrate our approach in the following, we introduce a scenario, which considers an excerpt of a mobile-enabled process of a tourist portal based on a day trip. Therefore, we adopt the structure of this process directly from the service science literature (Hara et al. 2016; Oizumi et al. 2013). In particular, these works describe a day trip process that is composed of a set of activities (e.g., eating) and each associated location of an activity is connected by a transport (e.g., bus). In this regard, a process is further defined as a set of tasks arranged in a temporal order (Corradini et al. 2007) and contains the following tasks in our scenario: *Breakfast or Café*, *Restaurant*, *Museum* or *Park* (cf. Fig. 3).

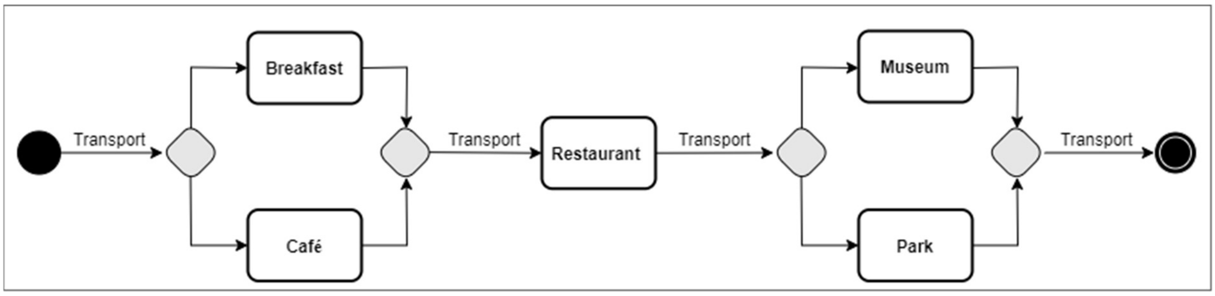

Fig. 7 Excerpt of a process model for a day trip (guided by Hara et al. 2016; Oizumi et al. 2013)

In the following, we instantiated this process for a day trip based on real-world data of the city of Melbourne (Australia). More precisely, in our scenario, three users participate in the illustrated process and can perform certain tasks such as Breakfast or Café independently of each other but can also conduct tasks together (based on their preferences). A task which is in the focus of the potential simultaneous execution among multiple users is defined as so-called *focus task* in the following. For each (focus) task different candidate services are available. These candidate services (e.g., the Restaurants "Da Guido 365" and "Little Billy" for task *"Restaurant", cf. Fig.* 3) built on real-world data of the city of Melbourne and are described for example by *name*, *duration*, *opening hours, price, weather suitability* and *GPS coordinates*. In the city of Melbourne there are a plethora of candidate services per task that users can select during a city day trip. For example, in downtown Melbourne there are about 1,250 restaurants, 1,250 cafés, 250 candidate services for arts such as museum, 105 sights such as parks and 220 locations for having breakfast. Given this high number of real candidate services, there exist on average about 1.2 billion possible solutions (i.e., service compositions) per user to realize the city day trip demonstrating the complexity of the selection task. In addition, each user in the process must be transported from a selected candidate service to the next candidate service. This transport is conceptualized by its own type of a task (i.e., transport task) and contains candidate services described by attributes such as *name*, *duration*, *price*, *type* or *favorite score*. Thus, additionally to the five tasks already introduced above, four more transport tasks have to be considered, resulting in nine tasks in sum. In Melbourne there is a wide range of candidate services for transport and therefore cars (including car sharing), bicycles (including bike sharing), walks and Melbourne's public transport, in particular metropolitan trams, metropolitan trains and the Melbourne city bus, can be used. Besides tasks and their candidate services, world states are required to cope with context-awareness (cf. Ghallab et al. 2008; Heinrich and Schön 2015). In particular, each user has an initial world state based on the initial context (e.g., time and GPS position to start the city day trip) and a corresponding end world state (e.g., time and GPS position at the end of the city day trip).

Table 2. Details of the presented real-world example of *Melbourne*

Aspect (as used in service science literature)	*Real-world example*
Process	Day trip in the city of Melbourne containing nine tasks
Task	*Task 1:* Transport *Task 2 / 3:* Breakfast or Café *Task 4:* Transport *Task 5:* Restaurant *Task 6:* Transport *Task 7 / 8:* Museum or Park *Task 9:* Transport
User	Three users can perform certain tasks such as Breakfast or Café independently of each other but can also conduct tasks together (based on their preferences).
User Preference	Favorite scores for each type of candidate service, etc.
Candidate Service *(based on real-world data)*	*Task 2 / 3:* 1,250 Cafes and 220 Locations for having breakfast *Task 5:* 1,250 Restaurants *Task 7 / 8:* 250 Museums and 105 Sights *Task 1, 4, 6, 9:* Car, Bicycle, Walk, Public Transport
Attribute *(based on real-world data)*	Name, duration, opening hours, price and type for each candidate service, etc.
Context *(based on real-world data)*	Weather suitability and GPS coordinates for each candidate service, etc.
Event Type	Disruptive Events such as closed restaurant, changing weather, exceeding opening hours, etc.
Event Occurrence *(based on real-world data)*	Disruptive Events depend on: Ø congestion level in Melbourne per year: approx. 23% Ø share of public transport delay in Melbourne per year: approx. 14.4% # of average rainy days in Melbourne per year: approx. 100 etc.

During a city day trip different types of disruptive events can occur such as closed restaurants, changing weather conditions (e.g., it starts raining) or exceeding opening hours due to delays in the process. Indicators for the occurrence of such disruptive events can be obtained from publicly available information on Melbourne. For example, in 2020 there was a total congestion level of 23%, which extends a 30-minute drive of the transport candidate service *car* to an average *duration* of 37 minutes, while at rush hour the congestion level can even reach 39% (TomTom 2021). Furthermore, 14.4% of all transports in Melbourne of the transport type “metropolitan tram” were not in time in 2019 and therefore the passengers arrived at their destination at least six minutes late (Public Transport Victoria 2019) also leading to an increase in the attribute *duration*. In addition, in 2020 Melbourne had an average of 100 rainy days per year (Australian Government 2021), potentially influencing the attribute *weather suitability* of a candidate service (e.g., the restaurant “Da Guido 365” has a weather suitability of “sunshine” because there is only outdoor seating available). Such disruptive events enforce a re-selection at runtime in case the predetermined service compositions of multiple users at planning time are no longer feasible for instance. The most important details from the presented real-world example are summarized in Table 2.

5 Model

5.1 Basic Notations

In this section, we introduce the basic notations for our multi user context-aware service system that can serve as a foundation for the service re-selection approach.

1 We consider a process p referring to a set of tasks T containing all tasks of the process p and $t \in T$ with t describing a single task. A process p is conducted by one user $a \in A$ or multiple users with A describing the set of users.

2 There is a set of focus tasks F with $F \subseteq T$, which can be conducted by multiple users simultaneously. Furthermore, there is a set of tasks I with $I \subseteq T$, which are executed by each user individually (non- simultaneously). In particular, it holds: $I \cup F = T \wedge I \cap F = \emptyset$. As the sets of tasks T, F and I are conducted by each user, we denote these sets also as T_a, F_a and I_a when referring to a user $a \in A$.

3 Each task t refers to a set of candidate services CS_{at} including the functional equivalent candidate services for a user a and a task t. Thereby, a single candidate service is defined as $cs_{ats} \in CS_{at}$, with s describing the index of the corresponding candidate service.

4 Each candidate service cs_{ats} is characterized by a set NCA of non-context-aware attributes with $nca_n \in NCA$ describing one single attribute such as rating and with n describing the index of the corresponding non-context-aware attribute and a set CA of context-aware attributes with $ca_l \in CA$ describing one single attribute such as time of day and with l describing the index of the corresponding context-aware attribute. The sets NCA and CA form the set of non-functional properties NFP (with $NCA \cup CA = NFP$).

5 To address context-awareness, we use the concept of world states (cf. Section 4). In detail, the set of world states WS_{at} represent the possible context information of user a in a task t. Each world state is defined as ws_{atk} with $ws_{atk} \in WS_{at}$ and with k describing the index of the corresponding world state.

6 To determine the value of a context-aware attribute ca_l the combination of a candidate service and a world state is needed (e.g., GPS coordinates and service destination). This combination refers to a world-state-candidate-service combination wsc_{aty} with $wsc_{aty} \in WSC_{at}$ where WSC_{at} is the set of all world-state-candidate-service combination for a user a in a task t and y describing the index of the corresponding world-state-candidate-service combination.

7 To enable spatial-temporal coordination between multiple users for each focus task in the set of focus tasks F, we use the concept of common world-state-candidate-service combination. A world-state-candidate-service combination wsc_{aty}, which refers to the same world state ws_{atk} and candidate service cs_{ats} by all users of set A is defined as a common world-state-candidate-service combination $cwsc_{tz}$. Thereby, it holds $cwsc_{tz} \in CWSC_t$ with $CWSC_t$ is the set of all common world-state-candidate-service combinations for the focus task t and z describing the index of the corresponding common world-state-candidate-service combination.

8 A service composition sc_a is noted as a realization of a process p for user a in the form of a set of world-state-candidate-service combinations and a set of common world-state-candidate-service combinations with exactly one (common) world-state-candidate-service combination for each task t of the process p.

9 A service composition sc_a for a user a can refer to a global constraint $con_a^{nca_n}$ for each non-context-aware attribute nca_n and to a global constraint $con_a^{ca_l}$ for each context-aware attribute ca_l.

10 We define the value of a non-context-aware attribute related to a candidate service cs_{ats} as $q_{ats}^{nca_n}$ (e.g., for a specific restaurant, the average star rating is 4.8) and the value of a context-aware attribute ca_l related to a world-state-candidate-service combination wsc_{aty} as $q_{aty}^{ca_l}$ (e.g., time of day to visit a restaurant is 12 a. m.). A function to aggregate the quantified values $q_{ats}^{nca_n}$ and $q_{aty}^{ca_l}$ for all attributes to a single utility value is represented by a utility function U. Here, we adopt the widely used utility function described in detail by Alrifai and Risse (2009), which applies the SAW (simple additive weighting) method.

11 At runtime a disruptive event ev_{ad} can occur with $d = 1\ to\ D_a$ and D_a defining the number of events for user a. An event ev_{ad} can affect an arbitrary candidate service cs_{ats} or several candidate services, for instance, an event causes a candidate service to become no longer available.

5.2 Optimization Model

To consider multiple users, context-awareness and in particular disruptive events in our approach for service re-selection at runtime, we establish a stateful representation by explicitly modelling a state space containing world states and candidate services. In the following, we present our optimization model for service re-selection at runtime.

Objective Function:

$$\sum_{a \in A} \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} U(wsc_{aty}) * wsc_{aty} \rightarrow max \quad (1)$$

s. t.:

one CS per Task:

$$\sum_{s=1}^{CS_{at} \setminus BL} cs_{ats} = 1 \ \forall\, t = 1, \ldots, T_a;\ \forall\, a = 1, \ldots, A \quad (2)$$

one WS per Task:

$$\sum_{k=1}^{WS_{at}} ws_{atk} = 1 \ \forall\, t = 1, \ldots, T_a;\ \forall\, a = 1, \ldots, A \quad (3)$$

one WSC per Task:

$$\sum_{y=1}^{WSC_{at}} wsc_{aty} = 1 \,\forall\, t = 1, \ldots, T_a;\ \forall\, a = 1, \ldots, A \quad (4)$$

WSC Constraint:

$$\sum_{y=1}^{WSC_{at}} CS(wsc_{aty}) + WS(wsc_{aty}) - 2 * wsc_{aty} \geq 0 \,\forall\, t = 1, \ldots, T_a \,\forall\, a = 1, \ldots, A \quad (5)$$

WS Constraint:

$$\sum_{y=1}^{WSC_{at}} CREATE\_WS(wsc_{aty}) - wsc_{aty} \geq 0 \,\forall\, t = 1, \ldots, (T_a - 1);\ \forall\, a = 1, \ldots, A \quad (6)$$

one CWSC per Focus Task:

$$\sum_{z=1}^{CWSC_t} cwsc_{tz} = 1 \,\forall\, t = 1, \ldots, F \quad (7)$$

Focus Task Constraint:

$$\sum_{z=1}^{CWSC_t} (|A| * cwsc_{tz} - \sum_{a=1}^{A} CWSC(a, cwsc_{tz})) = 0 \,\forall\, t = 1, \ldots, F \quad (8)$$

Non-Context-aware Attribute Constraints:

$$min,\ \sum_{t=1}^{T_a} \sum_{s=1}^{CS_{at}} q_{ats}^{nca_n} * cs_{ats} \leq con_a^{nca_n} \,\forall\, nca_n \in NCA^-;\ \forall\, a = 1, \ldots, A \quad (9)$$

$$max, \sum_{t=1}^{T_a} \sum_{s=1}^{CS_{at}} q_{ats}^{nca_n} * cs_{ats} \geq con_a^{nca_n} \,\forall\, nca_n \in NCA^+;\ \forall\, a = 1, \ldots, A \quad (10)$$

Context-aware Attribute Constraints:

$$min,\ \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} q_{aty}^{ca_l} * wsc_{aty} \leq con_a^{ca_l} \,\forall\, ca_l \in CA^-;\ \forall\, a = 1, \ldots, A \quad (11)$$

$$max,\ \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} q_{aty}^{ca_l} * wsc_{aty} \geq con_a^{ca_l} \,\forall\, ca_l \in CA^+;\ \forall\, a = 1, \ldots, A \quad (12)$$

Our optimization model consists of an objective function (1) determining the overall utility value and the constraints (2) to (12). The objective function (1) determines the overall utility by summing up the utility scores $U(wsc_{aty})$ of all selected world-state-candidate-service combinations, over all tasks in T_a and all users in A in the process p with the goal to maximize the accumulated utility value over all selected users' service compositions. To consider only the selected world-state-candidate-service combinations, we use a decision variable wsc_{aty} for each world-state-candidate-service combination where wsc_{aty} = 1 notes that wsc_{aty} is selected and $wsc_{aty} = 0$ that wsc_{aty} is not selected. For re-selecting, constraint (2) ensures that exactly one candidate service cs_{ats} from each task in T_a is selected for each user $a \in A$, which is not on a blacklist BL containing all candidate services that cannot be used for a re-selection due to an event ev_{ad}. To hold the condition that for each user $a \in A$ and for each task in T_a exactly one world state ws_{atk} must be selected, the constraint (3) is also be part of our optimization model. Constraint (4) is used to assure that exactly one wsc_{aty} for each user $a \in A$ is selected for each task in T_a. The constraint (5) links each wsc_{aty} to its corresponding cs_{ats} and ws_{atk} (cf. Section 5.1), which is realized by $CS(wsc_{aty})$ and $WS(wsc_{aty})$, within each task in T_a and for each user in A. Further, by constraint (6) is ensured that the resulting world state in the subsequent task (represented by $CREATE\_WS(wsc_{aty})$) for a wsc_{aty} also has to be selected within each task in T_a (except the last task) and for every user in A.

Constraint (7) integrates the concept of common world-state-candidate-service combinations, which is necessary to ensure the simultaneous use of the same world state ws_{atk} and candidate service cs_{ats} by multiple users $a \in A$ with respect to user-based dependencies (cf. Section 2.1). Therefore, we guarantee that exactly one $cwsc_{tz}$ for each focus task in F is selected for all users participating in the process. The constraint (8) checks whether all world-state-candidate-service combinations regarding a selected $cwsc_{tz}$ (i.e., we use the decision variable $cwsc_{tz}$ that is 1 if the corresponding $cwsc_{tz}$ is selected and 0 if not) are used in each users' service composition (represented by $CWSC(a, cwsc_{tz})$).

To ensure feasible solutions, the constraints (9), (10), (11) and (12) consider the global end-to-end constraints for non-context-aware attributes $nca_n \in NCA$ and context-aware attributes $ca_l \in$ CA defined for each user $a \in A$. In this regard, the sets NCA and CA are each divided into a subset NCA^- respective CA^- (with $NCA^- \subseteq NCA$ and $CA^- \subseteq CA$) of attributes that need to be minimized and a subset NCA^+ respective CA^+ (with $NCA^+ \subseteq NCA$ and $CA^+ \subseteq CA$) that need to be maximized. In particular, the constraints (9) and (10) describes if all selected candidate services (i.e., in this regard, we use the decision variable cs_{ats} that is 1 if the corresponding candidate service cs_{ats} is selected and 0 if not) from each task in T_a of the process meet the users' (a) constraints $con_a^{nca_n}$ in regard to all non-context-aware attributes $nca_n \in NCA$. Therefore, all minimizing attributes must not exceed a defined upper constraint with regard to their value $q_{ats}^{nca_n}$ (e.g., the maximum price limit for a user to have lunch in a restaurant is 10 Euro, cf. constraint (9)). Likewise, all maximizing attributes must not fall below a defined lower constraint with regard to their value $q_{ats}^{nca_n}$ (e.g., the average rating of a restaurant where the user is having lunch should be at least 4 stars, cf. constraint (10)). In order to incorporate context-based dependencies, the constraints (11) and (12) describe analogously the feasibility to the global end-to-end constraints of context-aware attributes, which means that the value $q_{aty}^{ca_l}$ is dependent on the selected wsc_{aty} for each task in T_a.

An overview of the formal notation of the optimization model can be found in the appendix of this work.

5.3 Algorithm for multi user context-aware service re-selection

The optimization model introduced above relies on a state space comprising candidate services, world states and resulting world-state-candidate-service combinations. This state space must be generated starting with the initial world state of each individual user and then determining the corresponding state transitions throughout the process until the end world state is reached (Bortlik et al. 2018; Ghallab et al. 2008; Heinrich and Schön 2015). After that, its size must be delimited in such a manner that it allows the optimization model to find a near-optimal solution at runtime with high performance. The algorithm presented in this section addresses this generation of the state space. At the core of the algorithm, the part of the state space in which the disruptive event occurs is generated by a region-based approach in order to enable a performant selection of a solution (Section 5.3.1). In case that no feasible solution exists in the initial region, the algorithm carefully extends to further regions based on a state space measure indicating whether the extensions are promising (Section 5.3.2). Thereby, the algorithm investigates the feasibility of the state space in the considered region and also checks in advance whether there can exist feasible solutions at all before conducting the optimization model (cf. Section 5.2). In case, a region with feasible solutions is found, the resulting state space is examined by our optimization model for multi user context-aware re-selection (cf. Section 5.2). Finally, the algorithm analyses whether several alternative solutions can be selected (Section 5.3.3). The following Table 3 introduces the concepts of our algorithm at a glance, which are discussed in the Sections 5.3.1 to 5.3.3 in detail.

Table 3. Algorithm for multi user context-aware re-selection

Function: MultiUserContextAwareServiceReselection()
Input: Process p, CandidateServices cs[][], World States ws[][], BlacklistedItems bl[], State Space Measures ssm[][], Events ev[], Users a[], State Space sp[][]
Output: A feasible solution regarding the underlying selection problem

```
BEGIN
  feasibilityInformation[] <- doFeasibilityChecks (p, ws[][], a[])
  region <- setInitialReselectionArea (p, ev[], feasibilityInformation[], a[])
  DO
     fixedWS[] <- fixWorldStates (region, ws[][], a[])
     sp[][] <- extendCurrentStateSpace (region, ws[][], fixedWS[], a[])
     sp[][] <- reduceCurrentStateSpace (region, ws[][], fixedWS[], cs[][], bl[], a[])
     reselectionSuccess <- reselect (p, region, sp[][], bl[], a[])
     IF (reselectionSuccess==false)
       ssm[][] <- getIndicatorForRegionalExpansion (p, region, a[])
       region <- expandReselectionArea (region, p, ev[], ssm[][], a[])
     ELSE
       WHILE (userExpectsAlternative==true & reselectionSuccessAlternative==true)
            reselectionSuccessAlternative <- reselect (p, region, sp[][], bl[], a[])
            userExpectsAlternative <- getUserPreferencesForAlternatives(a[])
       DO
    END IF
  WHILE (reselectionSuccess==false)
END
```

5.3.1 Algorithm for re-selection based on a single region

Focus task-based definition of regions

In a first step, the overall process is divided into regions (cf. Fig. 4) for each individual user based on the users' state spaces from the planning time, which aims to limit the search space and to preserve a large part of the process from change. To illustrate our region-based approach, it is assumed w. l. o. g. in the following that User 1 is located in Region 1 of the process and the disruptive event occurs in Region 3 (e.g., closed restaurant). Therefore, a re-selection within only Region 3 is focused on:

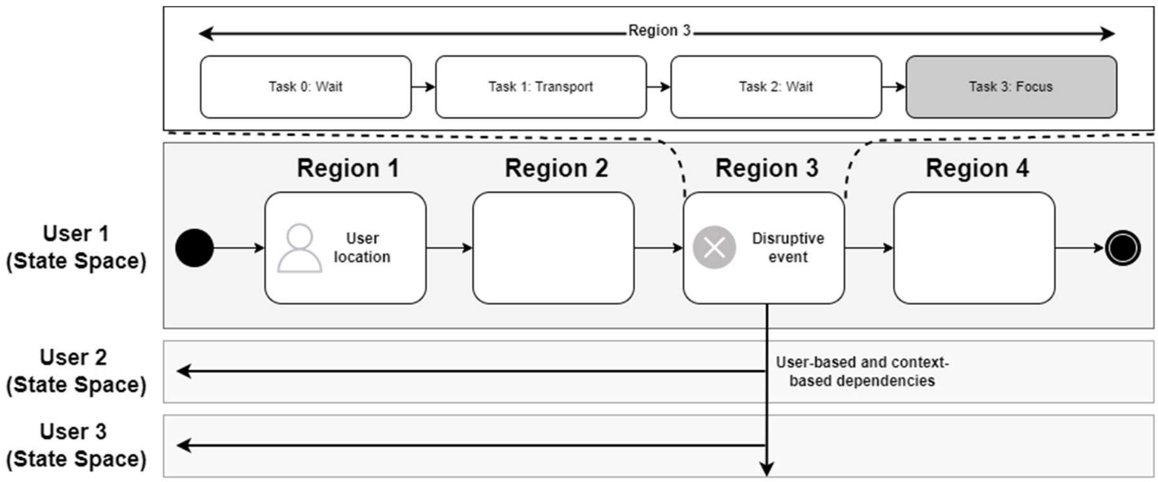

Fig. 4 Region-based approach

Each of these regions within the process refers to exactly one focus task (e.g., a restaurant) and further consists of particular tasks of type Wait, Transport and again Wait (cf. Fig. 4). The delimitation of a region with exactly these four tasks is mandatory in our approach as it can ensure spatial-temporal coordination and it further allows for a small (atomic) size of the region to keep the re-selection effort low and the robustness high: More precisely, based on the state space, user-based dependencies and context-based dependencies such as time of day and location between multiple users (cf. Section 2.1, cf. also Fig. 4) have to be considered within the defined region in order to select the most suitable services, while dealing with individual context information of users and their coordination.

Specifically, in multi user context-aware processes, *waiting tasks* (i.e., Task 0 and Task 2, e.g., waiting time for conducting the transport or focus task) are an important concept to represent a temporal-based coordination between multiple users (e.g., cf. Mayer 2017), while a *transport task* (i.e., Task 1, e.g., User 1 is transported to another restaurant when an event occurs) support and enable spatial coordination between multiple users.

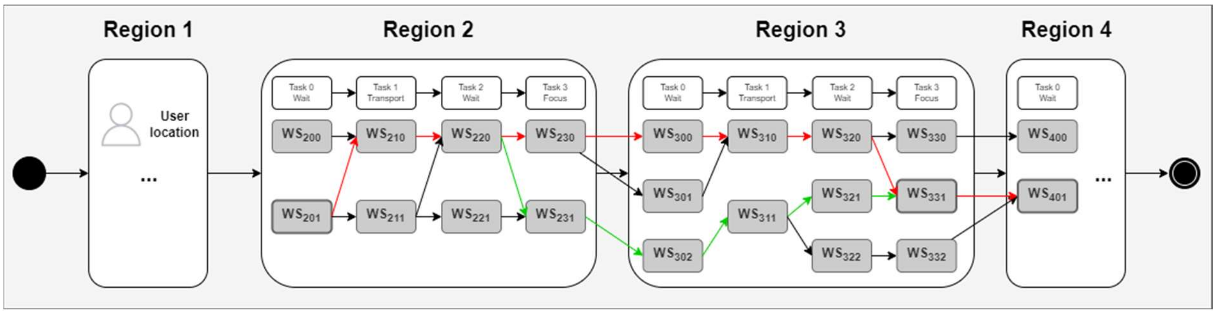

Fig. 5 Representation of state transitions within simplified state space

In order to preserve the context of a user in terms of time and location, the boundaries of the region to be re-selected must be set. Thus, we are the first approach which defines these boundaries in a stateful process based on *world states* as they represent context information in regard to time and location. Therefore, world states can be used to determine effectively the beginning and end of the region in which the disruptive event occurs and a re-selection must be performed. In this regard, Fig. 5 shows exemplary a simplified state space in which the state transitions between world states are represented (for illustration purpose, without their associated candidate services and world-state-candidate-service combinations). The red path constitutes the optimal service composition selected at planning time whereas the green path shows the optimal service composition after re-selection. Given this, the boundaries of Region 3, for instance, for the planned selected service composition can be determined. More precisely, world state ws_{300} is set as the beginning of Region 3 because this world state is automatically determined by the previous part of the service composition (i.e., by world state ws_{230} and its related candidate service). Furthermore, world state ws_{401} is set in the succeeding service composition (i.e., Region 4) as this world state needs to be reached again after the re-selection in Region 3 in order to maintain the state transitions between these neighboring regions.

Event-based re-structuring of the regions state space

When a disruptive event occurs, the context within the state space may change for one or more users. As a result, on the one hand, states which reflect the changed context of these users are potentially not present in the state space and, on the other hand, already existing states in the state space may no longer be feasible. Building a completely new state space is very time-consuming and not promising for a runtime approach. Therefore, an elaborated state re-structuring is proposed at runtime, which includes only feasible context information in the state space. In particular, the state space re-structuring consists of three steps: *State space extension*, *State space reduction* and *Feasibility checks* all aiming to obtain a feasible state space:

For *state space extension* a world state is injected in the state space, which reflects the current user context (time and location) and all resulting feasible world states and the associated paths are re-structured forward in the process. Especially, this allows to consider additional feasible states, which were not feasible at planning time. In order to support high performance and robustness, the extension of the states only takes place within the considered region and not for the entire process (i.e., Task 0 to Task 3 in Region 3, cf. Fig. 4). Furthermore, *state space reduction* removes for each individual user all world states and candidate services from the state space that are not feasible (while considering user-based and context-based dependencies within the region, cf. Fig. 4). In particular, only world states and candidate services are considered, which can be re-selected with respect to the fixed beginning and end world state of a region (e.g., Region 3, cf. Fig. 5). This results in a sub state space (i.e., Task 0: $\{ws_{300}\}$, Task 1: $\{ws_{310}\}$, Task 2: $\{ws_{320}\}$, Task 3: $\{ws_{331}\}$) in which large parts of the state space can be excluded from the re-selection process. Finally, we extend the algorithm with *feasibility checks* in order to check the feasibility of the sub state space before a re-selection takes place. For a region, context information is analyzed to check whether feasible solutions can exist in the sub state space at all before executing our optimization model (cf. Section 5.2). In particular, feasibility checks can recognize when no feasible solution can be selected within a regions state space and a region-based expansion is necessary anyway. In this regard, we focused on feasibility checks related to time (e.g., there is no feasible candidate service in the region referring to opening times and the users' context) and location (e.g., there is no feasible transport in the regions state space referring to transport duration). In case that the sub state space contains feasible solutions, the optimization model introduced in Section 5.2 can be performed in order to find a new feasible and near-optimal solution based on this re-structured sub state space of the region.

5.3.2 Algorithm for re-selection based on region-based expansions

In case that within a single region (e.g., Region 3) no feasible solution can be re-selected, further regions have to be considered for re-selection. Precisely, if the initially selected service composition for Region 3 (i.e., marked red path in Region 3, cf. Fig. 5) is potentially no longer feasible (e.g., the candidate service for world state ws_{310} is no longer available and no alternative candidate service exists) other paths may be feasible by considering further regions. Thus, alternative world states have to be re-selected within the process (e.g., marked green path world state ws_{231} instead of world state ws_{230}, cf. Fig. 5). These expansions of the state space have to be done carefully to address computational complexity. To achieve this, we describe two extensions for the presented elaborated region-based algorithm: 1) Expansion of regions based on a state space measure (indicator), and 2) Examination of the feasibility of the state space.

Region-based expansion based on state space measure

The expansion with a neighboring region (i.e., Region 2 or Region 4, cf. Fig. 5) has a great impact on the performance and solution quality of the re-selection and must therefore be performed in an elaborate manner. In this regard, we choose the neighboring region based on the state space measure *number of world states of all users per region.* This can be reasoned, as adding the neighboring region with the largest number of world states (summed over all users) results in a larger search space and is therefore more promising for finding a feasible solution. Although the time for re-selection of the larger search space results in a higher computation time (i.e., due to the higher number of world states), in general fewer expansions to further regions are required. This leads to a tradeoff between the number of region-based expansions and the size of the considered state space until a feasible solution can be found. However, searching within a larger initial search space is – from the perspective of a heuristic – faster than performing several stepwise expansions each including re-structuring of the state space. Further, we use the measure *number of world states* from the planning state space (as an indicator) because determining the number of world states of the re-structured state space at runtime is very time-consuming. If after expansion no feasible solution can be re-selected in the joined regions (e.g., joining of Regions 2 and 3), a further region is added (e.g., Region 4) at a time (based on the above-mentioned state space measure) until a feasible solution is found by the optimization model or the end of the process is reached (then the heuristic will fail because no feasible solution can be found).

Examining the feasibility of the state space

As stated in the last sub section, time-consuming expansions must be reduced in order to support high performance. Therefore, our algorithm is able to manage the size and feasibility of the regions state space (cf. Section 5.3.1) in order to select a near-optimal solution with the optimization model presented in Section 5.2. In that respect, feasibility checks provide a sophisticated procedure to determine a feasible state space. In particular, our feasibility check can not only determine that no feasible solution can be re-selected in a single region, but can directly determine the size of a larger region in which the re-selection of a feasible solution is promising (i.e., e.g., in Region 3 there is no feasible solution, then an expansion and re-selection to Regions 2 ∪ 3 takes place directly). In this regard, the restructuring of the state space (i.e., extension and reduction) can be applied dynamically to regions of any size. If nevertheless no feasible solution can be re-selected (e.g., due to context interdependencies between multiple users) in the joined regions, a further region must be dynamically added step-by-step until a feasible solution is found (cf. Region-based expansion based on state space measure).

5.3.3 Algorithm for selecting alternative solutions

The re-selected solution may not meet the users' expectations due to, for example, imprecise or incomplete context information or changing user preferences. Therefore, it is important that the algorithm can select several alternative and feasible solutions in near-time with a high solution quality. An alternative solution is given if at least one candidate service from the re-selected solution is modified in the focus task within the considered region even if the rest of the solution still matches the previously selected solution. In this regard, we blacklist this candidate service (or potentially more candidate services based on user preferences) and re-select the alternative solution within the considered region with the (next) higher utility. All steps and information of the region-based approach (e.g., cf. Focus task-based definition of regions, Event-based re-structuring of the regions state space) can be reused from the previous re-selection resulting in significant performance advantages. This approach enables the selection of several alternative solutions while maintaining high performance and solution quality.

The following Table 4 summarizes the contribution and shows which features of the solution are supported by the presented region-based multi user context-aware service system:

Table 4. Summary of the contribution to the presented algorithm

Concept	*Feature*				*Contribution*
	Performance	*Solution quality*	*Solution robustness*	*Alternative solutions*	
Re-selection based on a single region (Section 5.3.1)	*X*	*X*	*X*		- Determination of the user context in terms of time and location by fixing the boundaries of a region with world states - Dynamic state space extension to address changing user context during re-selection - Dynamic state space reduction to only feasible states, while maintaining user-based and context-based dependencies to neighboring regions
Re-selection based on region-based expansions (Section 5.3.2)	*X*	*X*	*X*		- Elaborated extension of the search space by using a state space measure in order to reduce the number of region-based expansions - Examining the feasibility of the state space in order to reduce the number of time-consuming re-selection steps
Selection of additional solutions (Section 5.3.3)	*X*	*X*		*X*	- Providing proactively several additional solutions at runtime - Re-using already performed re-selection steps (e.g., state space extension or state space reduction) from the initial solution determined at runtime

6. Evaluation

In this section, we evaluate our heuristic technique regarding the features performance, solution quality, solution robustness and alternative solutions, as discussed in the methodical foundations. The design of our evaluation follows the compositional styles demonstration as well as simulation- and metric-based benchmarking of artefacts (cf. Prat et al. 2015, cf. Section 3). Thus, the evaluation is conducted by means of a *simulation experiment* based on a real-world data set in the tourism domain (cf. Section 4). Further, in order to put the quality of the proposed heuristic (in the following abbreviated with *MUCARS (multi user context-aware re-selection)*) into perspective with respect to the above features, a comparison to existing multi user context-aware service systems is performed.

6.1 Setup and Data Preparation

When aiming at a comparison to extant approaches, it is necessary that these works can cope with multiple participating users, context-awareness and disruptive events. As discussed in the Section Related Work, there are no approaches that address these three aspects. The only technique that considers user-based as well as context-based dependencies and represents a heuristic explicitly focusing on performance as a key criterion for evaluation is the one presented by Bortlik et al. 2018 (in the following abbreviated with *MUCAHA (multi user context-aware heuristic approach)*). For addressing disruptive events, we run the *MUCAHA* approach at every event to the relevant part of the process (i.e., start with the task the user conducts until the end of the process) to be able to handle these events. The second work by Heinrich and Mayer (2018) also dealing with multiple participating users and context-aware service selection presents an approach that provides an optimal (exact) solution at planning time (in the following abbreviated with *MUCAOS (multi user context-aware optimal solution)*). This means, in a runtime setting showing (very) high computational complexity, a broad comparison regarding runtime relevant features like performance, solution robustness and alternative solutions is not meaningful and above all not possible (i.e., finding an optimal solution in appropriate time is not realistic since the underlying decision problem is NP-hard). Thus, we can use *MUCAOS* for comparison purposes only in individual cases (e.g., selected simple settings to determine the feature solution quality).

Table 5. Data preparation for the initial basic setting

Data Type	***Parameter***	***Task***	***Data preparation***
Real-world data	Price (non-context-aware)	Focus Task	real price
	Rating (non-context-aware)	Focus Task	real rating
	Location Area (non-context-aware)	Focus Task	real location area
	Parking distance (non-context-aware)	Focus Task	real parking distance
Particular definitions of data	Duration (non-context-aware)	Focus Task	15 - 120 min in steps of 15 respective 30 min depending on the respective candidate service type
		Transport Task	15 - 60 min in steps of 15 min
	Price (non-context-aware)	Transport Task	walk: 0 €; bike: 1 €; car: 4 €
	Waiting Time (non-context-aware)	Waiting Task	0 - 60 min in steps of 15 min
	Users' Constraints	-	(0.99 * max. possible aggregated non-functional property value) for all non-context-aware and context-aware attributes for each user
Data generated in each simulation run	Users' Preferences	-	randomly generated for all non-context-aware and context-aware attributes for each user
	Users' Initial Contexts	-	start point is a randomly selected time between 11 – 12 a. m. (in steps of 15 min); randomly selected GPS position in the area of Melbourne, Australia, for each user (for the start and the end point)
	Event Occurrence	Focus Task	random selection of the focus task in the future part of the process according to the user location

For our evaluation we draw on a real-world data set in the tourism domain (cf. Section 4). More precisely, in the initially introduced *basic setting*, three users conduct a day trip in the city of Melbourne, which comprises three focus tasks (e.g., having breakfast, having lunch, visiting museum; in an extended setting further tasks such as go shopping are considered). Therefore, we have extracted real-world data from a very popular web portal, which provide information about local businesses. As a result, we obtained candidate services for the tasks *Breakfast*, *Shopping*, *Restaurant*, *Arts*, *Café*, *Sight*, *Beauty* and *Bar* and their corresponding non-functional properties. In particular, for these candidate services, the real data on prices, ratings, locations and distances to parking were extracted. In order to perform the simulation experiment, few further data was defined in addition to real-world data: First, based on the non-context-aware and context-aware attributes and the corresponding non-function properties values, the *constraints of the users* were defined. Moreover, the durations of the selected candidate services were determined depending on the candidate service type (i.e., 15 – 120 minutes in steps of 15 respective 30 minutes). To enable the *transport* of the users between tasks, the candidate services *Walk*, *Bike* and *Car* represent the transport options for each transport task. Therefore, corresponding durations (i.e., 15 – 60 minutes in steps of 15 minutes) and prices (i.e., walk: 0 €, bike: 1 €, car: 4 €) were generated for the different transportation types. Furthermore, to bridge possible *waiting times* of the users, five candidate services for waiting times (i.e., 0 – 60 minutes in steps of 15 minutes) were modeled in the waiting tasks. In addition to these careful definitions, *user-specific data* were determined randomly in each simulation run based on existing real-world data, since no real users are present. In this regard, each user has his own context within the process. Therefore, for each simulation run we randomly create the start and end context (e.g., time and location in the city of Melbourne) for each individual user as well as the values for their preferences (including favorite scores for different candidate service types and transports) for all non-context-aware and context-aware attributes within defined minimum and maximum values. Consequently, we use the non-context aware attributes price, rating, duration, waiting time and favorite score as well as the context-aware attributes location area and parking distance for the description of candidate services in this process. Furthermore, the occurrences of the disruptive events were also randomly generated. To evaluate arising events during the city day trip at runtime (e.g., closed restaurant), we simulate the resulting service failures in the model by randomly determining a task in which the real-world event occurs (i.e., within the set of tasks in the process that the user has not yet conducted). Therefore, a task is randomly selected whose predetermined candidate service from the service composition is no longer available. This procedure ensures that events occur in alternating tasks and thus resulting re-selections are performed on different parts and sizes of the process. In summary, the *basic setting* contains three focus tasks, three users, five events and 50 randomly selected real-world objects (candidate services) per focus task in the area of Melbourne, Australia. Table 5 summarizes the data for the initial basic setting.

Based on the described basic setting, we define four *extended settings*. More precisely, in each extended setting we stepwise increase exactly one parameter (i.e., the number of focus tasks, candidate services, user or events), while all other parameters of the basic setting remain the same (i.e., ceteris paribus). This allows us to analyze the effects of the modified parameters on our evaluation features (e.g., solution quality). The intervals and steps resulting for each extended setting are shown in the following table (cf. Table 6).

Table 6. Extended settings

Setting	***Extension***	***Intervals and steps of the parameters***
I	# of candidate services per focus task	10 to 120 (in steps of 10) 140 to 200 (in steps of 20)
II	# of users	2 to 8 (in steps of 1)
III	# of focus tasks	1 to 10 (in steps of 1)
IV	# of events	1 to 20 (in steps of 1)

In our simulation experiment, we examine the evaluation features *performance*, *solution quality*, *solution robustness* and *alternative solutions*. Each of the interval steps of the extended setting is simulated twenty times and, on this basis, we determine the average results for each evaluation criterion. To ensure a correct implementation of our algorithm, we conducted intensive testing of the source code, namely manual code reviews by persons other than the programmers, unit tests, runs with extreme values and feasibility checks. After this, each simulation run was performed on an Intel Xeon E5-2650 v4 processor, 512 GB RAM, Debian 9, Java 1.8, and the mathematical solver SCIP Optimization Suite 7.0.1.

6.2 Results

In the following, the features performance, solution quality, solution robustness and alternative solutions of the *MUCARS* are set into perspective to the *MUCAHA* heuristic and – if possible – to the *MUCAOS* exact approach. Moreover, when analyzing the results regarding solution robustness, a general baseline is needed for a comparison between the approaches. In this regard, we use the initial results of the *MUCAHA* approach at planning time without any runtime restrictions and without any disruptive events (in the following abbreviated with *BASE_MUCAHA*).

6.2.1 Performance

In this section, we analyze the performance by setting the computation time of *MUCARS* into perspective to *MUCAHA*. Thus, we define the feature *Performance* as follows:

$$Performance = \frac{ComputationTime_{MUCARS}}{ComputationTime_{MUCAHA}}$$

We choose to assess the performance relatively to the computation time of *MUCAHA* since this allows a comparison independent from the hardware used for the simulation experiment. With our runtime-optimized approach, we expected less increase in computation time with growing problem size, which is supported by our results in almost all extended settings.

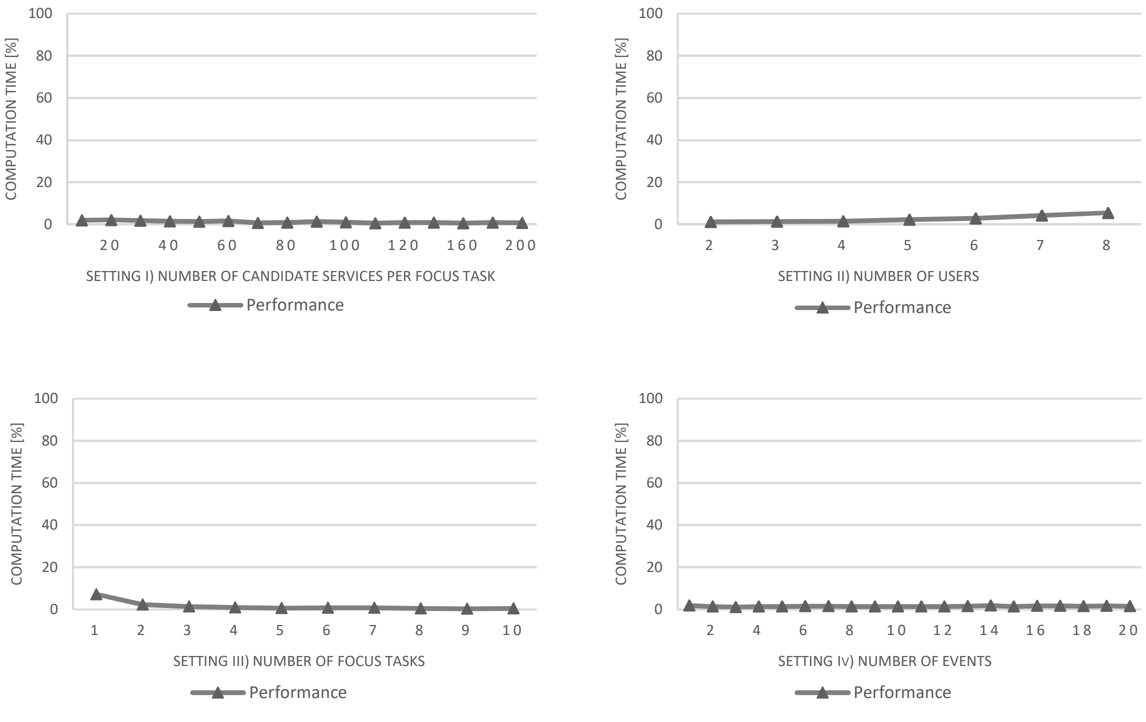

Fig. 6 Evaluation results for the criterion performance

The *MUCARS* approach can select a new solution for the user in only 1.6% on average of the computation time of the *MUCAHA* across all settings and is therefore on average 60 times faster than *MUCAHA*. This proportion remains relatively constant across all four settings (cf. Fig. 6). In particular, for Setting I with ten candidate services the *MUCARS* needs 1.9% and for 200 candidate services 0.7% of the computation time compared to the *MUCAHA* (across all values of the x-axis, the average is 1.2% (11 sec. in average by absolute numbers)). Setting II requires 1.2% for two users and 5.5% for eight users compared to the *MUCAHA* (across all expressions of the x-axis, the average is 2.7%). Furthermore, Setting III shows that the *MUCARS* needs 7.2% for one focus task and 0.4% for ten focus tasks (across all expressions of the x-axis, the average is 1.5%). Finally, the *MUCARS* in Setting IV requires 1.8% for considering one event and 1.5% for considering 20 events of the *MUCAHA* runtime (across all expressions of the x-axis, the average is 1.5%). The analysis in Setting III shows that the advantages of the *MUCARS* are more significant for large processes (i.e., 7.2% performance for small processes vs. 0.4% performance for large processes) as the *MUCARS* focuses on smaller parts of the process during the re-selection (i.e., region-based approach), which is more advantageous for larger processes. When further analyzing the performance, it is noticeable that in Setting II the only case occurs in which the increase in computation time is no longer proportional for the *MUCARS* starting from six users and the runtime of the *MUCARS* increases more compared to the *MUCAHA*. On the one hand, this is due to the over-proportional increase in common world-state-candidate-services (i.e., concept to ensure the simultaneous use of the same world state and candidate service by multiple users) with increasing number of users. On the other hand, an increasing number of users leads to a smaller solution space due to preferences and restrictions among all users and thus more region expansions are necessary. Hence, both reasons influence the computation time more strongly with a high number of users than with few users leading to an over-proportional increase of the computation time. Nevertheless, the *MUCARS* is still 20 times faster with eight users than the *MUCAHA*.

Finally, the *MUCARS* approach shows a proportional increase in computation time in almost all extended settings despite increasing user-based and context-based dependencies.

6.2.2 Solution Quality

The feature *solution quality* is analyzed by comparing the Utility U (cf. Section 5.2) of the optimal solution provided by the *MUCAOS* approach with the corresponding Utility U of the approaches *MUCAHA* respectively *MUCARS* (i.e., near-optimal solution) as a percentage. Therefore, the solution quality for each multi user context-aware service selection can be defined as:

$$Solution\ quality_{MUCARS} = \frac{U_{MUCARS}}{U_{MUCAOS}} \qquad Solution\ quality_{MUCAHA} = \frac{U_{MUCAHA}}{U_{MUCAOS}}$$

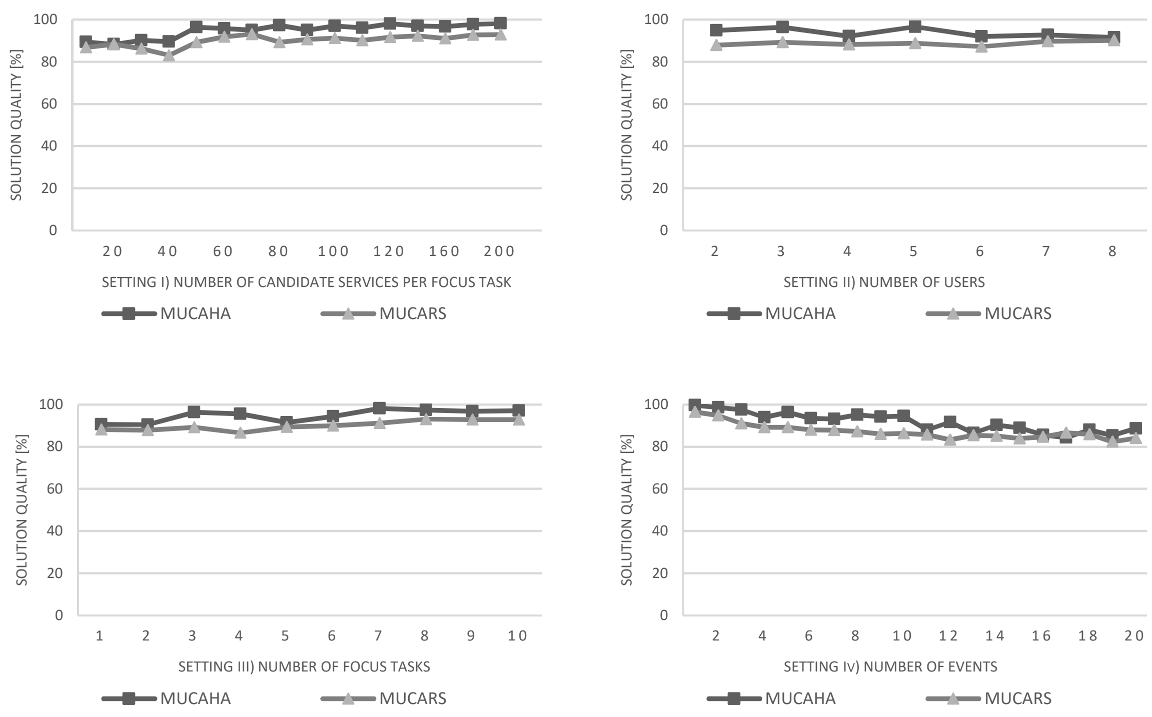

Fig. 7 Evaluation results for the criterion solution quality

The *MUCAHA* reaches an average value of 93.5% and the *MUCARS* 88.8% of the solution quality over all settings in comparison to the optimal solution from *MUCAOS* at planning time. These results show that our approach generally achieves a high solution quality across all settings, although disruptive events are considered at runtime. Therefore, our results mark the lower bound of the actual possible solution quality, which would be even higher compared to runtime results of an exact approach (which is not realistic due to runtime complexity). In more detail, the slight increase in solution quality with an increasing number of candidate services per focus task (i.e., Setting I) can be reasoned by the fact that if a disruptive event occurs, a service composition with a higher solution quality can be re-selected if the number of suitable alternatives (=candidate services) is large. On the other hand, increasing the number of events (i.e., Setting IV) reduces the solution quality of the re-selected service composition, because the search space and thus the amount of suitable alternative candidate services decreases as the number of events increases (compared to the predetermined solution of *MUCAOS*).

In addition, the *MUCARS* approach was able to select a feasible service composition in 100% of all cases. The *MUCAHA* approach could not find a feasible solution in two cases with increasing number of events (cf. Setting IV) and therefore found a feasible solution in 99.8% of all cases. In these two cases, a failed candidate service (i.e., duration: 30 minutes) is replaceable in the same region only by candidate services, which have a longer duration (i.e., duration: 60 minutes). Because an extension of the duration is not feasible due to a defined end time of the process, no feasible solution can be selected by the *MUCAHA*.

6.2.3 Solution Robustness

Furthermore, we analyze the solution robustness across all settings indicating whether there are fundamental *changes* to the service composition determined by the *BASE_MUCAHA* at planning time. A change is any replacement of an original candidate service in one of the focus tasks of any user in the process (excepting the failed candidate service) resulting in a reduction of the solution robustness. The solution robustness for exactly one re-selection can be determined as follows:

$$Solution\ robustness = 1 - \left(\frac{\#\ ReselectedFT}{MaximumFT}\right)$$

$\#\ ReselectedFT$ describes the number of focus tasks whose selected candidate service from the original service composition has changed due to a re-selection at runtime across all users. $MaximumFT$ describes the maximum number of focus tasks that can change during a re-selection across all users and processes.

The *MUCARS* approach achieves a continuously high value for solution robustness with an average solution robustness of 92% across all settings (in comparison, the *MUCAHA* achieves 73%). In particular, the *MUCARS* achieves an average solution robustness of 94% in Setting I (*MUCAHA* 74%), 90% in Setting II (*MUCAHA* 72%), 94% in Setting III[5] (*MUCAHA* 77%) and 91% in Setting IV (*MUCAHA* 70%). Thus, the advantages of the *MUCARS* over the *MUCAHA* approach are evident in all extended settings.

[5] Robustness with only one focus task is obviously not determinable.

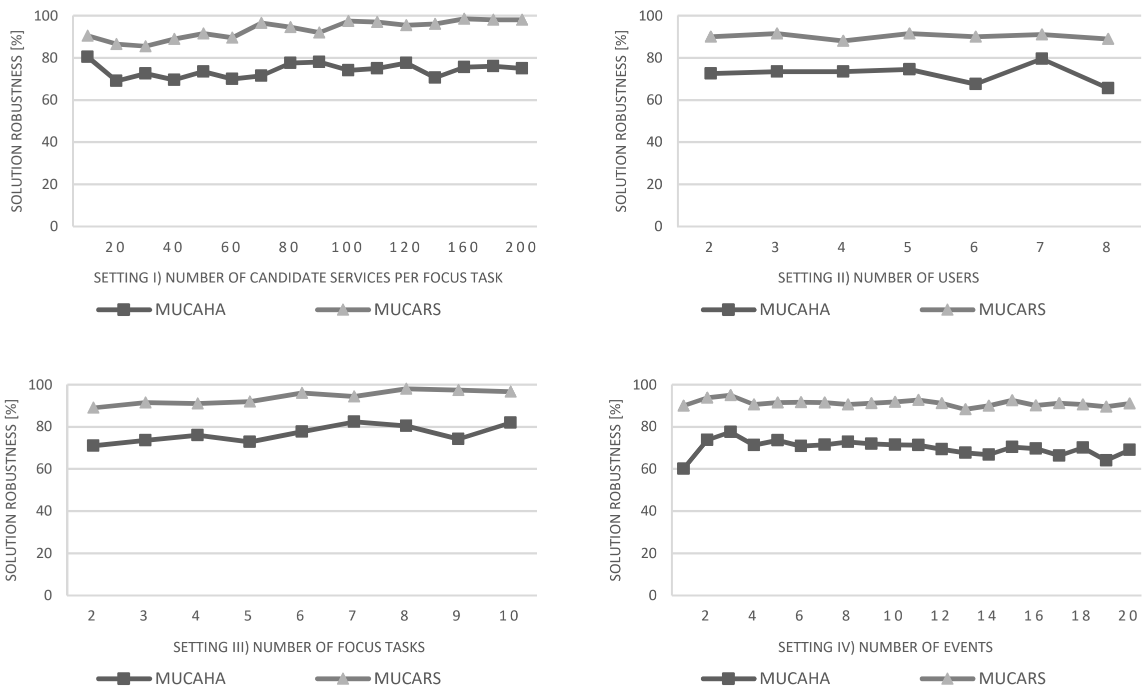

Fig. 8 Evaluation results for the criterion solution robustness

6.2.4 Alternative Solutions

To evaluate our approach with regard to alternative solutions, we selected five additional alternative solutions (cf. Section 5.3.3) for the *MUCARS* approach for each problem setting towards an increasing number of events to determine both the average *solution quality* and the average *performance* (note that *MUCAHA* does not provide the functionality to determine alternative solutions).

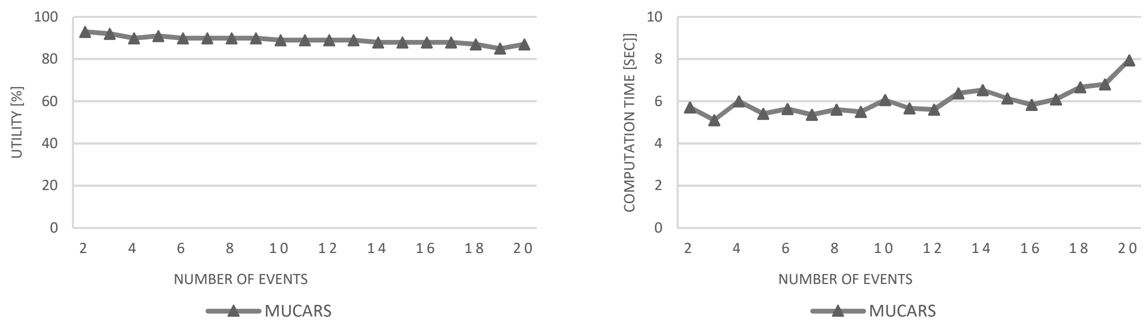

Fig. 9 Evaluation results for the criterion alternative solutions

Considering the highly relevant setting of an increasing number of events at runtime (cf. Setting IV), the *MUCARS* approach provides alternative solutions with an average solution quality of 89% compared to the originally selected service composition at runtime. Furthermore, an alternative solution can be selected in 100% of all settings and can be presented to the user in an average of 6.1 seconds per event. This further shows the high effectiveness and stability of the *MUCARS* approach.

6.3 Discussion of Results and Implications

We propose a heuristic technique that is able to consider multiple users, context-awareness and in particular disruptive events, while maintaining solutions at runtime with high *performance*, *solution quality*, *solution robustness* and also provide *alternative solutions*. Analyzing the evaluation results, *MUCARS* offers significant advantages compared to the competing algorithm *MUCAHA*. In particular, *MUCARS* can select a solution on average 60 times faster than *MUCAHA* with an overall solution robustness of 92%, while maintaining a high solution quality comparable to *MUCAHA*. Finally, the evaluation reveals that up to five *alternative solutions* can be selected within a few seconds with a high solution quality. Since there is no approach that fulfills all these features at runtime, these contributions show the novelty and efficacy of the approach.

The selection of a fast and robust solution with a nevertheless high solution quality can be obtained by dividing the process into regions (i.e., region-based approach). In particular, we promote the feature solution robustness by clearly delimiting the respective region boundaries with the help of *world states,* which is different from existing approaches in literature where regions are delimited by services (e.g., cf. Lin et al. 2010; Zhai et al. 2009). To cope with the problem of spatial-temporal coordination and the corresponding mapping of the context, the consideration of world states within a region is indispensable in context-aware service systems.

Furthermore, we have significantly extended the concept of regional expansion from the literature (e.g., cf. Gao et al. 2018; Lin et al. 2010; Zhai et al. 2009) by introducing *feasibility checks* and a *state space measure*. The feasibility checks enable to directly examine regions within the state space in which feasible solutions exist. Thus, unnecessary and time-consuming expansions as well as re-selections can be avoided (in contrast, the execution of the feasibility check in the basic setup only need 5 milliseconds on average). This becomes also evident in a more detailed analysis of our results, which shows that through the feasibility checks across all settings 1,711 region-based expansions can be avoided, since in each of the 1,711 regions a feasible re-selection can be correctly ruled out. This accelerates the selection process and thus leads to a higher performance. In case all feasible solutions can be ruled out in a region due to user-based and context-based dependencies, we introduce a state space measure for the region-based expansion. This measure enables a careful selection of the neighboring region by analyzing the available context information (i.e., mapped by world states) across all users. A deeper analysis shows that better solutions can be achieved when the region with the larger number of world states (i.e., larger search space) is selected thus supporting the feature solution quality. Additionally, fewer expansions are required. In particular, we examined all runs of the basic setting for which an expansion was required. In 100% of these cases, a solution was found directly after the first expansion when the region with the larger number of world states was used. In contrast, expanding the region with the fewer number of world states found a solution in only 25% of these cases, which would lead to further time-intensive expansions. The evaluation results show that in a total of 97% of all our evaluation settings (cf. Table 6) a solution could be selected after the feasible region has been directly examined by *feasibility checks* and a maximum of only one additional regional expansion has been performed with the help of our *state space measure* (cf. Section 5.3.2), which demonstrates the high relevance of the regions selection.

The evaluation results are also favored by the proposed re-structuring of the regions state space. In the literature, the use of states is a common concept to represent, for example, context interdependencies (Heinrich and Schön 2015). However, the creation of an entire and feasible state space requires a lot of computation time. Our approach expands the concept of states by dynamic and targeted *extension* of the state space based on runtime conditions, thus eliminating the need to create an entire new state space. In this respect, an analysis of our basic setting shows that the *BASE_MUCAHA* needs on average 210 seconds to build an entire state space and the *MUCARS* needs on average only 13.6 seconds for the dynamic extension per event, and thus the computation time can be reduced by 93.5%. On the other hand, we do not perform the selection on the entire state space like other approaches (Bortlik et al. 2018; Heinrich and Mayer 2018) as this is very time-consuming. Therefore, the state space is narrowed down to the feasible states (based on the valid context information) before the selection is performed. Therefore, a selection of the basic setting (cf. Section 6.1) is not performed on over 156,000 states but only on about 4,000 states, which further improves the performance with simultaneously high solution quality. Further, in contrast to existing service selection approaches from the literature, our approach allows the determination of alternative solutions. By carefully reusing already performed tasks (e.g., Focus task-based definition of regions) from the initial solution selected at runtime, the computation time can be significantly reduced while maintaining a high solution quality. Analyzing our evaluation results in detail shows, that the originally defined region of the initial solution selected at runtime could be reused in 86% for each problem setting towards an increasing number of events (cf. Section

6.2.4) to find a feasible alternative solution. Thus, a further time-intensive expansion to further regions can be dispensed with in these cases.

Our approach also has implications for science and practice. Starting with *implications for science*, service systems (cf. Maglio and Spohrer 2008) with their characteristics and their complexity regarding user-based dependencies, context-based dependencies and event-based disruptions lead to novel, real decision problems. These problems should not only be discussed conceptually in the scientific literature (Fakhfakh et al. 2020; Fisk et al. 2011), but for which concrete solution techniques should also be developed. In this regard, collaboration and contextualization are already an important part of service-dominant design which forms the basis for modern service systems (Alter 2012; Böhmann et al. 2014; Faieq et al. 2021; Maleki et al. 2018; Yuan and Hsu 2017). In particular, contextualization comprises information that characterizes the actual state of an environment (e.g., cf. Romero et al. 2020) and thus affects the design of service systems due to resulting uncertainties (Alter 2017). These uncertainties are already considered in the design of service systems. Therefore, Alter (2017) describes a (context-aware) service system through proposed axioms as a service system that is affected by direct or indirect interactions with the environment in which it operates (i.e., Axiom 4) and further states that the success of a service system depends on responding appropriately to the diversity of situations that the service system will encounter (i.e., Axiom 18). For this reason, *adaptive context-aware service systems* were discussed in the literature (e.g., cf. Bucchiarone et al. 2012; Faieq et al. 2021; Fkaier et al. 2017). One of the main task of these systems is *managing adaptations based on the existing context information* (Fkaier et al. 2017). Managing adaptations based on the existing context information has high performance requirements (i.e., real-time, cf. Fkaier et al. 2017). Existing adaptive context-aware service systems from the literature primarily focus on the requirements of the service provider. In particular, the flexible response of service providers to changes and the corresponding adaptation of business processes (Faieq et al. 2021) in compliance with the service goals pre-established in the Service Level Agreements contracts (Hidri et al. 2019) represent the key role in the design of adaptive context-aware service systems. In our research, we design the system mainly on the user's perspectives (e.g., companies or users who use the services, i.e., service consumers) whereby mechanisms were developed which come close to real-time support. Thus, we substantively extend existing meta-models from the literature (e.g., Hidri et al. 2019) in the sense that user requirements (e.g., high performance) are detailed and included directly in the design of the adapted process. Furthermore, an essential part of managing adaptations is also an *analysis component*. Thus, existing designs of adaptive context-aware service systems firmly anchor this component directly in the meta-model (e.g., cf. Boudaa et al. 2017). Thereby rules are defined, which check whether an adaptation is necessary when events occur. In our approach, we methodically extend and realize this analysis component with *feasibility checks* and a *state space measure* that control the adaptation mechanism at runtime. This procedure opens the research field to develop, for example, further measures to address the existing challenges of user-based dependencies, context-based dependencies and event-based disruptions.

For *practitioners*, our heuristic approach also provides some important benefits in terms of supporting multi user context-aware processes (i.e., maintaining the contextual environment and the collaboration of multiple users in terms of spatial-temporal coordination) where runtime features are elementary. Thereby, the approach can be applied to multi user context-aware processes in different domains relying on the re-selection of services (i.e., service systems) and can have a relevant impact in practice. We will exemplarily discuss this for the tourism domain. Existing mobile applications that support processes such as a city day trip are characterized by certain possibilities for user individualization. For instance, the app *Culture Trip* enables guided tours, the creation of individual plans or the search for sights in the vicinity of the current location. However, in these *tourist apps* or *platforms* (the latter often takes on the role of service integrators, cf. Heinrich et al. 2011) it is currently not possible to handle disruptive events anyway in regard to *businesses* (e.g., restaurants or museums, i.e., service providers) and *users* (i.e., service consumers). The approach proposed in this paper, allows to consider such real-world events combined with a near-time re-selection, i.e., it is possible to continue the planned tourist activity – and potentially also any other planned activities – with a high quality without interruption. By implementing the proposed approach as extension of existing mobile applications in tourism, users can react on disruptive events by re-selecting an alternative service composition, which is robust regarding the initially planned service composition. This can be done by directly using publicly available information on disruptive events (e.g., weather, traffic jam) and proactively inform all affected users and preventively offer alternative solutions. Furthermore, tourist platforms can use the presented approach not only for disruptive events, but also to immediately adapt an existing feasible solution in general. Thus, a re-selection is not only needed to address an external, given disruptive event (e.g., sudden change in weather) but provides the opportunity for proactive adaptation ("initiated events") of the current feasible solution to further improve the user experience. In particular, users can be enabled to actively initiate changes shortly before the start of the tourist activity or during the tourist activity and share them with other participants. Thus, flexible alternatives can be proposed and perceived between users (e.g., based on current conditions, preferences etc.), which greatly increases social interaction and enables the possibility of a spontaneous realization or adaptation of a tourist activity. Moreover, it will be possible for tourist platforms to draw the user's attention based on real-time information ad hoc to service candidates that might appeal to the individual user (or multiple users) on the basis of the preferences or context information (e.g., time, GPS position) and to immediately suggest an alternative solution (i.e., service recommendation). Thus, user experience can be made even more individual. Finally, businesses such as restaurants or museums can also benefit from our approach by providing current data of their business to the tourist platform via an interface. For example, offers for group discounts (i.e., discounts for multiple users) or time-dependent discounts (i.e., e.g., discounts in the evening hours based on context information) can be proposed to users during the realization of a tourist activity in order to proactively enable adaptations of the tour activity. Besides this discussion for the tourism domain, the application of the presented heuristic approach to further activities or domains in the area of multi user context-aware processes is also possible and promising such as logistics or production (Beverungen et al. 2019; Hohmann and Posselt 2019), healthcare, disaster relief assistance or field work in companies in order to handle disruptions and proactive adaptations within such processes in terms of spatial-temporal coordination.

7. Conclusion, Limitations and Future Research

In this paper we present a heuristic technique for service re-selection, which is aimed at considering multiple users, context-awareness and in particular disruptive events at runtime. In this regard, we propose an approach that can maintain the features performance, solution quality, solution robustness and alternative solutions at runtime in the case of disruptive events. Existing service systems from the literature already refer to the complexity of user-based and context-based dependencies but do not consider disruptive events at runtime. To address this research gap, we developed a heuristic technique, which carefully divides the process into regions and efficiently select a service composition by using feasibility checks and a state space measure. In addition, the concept of a state space is considerably extended by a dynamic as well as precise state space re-structuring at runtime to deal with disruptive events. Alternative solutions are also available to the user in near-time if the selected solution does not meet the users' preferences and expectations. The evaluation results show that *MUCARS* can achieve significant improvements in performance and solution robustness at runtime compared to competing artifacts (i.e., Bortlik et al. 2018), while maintaining a high solution quality.

However, our approach is also subject to some limitations that need to be addressed in future research: *First*, we initially focused on real-world events (e.g., closed restaurant), which leads to a "service failure of a candidate service" in the model. However, mobile environments are volatile and different event types can occur at runtime. Our starting point in this paper could be used to investigate further changes in the model due to real-world events (e.g., changes in non-function properties values such as prices). Here, we are very confident that the presented approach can be extended and transferred to a wide range of further event types as important parts of the proposed heuristic such as the dynamic re-structuring of the state space can be directly re-used. In addition, synergies can arise from the concurrent processing of several different events leading to fewer changes in the process flow for all users. Here, a promising idea may be the use of monitoring and event handling approaches that enable the efficient processing of multiple events at the same time (Ayed et al. 2013; Chen and Rabhi 2016; Flouris et al. 2017; Kum 2020; Wang et al. 2017). *Second*, the evaluation shows that with an increasing number of users no proportional increase of the computation time can be achieved. The reason for this is the over-proportional increase in the number of states within the state space required to coordinate and select a solution for multiple participating users. To overcome this limitation, further research in the area of service systems may focus on enhancing the concept of these coordination-relevant states and therefore the mapping of user-based dependencies. Here, a promising idea may be the development of an approach in order to efficiently identify and omit non-relevant states and thus to improve the creation and processing of coordination-relevant states. *Finally*, with *MUCARS* a fast support at runtime is possible and thus the approach enables near-time support (i.e., *MUCARS* can select a solution on average 60 times faster than existing approaches). However, interactive support is becoming increasingly important in mobile environments (Li and Chen 2019). Since the *MUCARS* runtime is in the lower seconds range (e.g., *MUCARS* requires about 11 seconds in Setting I), real-time interaction is only possible to a limited extent, which possibly influences the user satisfaction (cf. Section 2.1). To improve real-time interaction, a dialogue could be built in that visualizes the user the resulting problem when a disruptive event occurs. The time until the user comprehends this problem and reacts accordingly could then be used to re-select a new solution in the background and present it immediately to the user if preferred. Currently, we work on such concepts of intelligent user interaction with other researchers. In conclusion, the provided multi user context-aware service re-selection approach can serve as promising first step for contributing to the important topic of disruptive events at runtime.

References

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., & Sikora, F. (2015). On the Complexity of QoS-Aware Service Selection Problem. In A. Barros, D. Grigori, N. C. Narendra, & H. K. Dam (Eds.), *Service-Oriented Computing* (Vol. 9435, pp. 345–352, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ai, L., & Tang, M. (2008). QoS-Based Web Service Composition Accommodating Inter-service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm. In *2008 IEEE Fourth International Conference on eScience (eScience), Indianapolis, USA* (pp. 119–126). IEEE. doi:10.1109/eScience.2008.110.

Allied Market Research. (2021). Location Based Services Market. https://www.alliedmarketresearch.com/location-based-services-market. Accessed 16 November 2021.

Alrifai, M., & Risse, T. (2009). Combining global optimization with local selection for efficient QoS-aware service composition. In J. Quemada, G. León, Y. Maarek, & W. Nejdl (Eds.), *18th International Conference on World Wide Web, Madrid, Spain, 20.04.2009 - 24.04.2009* (p. 881). New York, New York, USA: ACM Press. doi:10.1145/1526709.1526828.

Alrifai, M., Risse, T., & Nejdl, W. (2012). A hybrid approach for efficient Web service composition with end-to-end QoS constraints. *ACM Transactions on the Web, 6*, 1–31. doi:10.1145/2180861.2180864.

Alter, S. (2012). Metamodel for service analysis and design based on an operational view of service and service systems. *Service Science, 4*(3), 218–235.

Alter, S. (2017). Service System Axioms that Accept Positive and Negative Outcomes and Impacts of Service Systems. In .

Angarita, R., Cardinale, Y., & Rukoz, M. (2013). Dynamic recovery decision during composite web services execution. In L. Ladid, A. Montes, P. A. Bruck, F. Ferri, & R. Chbeir (Eds.), *5th International Conference on Management of Emergent Digital EcoSystems, Luxembourg, Luxembourg, 28.10.2013 - 31.10.2013* (pp. 187–194). New York, New York, USA: ACM Press. doi:10.1145/2536146.2536152.

Angarita, R., Cardinale, Y., & Rukoz, M. (2014). Reliable composite web services execution: towards a dynamic recovery decision. *Electronic Notes in Theoretical Computer Science, 302*(0), 5–28.

Aouatef, C., Iman, B., & Allaoua, C. (2008). Adaptive composition of services in context-aware ambient intelligent systems. In B. C. Desai (Ed.), *the 2014 International C* Conference, Montreal, QC, Canada, 03.08.2014 - 05.08.2014* (pp. 1–5). New York, New York, USA: ACM Press. doi:10.1145/2641483.2641535.

Ardagna, D., & Pernici, B. (2006). Global and Local QoS Guarantee in Web Service Selection. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Business Process Management Workshops* (Vol. 3812, pp. 32–46, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ardagna, D., & Pernici, B. (2007). Adaptive Service Composition in Flexible Processes. *IEEE Transactions on Software Engineering, 33*, 369–384. doi:10.1109/TSE.2007.1011.

Australian Government. (2021). Climate statistics for Australian locations: Monthly climate statistics. http://www.bom.gov.au/climate/averages/tables/cw_086071.shtml. Accessed 6 November 2021.

Ayed, D., Asim, M., & Llewellyn-Jones, D. (2013). An event processing approach for threats monitoring of service compositions. In *International Conference on Risks and Security of Internet and Systems (CRiSIS), La Rochelle, France* (pp. 1–10).

Baccari, S., & Neji, M. (2016). Design for a context-aware and collaborative mobile learning system. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai* (pp. 1–6). IEEE. doi:10.1109/ICCIC.2016.7919578.

Barber, F., & Salido, M. A. (2015). Robustness, stability, recoverability, and reliability in constraint satisfaction problems. *Knowledge and Information Systems, 44*, 719–734. doi:10.1007/s10115-014-0778-3.

Bearzotti, L. A., Salomone, E., & Chiotti, O. J. (2012). An autonomous multi-agent approach to supply chain event management. *International Journal of Production Economics, 135*, 468–478. doi:10.1016/j.ijpe.2011.08.023.

Beverungen, D., Müller, O., Matzner, M., Mendling, J., & Vom Brocke, J. (2019). Conceptualizing smart service systems. *Electronic Markets, 29*, 7–18. doi:10.1007/s12525-017-0270-5.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization. *ACM Computing Surveys, 35*, 268–308. doi:10.1145/937503.937505.

Bobek, S., & Nalepa, G. J. (2017). Uncertain context data management in dynamic mobile environments. *Future Generation Computer Systems, 66*, 110–124. doi:10.1016/j.future.2016.06.007.

Böhmann, T., Leimeister, J. M., & Möslein, K. (2014). Service systems engineering. *Business & Information Systems Engineering, 6*(2), 73–79.

Bortlik, M., Heinrich, B., & Mayer, M. (2018). Multi User Context-Aware Service Selection for Mobile Environments. *Business & Information Systems Engineering, 60*, 415–430. doi:10.1007/s12599-018-0552-2.

Boudaa, B., Hammoudi, S., Mebarki, L. A., Bouguessa, A., & Chikh, M. A. (2017). An aspect-oriented model-driven approach for building adaptable context-aware service-based applications. *Science of Computer Programming, 136*, 17–42.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82–117. doi:10.1016/j.ins.2013.02.041.

Bucchiarone, A., Cappiello, C., Di Nitto, E., Pernici, B., & Sandonini, A. (2012). A Variable Context Model for Adaptable Service-Based Applications. *International Journal of Adaptive, Resilient and Autonomic Systems, 3*, 35–53. doi:10.4018/jaras.2012070103.

Buys, J., Florio, V. de, & Blondia, C. (2011). Towards context-aware adaptive fault tolerance in SOA applications. In O. Etzion, A. Gal, S. Zdonik, P. Vincent, & D. Eyers (Eds.), *the 5th ACM international conference, New York, USA, 11.07.2011 - 15.07.2011* (p. 63). New York, USA: ACM Press. doi:10.1145/2002259.2002271.

Cao, H., Feng, X., Sun, Y., Zhang, Z., & Wu, Q. (2007). A Service Selection Model with Multiple QoS Constraints on the MMKP. In *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), Dalian, Liaoning, China* (pp. 584–589). IEEE. doi:10.1109/NPC.2007.35.

Cao, Z., Zhang, X., Zhang, W., Xie, X., Shi, J., & Xu, H. (2015). A Context-Aware Adaptive Web Service Composition Framework. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India* (pp. 62–66). IEEE. doi:10.1109/CICT.2015.68.

Caserta, M., & Voß, S. (2019). The robust multiple-choice multidimensional knapsack problem. *Omega, 86*, 16–27. doi:10.1016/j.omega.2018.06.014.

Chen, W., & Rabhi, F. A. (2016). Enabling user-driven rule management in event data analysis. *Information Systems Frontiers, 18*, 511–528. doi:10.1007/s10796-016-9633-2.

Cherif, S., Djemaa, R. B., & Amous, I. (2019). Reflective approach to improve self-adaptation of Web service compositions with autonomic middleware. *International Journal of Pervasive Computing and Communications, 15*, 144–173. doi:10.1108/IJPCC-01-2019-0001.

Corradini, F., Meschini, G., Polzonetti, A., & Riganelli, O. (2007). A Rule-Driven Business Process Design. In *2007 29th International Conference on Information Technology Interfaces, Cavtat, Croatia* (pp. 401–406). IEEE. doi:10.1109/ITI.2007.4283804.

David, K., Geihs, K., Leimeister, J. M., Roßnagel, A., Schmidt, L., Stumme, G., et al. (Eds.). (2014). *Socio-technical Design of Ubiquitous Computing Systems*. Cham: Springer International Publishing.

Deng, S., Huang, L., Hu, D., Zhao, J. L., & Wu, Z. (2016). Mobility-Enabled Service Selection for Composite Services. *IEEE Transactions on Services Computing, 9*, 394–407. doi:10.1109/TSC.2014.2365799.

Di Napoli, C., Ribino, P., & Serino, L. (2021). Correction to: Customisable assistive plans as dynamic composition of services with normed-QoS. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.1007/s12652-021-02978-4.

Evers, C., Kniewel, R., Geihs, K., & Schmidt, L. (2014). The user in the loop: Enabling user participation for self-adaptive applications. *Future Generation Computer Systems, 34*, 110–123. doi:10.1016/j.future.2013.12.010.

Faieq, S., Front, A., Saidi, R., El Ghazi, H., & Rahmani, M. D. (2019). A context-aware recommendation-based system for service composition in smart environments. *Service Oriented Computing and Applications, 13*, 341–355. doi:10.1007/s11761-019-00277-7.

Faieq, S., Saidi, R., El Ghazi, H., Front, A., & Rahmani, M. D. (2021). Building adaptive context-aware service-based smart systems. *Service Oriented Computing and Applications, 15*, 21–42. doi:10.1007/s11761-020-00310-0.

Fakhfakh, S., Am Hein, Jankovic, M., & Chazal, Y. (2020). A meta-model for product service systems of systems. In *Proceedings of the Design Society: DESIGN Conference* (Vol. 1, pp. 1235–1244). Cambridge University Press.

Fekih, H., Mtibaa, S., & Bouamama, S. (2019a). An Efficient User-Centric Web Service Composition Based on Harmony Particle Swarm Optimization. *International Journal of Web Services Research, 16*, 1–21. doi:10.4018/IJWSR.2019010101.

Fekih, H., Mtibaa, S., & Bouamama, S. (2019b). The dynamic reconfiguration approach for fault-tolerance web service composition based on multi-level VCSOP. *Procedia Computer Science, 159*, 1527–1536. doi:10.1016/j.procs.2019.09.323.

Femenia-Serra, F., Neuhofer, B., & Ivars-Baidal, J. A. (2019). Towards a conceptualisation of smart tourists and their role within the smart destination scenario. *The Service Industries Journal, 39*, 109–133. doi:10.1080/02642069.2018.1508458.

Fisk, R. P., Patricio, L., Edvardsson, B., Ng, G., Min, C. Z., Firth, R., et al. (2011). Does service-dominant design result in a better service system? *Journal of Service Management*(22), 540–556.

Fkaier, S., Romdhani, M., Khalgui, M., & Frey, G. (2017). Context-awareness Meta-model for Reconfigurable Control Systems. In *ENASE* (pp. 226–234). Enase.

Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., & Mock, M. (2017). Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software, 127*, 217–236.

Galletta, D. F., Henry, R., McCoy, S., & Polak, P. (2004). Web site delays: How tolerant are users? *Journal of the Association for Information Systems, 5*(1), 1.

Gao, H., Huang, W., Yang, X., Duan, Y., & Yin, Y. (2018). Toward service selection for workflow reconfiguration:An interface-based computing solution. *Future Generation Computer Systems, 87*, 298–311. doi:10.1016/j.future.2018.04.064.

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of network and computer applications, 39*, 319–333.

Gendreau, M., & Potvin, J.-Y. (2005). Metaheuristics in Combinatorial Optimization. *Annals of Operations Research, 140*, 189–213. doi:10.1007/s10479-005-3971-7.

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics* (Vol. 146). Boston, MA: Springer US.

Ghallab, M., Nau, D., & Traverso, P. (2008). *Automated planning: Theory and practice*. Amsterdam: Elsevier.

Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence, 25*, 503–526. doi:10.1080/0952813X.2013.782347.

Grotherr, C., Semmann, M., & Böhmann, T. (2018). Engaging Users to Co-Create-Implications for Service Systems Design by Evaluating an Engagement Platform. *Hawaii International Conference on System Sciences (HICSS)*.

Hara, T., Aoyama, K., Kurata, Y., & Yabe, N. (2016). Service Design in Tourism: Encouraging a Cooperative Relationship Between Professional Design and Non-professional Design. In S. K. Kwan, J. C. Spohrer, & Y. Sawatani (Eds.), *Global Perspectives on Service Science: Japan* (Vol. 75, pp. 119–135, Service Science: Research and Innovations in the Service Economy). New York, NY: Springer New York.

Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science, 1*, 1. doi:10.1186/2194-0827-1-1.

He, Q., Han, J., Yang, Y., Grundy, J., & Jin, H. (2012). QoS-Driven Service Selection for Multi-tenant SaaS. In *2012 IEEE 5th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA* (pp. 566–573). IEEE. doi:10.1109/CLOUD.2012.125.

Heinrich, B., Klier, M., Lewerenz, L., & Mayer, M. (2015). Enhancing decision support in multi user service selection. *36th International Conference on Information Systems (ICIS)*.

Heinrich, B., & Lewerenz, L. (2015). Decision support for the usage of mobile information services: A context-aware service selection approach that considers the effects of context interdependencies. *Journal of decision systems, 24*, 406–432. doi:10.1080/12460125.2015.1080498.

Heinrich, B., & Mayer, M. (2018). Service selection in mobile environments: considering multiple users and context-awareness. *Journal of decision systems, 27*, 92–122. doi:10.1080/12460125.2018.1513223.

Heinrich, B., & Schön, D. (2015). Automated Planning of Context-aware Process Models. *European conference on information systems*. doi:10.18151/7217352.

Heinrich, B., Zellner, G., & Leist, S. (2011). Service integrators in business networks—the importance of relationship values. *Electronic Markets, 21*, 215–235. doi:10.1007/s12525-011-0075-x.

Hidri, W., M'tir, R. H., Saoud, N. B. B., & Ghedira-Guegan, C. (2019). A Meta-model for context-aware adaptive Business Process as a Service in collaborative cloud environment. *Procedia Computer Science, 164*, 177–186.

Hohmann, C., & Posselt, T. (2019). Design challenges for CPS-based service systems in industrial production and logistics. *International Journal of Computer Integrated Manufacturing, 32*, 329–339. doi:10.1080/0951192X.2018.1552795.

Hoxmeier, J. A., & DiCesare, C. (2000). System response time and user satisfaction: An experimental study of browser-based applications. *AMCIS 2000 Proceedings*, 347.

Jin, H., Lin, R., Zou, H., & Shuai, T. (2012). A Novel Method for Optimizing Multi-User Service Selection. *Journal of Convergence Information Technology, 7*, 296–310. doi:10.4156/jcit.vol7.issue5.36.

Khan, S., Li, K. F., Manning, E. G., & Akbar, M. M. (2002). Solving the knapsack problem for adaptive multimedia systems. *Stud. Inform. Univ., 2*(1), 157–178.

Koskela-Huotari, K., Patrício, L., Zhang, J., Karpen, I. O., Sangiorgi, D., Anderson, L., et al. (2021). Service system transformation through service design: Linking analytical dimensions and service design approaches. *Journal of Business Research, 136*, 343–355. doi:10.1016/j.jbusres.2021.07.034.

Kum, D. K. (2020). Extended CEP Model for Effective Enterprise Systems Service Monitoring. *KSII Transactions on Internet and Information Systems (TIIS), 14*(2), 807–825.

Kurdija, A. S., Silic, M., Delac, G., & Vladimir, K. (2019). Fast Multi-Criteria Service Selection for Multi-User Composite Applications. *IEEE Transactions on Services Computing*, 1. doi:10.1109/TSC.2019.2925614.

Lewerenz, L. (2015). A Heuristic Technique for an Efficient Decision Support in Context-aware Service Selection. *Proceedings of the 36th International Conference on Information Systems (ICIS)*.

Li, S., & Chen, C.-H. (2019). The effects of visual feedback designs on long wait time of mobile application user interface. *Interacting with Computers, 31*, 1–12. doi:10.1093/iwc/iwz001.

Lin, K.-J., Zhang, J., Zhai, Y., & Xu, B. (2010). The design and implementation of service process reconfiguration with end-to-end QoS constraints in SOA. *Service Oriented Computing and Applications, 4*(3), 157–168.

Ma, Z., Zhang, X.-G., Zhu, Z.-Y., & Wang, H. (2015). The Implementation of Adaptive Web Service Composition. In *2015 International Conference on Computer Science and Applications (CSA), Wuhan, China* (pp. 303–307). IEEE. doi:10.1109/CSA.2015.35.

Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. *Journal of the Academy of Marketing Science, 36*, 18–20. doi:10.1007/s11747-007-0058-9.

Maleki, E., Belkadi, F., & Bernard, A. (2018). A meta-model for product-service system based on systems engineering approach. *Procedia CIRP, 73*, 39–44.

Mayer, M. (2017). Multi-user Service Re-Selection: React Dynamically to Events occuring at process Execution. *25th European Conference on Information Systems (ECIS)*.

Meredith, J. R., Raturi, A., Amoako-Gyampah, K., & Kaplan, B. (1989). Alternative research paradigms in operations. *Journal of Operations Management, 8*, 297–326. doi:10.1016/0272-6963(89)90033-8.

Moghaddam, M., & Davis, J. G. (2014). Service Selection in Web Service Composition: A Comparative Review of Existing Approaches. In A. Bouguettaya, Q. Z. Sheng, & F. Daniel (Eds.), *Web Services Foundations* (Vol. 33, pp. 321–346). New York, NY: Springer New York.

Mostofa Akbar, M., Sohel Rahman, M., Kaykobad, M., Manning, E. G., & Shoja, G. C. (2006). Solving the Multidimensional Multiple-choice Knapsack Problem by constructing convex hulls. *Computers & Operations Research, 33*, 1259–1273. doi:10.1016/j.cor.2004.09.016.

Muhammad, S. S., Dey, B. L., & Weerakkody, V. (2018). Analysis of Factors that Influence Customers' Willingness to Leave Big Data Digital Footprints on Social Media: A Systematic Review of Literature. *Information Systems Frontiers, 20*, 559–576. doi:10.1007/s10796-017-9802-y.

Nagarajan, N., & Thirunavukarasu, R. (2020). Service-oriented Broker for Effective Provisioning of Cloud Services – a Survey. *International Journal of Computing and Digital Systems, 9*, 863–879. doi:10.12785/ijcds/090508.

Neville, K., O'Riordan, S., Pope, A., Rauner, M., Rochford, M., Madden, M., et al. (2016). Towards the development of a decision support system for multi-agency decision-making during cross-border emergencies. *Journal of decision systems, 25*(sup1), 381–396.

Oizumi, K., Meguro, Y., Wu, C. X., & Aoyama, K. (2013). Supporting System for Tour Lineup Design based on Tour Service Modeling. In Y. Shimomura & K. Kimita (Eds.), *The Philosopher's Stone for Sustainability* (pp. 143–148). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pang, B., Hao, F., Yang, Y., & Park, D.-S. (2020). An efficient approach for multi-user multi-cloud service composition in human–land sustainable computational systems. *The Journal of Supercomputing, 76*, 5442–5459. doi:10.1007/s11227-019-03140-w.

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

Public Transport Victoria. (2019). Data and reporting: Monthly performance. https://www.ptv.vic.gov.au/footer/data-and-reporting/network-performance/monthly-performance/. Accessed 6 November 2021.

Rahmani, D., & Ramezanian, R. (2016). A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study. *Computers & Industrial Engineering, 98*, 360–372. doi:10.1016/j.cie.2016.06.018.

Roda, C., Navarro, E., Zdun, U., López-Jaquero, V., & Simhandl, G. (2018). Past and future of software architectures for context-aware systems: A systematic mapping study. *Journal of Systems and Software, 146*, 310–355. doi:10.1016/j.jss.2018.09.074.

Romero, M., Guédria, W., Panetto, H., & Barafort, B. (2020). Towards a characterisation of smart systems: A systematic literature review. *Computers in industry, 120*, 103224.

Saleh, A., Ismail, R., & Fabil, N. (2017). Evaluating Usability for Mobile Application. In Unknown (Ed.), *the 2017 International Conference, Hong Kong, Hong Kong, 28.12.2017 - 30.12.2017* (pp. 71–77). New York, USA: ACM Press. doi:10.1145/3178212.3178232.

Sandionigi, C., Ardagna, D., Cugola, G., & Ghezzi, C. (2013). Optimizing Service Selection and Allocation in Situational Computing Applications. *IEEE Transactions on Services Computing, 6*, 414–428. doi:10.1109/TSC.2012.18.

Sbihi, A. (2007). A best first search exact algorithm for the Multiple-choice Multidimensional Knapsack Problem. *Journal of Combinatorial Optimization, 13*, 337–351. doi:10.1007/s10878-006-9035-3.

Sedighiani, K., Shokrollahi, S., & Shams, F. (2021). BASBA: A framework for Building Adaptable Service-Based Applications. *Journal of Systems and Software, 179*, 110989. doi:10.1016/j.jss.2021.110989.

Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal, 14*, 159–178. doi:10.1007/s11219-006-7600-8.

Shen, Y., Wang, M., Tang, X., Luo, Y., & Guo, M. (2012). Context-aware HCI service selection. *Mobile Information Systems, 8*(3), 231–254.

Shen, Y.-h., & Yang, X.-h. (2011). A self-optimizing QoS-aware service composition approach in a context sensitive environment. *Journal of Zhejiang University SCIENCE C, 12*, 221–238. doi:10.1631/jzus.C1000031.

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences, 280*, 218–238. doi:10.1016/j.ins.2014.04.054.

Statista. (2019a). Number of available apps in the Apple App Store from July 2008 to January 2017. https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/. Accessed 16 November 2021.

Statista. (2019b). Size of mobile professional services market by segment worldwide in 2014 and 2018. https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/. Accessed 16 November 2021.

Tan, J., Ronkko, K., & Gencel, C. (2013). A Framework for Software Usability and User Experience Measurement in Mobile Industry. In *IWSM-MENSURA 2013* (pp. 156–164).

Tedeschi, L. O. (2006). Assessment of the adequacy of mathematical models. *Agricultural Systems, 89*, 225–247. doi:10.1016/j.agsy.2005.11.004.

TomTom. (2021). Melbourne traffic: Australia. https://www.tomtom.com/en_gb/traffic-index/melbourne-traffic/. Accessed 6 December 2021.

Tretola, G., & Zimeo, E. (2019). Reactive behavioural adaptation of service compositions. *Journal of Software: Evolution and Process, 31*, 161. doi:10.1002/smr.2201.

Tung, W.-F., Yuan, S.-T., Wu, Y.-C., & Hung, P. (2014). Collaborative service system design for music content creation. *Information Systems Frontiers, 16*, 291–302. doi:10.1007/s10796-012-9346-0.

Ventola, C. L. (2014). Mobile devices and apps for health care professionals: uses and benefits. *Pharmacy and Therapeutics, 39*(5), 356.

Wanchun, D., Chao, L., Xuyun, Z., & Chen, J. (2011). A QoS-Aware Service Evaluation Method for Co-selecting a Shared Service. In *2011 IEEE International Conference on Web Services (ICWS), Washington, DC, USA* (pp. 145–152). IEEE. doi:10.1109/ICWS.2011.11.

Wang, H., Peng, S., & Yu, Q. (2019). A parallel refined probabilistic approach for QoS-aware service composition. *Future Generation Computer Systems, 98*, 609–626. doi:10.1016/j.future.2019.03.053.

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., et al. (2010). Web Service Selection for Multiple Agents with Incomplete Preferences. In *2010 IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology (WI-IAT), Toronto, AB, Canada* (pp. 565–572). IEEE. doi:10.1109/WI-IAT.2010.27.

Wang, S., Hsu, C.-H., Liang, Z., Sun, Q., & Yang, F. (2014). Multi-user web service selection based on multi-QoS prediction. *Information Systems Frontiers, 16*, 143–152. doi:10.1007/s10796-013-9455-4.

Wang, X., Feng, Z., Huang, K., & Tan, W. (2017). An automatic self-adaptation framework for service-based process based on exception handling. *Concurrency and Computation: Practice and Experience, 29*(5), e3984.

Weinert, T., Janson, A., & Leimeister, J. M. (2020). Does Context Matter for Value Co-Creation in Smart Learning Services? *15th International Conference on Wirtschaftsinformatik.*

Will M. Bertrand, J., & Fransoo, J. C. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management, 22*, 241–264. doi:10.1108/01443570210414338.

Xu, L., & Jennings, B. (2010). A Cost-Minimizing Service Composition Selection Algorithm Supporting Time-Sensitive Discounts. In *2010 IEEE International Conference on Services Computing (SCC), Miami, FL, USA* (pp. 402–408). IEEE. doi:10.1109/SCC.2010.76.

Yu, H. Q., & Reiff-Marganiec, S. (2009). A Backwards Composition Context Based Service Selection Approach for Service Composition. In *2009 IEEE International Conference on Services Computing, Bangalore, India* (pp. 419–426). IEEE. doi:10.1109/SCC.2009.25.

Yu, T., & Lin, K.-J. (2007). Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Service-Oriented Computing – ICSOC 2007* (Vol. 4749, pp. 130–143, Lecture Notes in Computer Science). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yuan, S.-T. D., & Hsu, S.-T. (2017). Enhancing service system design: An entity interaction pattern approach. *Information Systems Frontiers, 19*, 481–507. doi:10.1007/s10796-015-9604-z.

Zhai, Y., Zhang, J., & Lin, K.-J. (2009). SOA middleware support for service process reconfiguration with end-to-end QoS constraints. In *IEEE International Conference on Web Services* (pp. 815–822).

Zhang, D., Adipat, B., & Mowafi, Y. (2009). User-centered context-aware mobile applications—The next generation of personal mobile computing. *Communications of the Association for Information Systems, 24*(1), 3.

Zhang, J., Lee, J., & Lin, K.-J. (2012a). Context-Aware Proactive Process Reconfiguration in Service-Oriented Architecture. In *2012 IEEE 14th International Conference on Commerce and Enterprise Computing (CEC), Hangzhou, China* (pp. 62–69). IEEE. doi:10.1109/CEC.2012.19.

Zhang, J., Zhou, A., Sun, Q., Wang, S., & Yang, F. (2018). Overview on Fault Tolerance Strategies of Composite Service in Service Computing. *Wireless Communications and Mobile Computing, 2018*, 1–8. doi:10.1155/2018/9787503.

Zhang, M., Liu, C., Yu, J., Zhu, Z., & Zhang, B. (2013a). A correlation context-aware approach for composite service selection. *Concurrency and Computation: Practice and Experience, 25*, 1909–1927. doi:10.1002/cpe.2988.

Zhang, M., Ranjan, R., Haller, A., Georgakopoulos, D., & Strazdins, P. (2012b). Investigating decision support techniques for automating Cloud service selection. In *2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), Taipei, Taiwan, 2012* (pp. 759–764). IEEE. doi:10.1109/CloudCom.2012.6427501.

Zhang, Z., Zheng, S., Li, W., Tan, Y., Wu, Z., & Tan, W. (2013b). Leveraging Genetic Algorithm to Compose Web Services in a Context-Aware Environment. In *2013 IEEE International Conference on Systems, Man and Cybernetics (SMC 2013), Manchester* (pp. 829–834). IEEE. doi:10.1109/SMC.2013.147.

Zheng, Z., & Lyu, M. R. (2009). A QoS-aware fault tolerant middleware for dependable service composition. In *Networks (DSN), Lisbon, Portugal* (pp. 239–248). IEEE. doi:10.1109/DSN.2009.5270332.

Zheng, Z., Zhang, Y., & Lyu, M. R. (2014). Investigating QoS of Real-World Web Services. *IEEE Transactions on Services Computing, 7*, 32–39. doi:10.1109/TSC.2012.34.

Zhu, W., Yin, B., Gong, S., & Cai, K.-Y. (2017). An Approach to Web Services Selection for Multiple Users. *IEEE Access, 5*, 15093–15104. doi:10.1109/ACCESS.2017.2722228.

Appendix: Paper 4

Detailed comparison of the proposed approach with the approach by Bortlik et al. (2018)

	Approach by Bortlik et al. (2018)	*Proposed approach*
Research Question	How to design a heuristic technique for the multi user context-aware service selection that determines a close-to-optimal solution in a short amount of time while scaling efficiently with problem size?	How to design an optimization-based heuristic technique for *re-selecting* services under consideration of multiple users, context information and in particular *disruptive events* at *runtime*?
Main Focus	Dealing with user-based dependencies and context-based dependencies at *planning time*, i.e., *determining* a service *composition* for each user *from scratch*	Dealing with *disruptive events* (respecting user-based dependencies and context-based dependencies) at *runtime*, i.e., *adapting a given* service *composition* for each user
Theoretical Grounding	Service selection Service systems engineering at planning time	Service re-selection Runtime decision support
Approach	Heuristic technique based on a stateful representation (i.e., state space): - Building entire state space - Service selection on the entire state space	Heuristic technique based on a stateful representation (i.e., state space): - Adapting only parts of the state space in a careful manner - Service re-selection on these parts of the state space
Findings	Approach to support multi user context-aware service systems providing a multi user-oriented decomposition and service selection to enable and enhance the contextualization and collaboration.	Approach to handle disruptive events at runtime providing a sophisticated region-based method and a dynamic as well as precise state space re-structuring. Efficient re-selection based on feasibility checks and a state space measure.
Target Criteria	Performance Solution quality (near-optimal solution)	Performance Solution quality (near-optimal solution) Solution robustness Alternative solutions

References

Bortlik, Michael; Heinrich, Bernd; Mayer, Michael (2018): Multi User Context-Aware Service Selection for Mobile Environments. In: *Business & Information Systems Engineering* 60 (5), S. 415–430. DOI: 10.1007/s12599-018-0552-2.

List of Abbreviations

Abbreviation	*Meaning*
CS	Candidate Service
GPS	Global Positioning System
MMKP	Multi-choice Multi-dimensional Knapsack Problem
MUCAHA	Multi User Context-Aware Heuristic Approach
MUCAOS	Multi User Context-Aware Optimal Solution
MUCAP	Multi User Context-Aware Process
MUCARS	Multi User Context-Aware Re-Selection
NP-hard	Non-deterministic Polynomial-time hardness
WS	World State

Formal Notation of the Optimization Model

A	set of all users conducting the multi user process
$\lvert A \rvert$	cardinality of all users
a	index of single user a
BL	set of all blacklisted candidate services
CA	set of all context-aware attributes
CA^-	set of all minimizing context-aware attributes
CA^+	set of all maximizing context-aware attributes
ca_l	single context-aware attribute with the index l
$con_a^{ca_l}$	single global non-context-aware-attribute constraint for context-aware attribute ca_l and user a
$con_a^{nca_n}$	single global non-context-aware-attribute constraint for non-context-aware attribute nca_n and user a
$CREATE\_WS(wsc_{aty})$	function which returns the corresponding world state that emerges from the world-state-candidate-service combination wsc_{aty}
CS_{at}	set of functional equivalent candidate services for user a and task t
cs_{ats}	single candidate service for user a and task t with the index s
$CS(wsc_{aty})$	function which returns the corresponding candidate service for the world-state-candidate-service combination wsc_{aty}
$CWSC_t$	set of common world-state-candidate-service combination for task t
$cwsc_{tz}$	single common world-state-candidate-service combination for task t with index z
$CWSC(a, cwsc_{tz})$	function which returns the corresponding world-state-candidate-service combination of user a for $cwsc_{tz}$
d	index of event d
ev_{ad}	single event ev_{ad}
F	set of tasks which can be conducted by multiple users simultaneously
F_a	set of tasks which can be conducted by user a simultaneously
I	set of tasks which can be conducted by multiple users individually
I_a	set of tasks which can be conducted by user a individually
k	index of world state k
l	index of context-aware attribute l
n	index of non-context-aware attribute n
NCA	set of non-context-aware-attributes
NCA^-	set of all minimizing non-context-aware attributes
NCA^+	set of all maximizing non-context-aware attributes

nca_n	single non-context-aware-attribute with index n
NFP	set of non-functional properties
p	process p referring to a set of tasks T
$q_{aty}^{ca_l}$	single quantified non-context-aware-attribute value for context-aware attribute ca_l referring wsc_{aty}
$q_{ats}^{nca_n}$	single quantified non-context-aware-attribute value for non-context-aware attribute nca_n referring cs_{ats}
s	index of candidate service s
sc_a	single service composition for user a
T	set of all tasks
T_a	set of all tasks for user a
t	index of task t
$U(wsc_{aty})$	function which returns the utility of wsc_{aty}
WS_{at}	set of world states for user a and task t
ws_{atk}	single world state for user a and task t with the index k
$WS(wsc_{aty})$	function which returns the corresponding world state for the world-state-candidate-service combination wsc_{aty}
WSC_{at}	set of world-state-candidate-service combination for user a and for task t
wsc_{aty}	single world-state-candidate-service combination for user a and task t with the index y
y	index of world-state-candidate-service combination y
z	index of common world-state-candidate-service combination z

4 Conclusion

This section summarizes the major findings of this dissertation (cf. Section 4.1), implications (cf. Section 4.2) and limitations as well as starting points for further research (cf. Section 4.3).

4.1 Major Findings

The volume, variety and velocity of digital data is increasing due to developments in web and mobile environments such as social media, the internet of things, mobile technologies and mobile business. The resulting problems *information overload* and *uncertainty* lead to more complex decision-making for users. Decision support systems and, in particular, recommender and service systems offer enormous potential to help users making decisions in a wide variety of domains. Furthermore, two circumstances can be recognized: First, research in the field of data quality shows that the impact of data quality on the evaluation criteria of data-driven decision support systems is increasingly being investigated in order to support decision-making. Second, mobile devices that use physical sensors to gather context information are used primarily and cause unforeseen disruptive events during process execution. To analyze these developments in more detail, four central research questions in the following topics were defined: *Data Quality in Recommender Systems* (cf. Section 2) and *Disruptive Events in Service Systems* (cf. Section 3). Therefore, this dissertation provides valuable concepts and methods for decision support and contributes specifically to item and service recommendations for users in web and mobile environments. The major findings for these two topics are summarized in the following.

Topic 1: Data Quality in Recommender Systems

On the one hand, the dissertation presents new insights for extending item content data sets in the context of recommender systems (cf. Section 2.1) and also for the impact of data quality on the prediction accuracy of recommender systems (cf. Section 2.2). First, the dissertation proposes and evaluates a systematic and tangible procedure for the extension of a data set with 1) *item content data of another data set from the same domain* and 2) *imputation of missing values* in the context of recommender systems. In this regard, the procedure aims to address data quality, especially by increasing the completeness of item content data sets and, in consequence, to improve the recommendation quality of recommender systems. Therefore, RQ1 enables the exact elaboration of the necessary steps for increasing the completeness of item content data, which can serve as a template in a wide variety of domains. Second, the dissertation generally confirms the *positive impact between the completeness of item content data on the prediction accuracy of recommendations*. In this regard, RQ2 presents a theoretical model based on the literature and derives hypotheses to reveal that more features and feature values lead to a significantly higher prediction accuracy of recommendations. Furthermore, the impact of completeness on the increase in prediction accuracy is positively moderated by the amount of additional features and their feature values per items and per users, but is negatively moderated by features and especially by the diversity of features. This means that adding features, which differ significantly from already existing features, does not necessarily yield a higher increase in prediction accuracy.

Topic 2: Disruptive Events in Service Systems

On the other hand, the dissertation derives an *event handling strategy for multi user context-aware service systems* (cf. Section 3.1) and a *heuristic technique for service re-selection* (cf. Section 3.2). First, the dissertation presents a novel event handling strategy for multi user context-aware service systems based on user preferences to actively support user participation in mobile environments. Thus, RQ3 identifies user-based rules and mechanisms to systematically process events, especially with regard to context information and multiple users. Therefore, RQ3 aims to minimize the number of adaptations of already planned service compositions. In this regard, a novel mechanism was developed for the uniform distribution of local constraints for multiple users. Second, the dissertation presents a heuristic technique for service re-selection at runtime that explicitly deal with multiple users, context-awareness and in particular disruptive events at runtime. In this respect, RQ4 reveals that the regional division of a process has positive effects on runtime features (i.e., performance, quality and robustness) of the re-selected service composition. Furthermore, a dynamic re-structuring of the state space (extension and reduction) significantly increases the performance of a service system in contrast to the construction of an entire new state space. Sophisticated state space measures and feasibility checks enable to reduce the number of time-consuming re-selection steps and also have advantages in regard to runtime features. Finally, by re-using re-selection steps from the initial solution at planning time, alternative solutions can be proactively provided to the user with high performance.

In summary, the dissertation specifies concepts and methods for recommender and service systems to enable successful decision support considering challenges in the broader context of big data such as information overload and uncertainty.

4.2 Summary of Implications

This section summarizes the implications for research and practice based on the findings from section 4.1 for the two following topics of this dissertation in the context of decision support.

Topic 1: Data Quality in Recommender Systems

This work provides a tangible and evaluated procedure (cf. Section 2.1), which enables to increase the completeness of item content data. Furthermore, the detailed testing of hypotheses (cf. Section 2.2) extended the existing body of knowledge from the literature that the completeness of item content data has a significant positive impact on the prediction accuracy of recommender systems. The resulting improved recommendations favored by the procedure and examined by hypotheses strengthens the customer loyalty between businesses (i.e., e.g., providers of web platforms) and their users and increases the potential for cross- and up-selling opportunities (Jannach and Adomavicius 2016). The procedure can thus be used for a wide variety of domains for decision support in web environments (e.g., restaurants and movies). In this regard, the evaluation of the hypotheses shows that a targeted extension of item content data is elementary. This means, portals that potentially use different web sources (e.g., trivago.com), should also include item content data in addition to user ratings and user reviews in the data extension as this leads to a better reflection of user preferences and consequently to better recommendations. However, providers of web portals must ensure that only important additional features are extended (may be indicated by a high amount of available feature values), which create added value to the user. Furthermore, the hypotheses show that data quality is a significant driver for the improvement of recommendations, and therefore not only the recommendation algorithm itself is a decisive factor for the improvement of recommendation quality (Schnabel et al. 2018).

Topic 2: Disruptive Events in Service Systems

The proposed approach in this dissertation provides initial findings for event handling in multi user context-aware service systems on the basis of an empirical study (cf. Section 3.1) and thus forms the basis for further research in the field of service systems engineering. In particular, the research area of *Engineering Service Systems Interaction (Böhmann et al. 2014)* is expanded for fault tolerance and event processing with user-related rules and mechanisms, especially for context information and multiple users. Further, the heuristic technique for service re-selection in multi user context-aware service systems (cf. Section 3.2) extends existing meta models from the literature (c.f., e.g., Hidri et al. 2019) in the sense that user requirements are detailed and directly included in the design of the service system. In addition, the service re-selection algorithm extends the analysis component of service systems (c.f., e.g., Boudaa et al. 2017) with feasibility checks and state space measures that are able to control the re-selection at runtime. The results of this dissertation open the research field to develop further metrics related to dependencies on multiple users, context information and disruptive events. The individual rules and mechanisms for event handling and also for service re-selection enable the user to be proactively engaged in mobile applications such as *Culture Trip*. This enables the user to be involved in every decision during the process execution at runtime. In this way, the supporting service systems of mobile applications can inform the user about upcoming events at an early stage (e.g., context changes regarding the weather) and preventively offer more suitable alternative solutions on the basis of user preferences. Also, businesses have the opportunity to interact with users in real time (e.g., possibility of discounts). Thus, taking user preferences into account enables stronger user participation and ultimately increase the performance of mobile applications and especially service systems (Ye and Kankanhalli 2020).

There are a high number of implications for research and practice, but a plethora of interesting directions for further research remain.

4.3 Limitations and Directions for Further Research

The dissertation comprises two main topics in the area of decision support with a total of four research questions. The concepts and methods for decision support, in particular information filtering and service selection, are very distinct research areas that cannot be limited to the scope of this thesis. A plethora of promising challenges remain whereas some directions for further research are outlined in the following.

Topic 1: Data Quality in Recommender Systems

(I): The approaches of this dissertation are currently being applied and evaluated to e-commerce real-world scenarios in the domain's restaurants and movies (cf. Section 2.1 and 2.2). It seems highly promising to examine the data quality of recommender systems with *other data sets* (e.g., amazon dataset; cf. Da'u and Salim 2020), *other domains* (e.g., books; cf. Anwar et al. 2020) *and/or other real-world scenarios* outside of e-commerce (e.g., tourism or healthcare; cf. Ko et al. 2022). This extension to additional application contexts can further substantiate the evaluation results and it could also reveal some new interesting findings in the development of procedures (cf. Section 2.1) and the resulting effects on recommendation quality (cf. Section 2.1 and 2.2).

(II): Data quality is a multidimensional construct comprising several dimensions (Batini and Scannapieco 2016). It would therefore be fascinating to further investigate and assess the impact of *other data quality dimensions* in decision-making such as currency (cf., e.g., Heinrich and Klier 2011; Heinrich et al. 2023) or consistency (cf., e.g., Heinrich et al. 2018) in addition to the highly relevant dimension in this dissertation, *completeness*. This would gain new insights for the topic data quality in recommender systems and therefore provides valuable information about interdependencies between different data quality dimensions. Moreover, *artificial intelligence-based recommender algorithms* such as text mining, neural networks or clustering are increasingly used in the literature for recommendations (Ko et al. 2022). Thus, it would be interesting - in addition to the highly relevant recommender algorithm of this dissertation *matrix factorization* - to analyze the effects of the interplay between data quality and various (content-based) recommender algorithms.

(III): This dissertation analyzes the extension of *item content data* based on additional structured item content data. Another interesting perspective might be to incorporate the extension of *user data* and/or *web logs* (cf., e.g., Ko et al. 2022). Additional user data or web logs based on, for example, different online social network accounts could provide further user-specific information to improve recommendations. The use of artificial intelligence algorithms such as deep learning or natural language processing appears particularly promising for identifying and analyzing user data in order to extend an item content data set. Therefore, the following user data is especially relevant in current research: Social connections (Bai et al. 2020), user behavior (Li et al. 2018) or context information (Smirnova and Vasile 2017). Moreover, the literature reveals that the *analysis of unstructured data* (e.g., online customer reviews) in the context of recommender systems is steadily increasing (Ray et al. 2021; Srifi et al. 2020). There are already language models that extract information from unstructured data (e.g., BERT; i.e., deep learning algorithm, cf. Xu et al. 2019). It seems promising to analyze unstructured data and to extend item content data sets with further information of unstructured data to better understand (i.e., data understanding; cf. Ko et al. 2022) the individual needs of users.

Topic 2: Disruptive Events in Service Systems

(I): In order to investigate decision support for service systems in mobile environments under consideration of disruptive events, we utilized real-world examples from the tourism sector as

this domain represents a valuable part of the further development of service systems (Koskela-Huotari et al. 2021). However, there are *other domains* in which multi user context-aware processes including disruptive events can be found such as healthcare, disaster relief assistance, field work in companies, everyday efficiency and planning, roadside, logistics or production (Beverungen et al. 2019; Hohmann and Posselt 2019; Neville et al. 2016; Ventola 2014; Zhang et al. 2009). It would be highly interesting to apply the presented approaches to further domains. Moreover, mobile environments are very volatile and a large number of (different) events can occur (Georgantas 2018; Verma and Srivastava 2018). Currently, the analyses of this dissertation are limited to the event types *changes in quality of service* and *changes of context information* (cf. Section 3.1) as well as *service failure of a candidate service* (cf. Section 3.2). An extension of the event types by, for example, *changes to user preferences* or *changes to user constraints* would provide new insights for the topic disruptive events in service systems.

(II): This dissertation presents an event handling strategy for multi user context-aware service systems based on an empirical study (cf. Section 3.1). Although the rules and mechanisms were derived directly and well-defined on the basis of surveyed user preferences, a *further validation* of the presented approach is required in the next step. To this end, the approach should be tested against defined runtime features from the literature (cf., e.g., Di Napoli et al. 2021; Wang et al. 2019) such as performance or solution quality and also be evaluated against competing artefacts that handle each event independently (c.f. Bortlik et al. 2023).

(III): Interactive support for users at runtime is becoming increasingly important in mobile environments (Li and Chen 2019). Although the current service re-selection approach is 60 times faster than competing artefacts (cf. Section 3.2), it can only provide limited real-time interaction, especially with an increasing number of users involved in the multi user context-aware process. The reason is the disproportionate increase in the number of states in the state space, which are required for finding solutions, while considering dependencies between multiple users and context information. Two interesting directions for further research are described in the following.

- In order to achieve real-time interaction, a further *optimization of the stateful representations* (cf., e.g., Heinrich and Lewerenz 2015) is necessary in the first step. Therefore, the approach would significantly benefit from a further reduction in the number of states, especially with an increasing number of users. In this context, only relevant states should remain in order to minimize the complexity for a runtime approach. Furthermore, there are also approaches in the literature that solve these complex decision problems with stateless models and thus integrate dependencies in relation to context information and multiple users directly into the optimization model (cf., e.g., Heinrich and Mayer 2018). The approach Heinrich and Mayer (2018) reveals that in principle the performance of a stateless model is better than a stateful model (in scenarios at planning time). It is therefore highly promising to develop a stateless model for a runtime optimized approach.
- Another direction of further research is to *analyze, test and integrate visual feedback mechanisms* (e.g., bar indicator, pie indicator, cartoon indicator; cf. Chen and Li 2020) into the

multi user context-aware approach in order to improve the users waiting experience. Therefore, the effectiveness of this mechanisms is heavily influenced by the users underlying perceptions (Li et al. 2022). Thus, the success of an application depends not only on its actual performance, but above all on the perceived performance (Hohenstein et al. 2016). In this regard, the literature suggests interactive loading screens (Cheng et al. 2023), for example, which shorten the perceived waiting time and increases user satisfaction.

To conclude, based on the major findings and implications, the dissertation enables many interesting directions for further research in the broader context of decision support in order to support users in decision-making. In sum, the thesis itself has presented concrete concepts and methods for the topics: *Data Quality in Recommender Systems* and *Disruptive Events in Service Systems* to contribute to the further development of decision support systems.

5 References

Abbasi, Ahmed; Sarker, Suprateek; Chiang, Roger H. L. (2016): Big data research in information systems: Toward an inclusive research agenda. In *JAIS* 17 (2), p. 3. DOI: 10.17705/1jais.00423.

Abu-Khzam, Faisal N.; Bazgan, Cristina; Haddad, Joyce El; Sikora, Florian (2015): On the Complexity of QoS-Aware Service Selection Problem. In Alistair Barros, Daniela Grigori, Nanjangud C. Narendra, Hoa Khanh Dam (Eds.): Service-Oriented Computing, vol. 9435. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 345–352.

Adomavicius, G.; Tuzhilin, A. (2005): Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions. In *IEEE Trans. Knowl. Data Eng.* 17 (6), pp. 734–749. DOI: 10.1109/TKDE.2005.99.

Alter, Steven (2012): Metamodel for service analysis and design based on an operational view of service and service systems. In *Service Science* 4 (3), pp. 218–235.

Anuradha, J. (2015): A brief introduction on Big Data 5Vs characteristics and Hadoop technology. In *Procedia Computer Science* 48, pp. 319–324.

Anwar, Khalid; Siddiqui, Jamshed; Sohail, Shahab Saquib (2020): Machine learning-based book recommender system: a survey and new perspectives. In *IJIIDS* 13 (2/3/4), Article 109457, p. 231. DOI: 10.1504/IJIIDS.2020.109457.

Bai, Bing; Fan, Yushun; Tan, Wei; Zhang, Jia (2020): DLTSR: A Deep Learning Framework for Recommendations of Long-Tail Web Services. In *IEEE Trans. Serv. Comput.* 13 (1), pp. 73–85. DOI: 10.1109/TSC.2017.2681666.

Batini, Carlo; Cappiello, Cinzia; Francalanci, Chiara; Maurino, Andrea (2009): Methodologies for data quality assessment and improvement. In *ACM Comput. Surv.* 41 (3), pp. 1–52. DOI: 10.1145/1541880.1541883.

Batini, Carlo; Scannapieco, Monica (2016): Data and Information Quality. Cham: Springer International Publishing.

Bearzotti, Lorena A.; Salomone, Enrique; Chiotti, Omar J. (2012): An autonomous multi-agent approach to supply chain event management. In *International Journal of Production Economics* 135 (1), pp. 468–478. DOI: 10.1016/j.ijpe.2011.08.023.

Betty Jane, J.; Ganesh, E. N. (2020): Big data and internet of things for smart data analytics using machine learning techniques. In *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI-2019)*. Springer, pp. 213–223.

Beverungen, Daniel; Müller, Oliver; Matzner, Martin; Mendling, Jan; Vom Brocke, Jan (2019): Conceptualizing smart service systems. In *Electron Markets* 29 (1), pp. 7–18. DOI: 10.1007/s12525-017-0270-5.

Böhmann, Tilo; Leimeister, Jan Marco; Möslein, Kathrin (2014): Service systems engineering. In *Business & Information Systems Engineering* 6 (2), pp. 73–79.

Bortlik, Michael; Heinrich, Bernd; Lohninger, Daniel (2023): Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime. In *Inf Syst Front*. DOI: 10.1007/s10796-023-10392-8.

Bortlik, Michael; Heinrich, Bernd; Mayer, Michael (2018): Multi User Context-Aware Service Selection for Mobile Environments. In *Business & Information Systems Engineering* 60 (5), pp. 415–430. DOI: 10.1007/s12599-018-0552-2.

Boudaa, Boudjemaa; Hammoudi, Slimane; Mebarki, Leila Amel; Bouguessa, Abdelkader; Chikh, Mohammed Amine (2017): An aspect-oriented model-driven approach for building adaptable context-aware service-based applications. In *Science of Computer Programming* 136, pp. 17–42.

Bunnell, Lawrence; Osei-Bryson, Kweku-Muata; Yoon, Victoria Y. (2019): RecSys Issues Ontology: A Knowledge Classification of Issues for Recommender Systems Researchers. In *Inf Syst Front* 97 (7), p. 667. DOI: 10.1007/s10796-019-09935-9.

Chen, Chien-Hsiung; Li, Shasha (2020): The effect of visual feedback types on the wait indicator interface of a mobile application. In *Displays* 61 (1), p. 101928. DOI: 10.1016/j.displa.2019.101928.

Cheng, Anping; Ma, Dongming; Qian, Hao; Pan, Younghwan (2023): The effects of mobile applications' passive and interactive loading screen types on waiting experience. In *Behaviour & Information Technology* 33 (9), pp. 1–12. DOI: 10.1080/0144929X.2023.2224901.

Da'u, Aminu; Salim, Naomie (2020): Recommendation system based on deep learning methods: a systematic review and new directions. In *Artif Intell Rev* 53 (4), pp. 2709–2748. DOI: 10.1007/s10462-019-09744-1.

Demirkan, Haluk; Bess, Charlie; Spohrer, Jim; Rayes, Ammar; Allen, Don; Moghaddam, Yassi (2015): Innovations with Smart Service Systems: Analytics, Big Data, Cognitive Assistance, and the Internet of Everything. In *CAIS* 37. DOI: 10.17705/1CAIS.03735.

Demirkan, Haluk; Delen, Dursun (2013): Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. In *Decision Support Systems* 55 (1), pp. 412–421.

Deng, Shuiguang; Huang, Longtao; Hu, Daning; Zhao, J. Leon; Wu, Zhaohui (2016): Mobility-Enabled Service Selection for Composite Services. In *IEEE Trans. Serv. Comput.* 9 (3), pp. 394–407. DOI: 10.1109/TSC.2014.2365799.

Di Napoli, Claudia; Ribino, Patrizia; Serino, Luca (2021): Correction to: Customisable assistive plans as dynamic composition of services with normed-QoS. In *J Ambient Intell Human Comput*. DOI: 10.1007/s12652-021-02978-4.

Dinh, Loan Thi Ngoc; Karmakar, Gour; Kamruzzaman, Joarder (2020): A survey on context awareness in big data analytics for business applications. In *Knowl Inf Syst* 62, pp. 3387–3415.

Edmunds, Angela; Morris, Anne (2000): The problem of information overload in business organisations: a review of the literature. In *International Journal of Information Management* 20 (1), pp. 17–28. DOI: 10.1016/S0268-4012(99)00051-1.

Fekih, Hela; Mtibaa, Sabri; Bouamama, Sadok (2019): The dynamic reconfiguration approach for fault-tolerance web service composition based on multi-level VCSOP. In *Procedia Computer Science* 159 (1), pp. 1527–1536. DOI: 10.1016/j.procs.2019.09.323.

Feldman, Michael; Even, Adir; Parmet, Yisrael (2018): A methodology for quantifying the effect of missing data on decision quality in classification problems. In *Communications in Statistics–Theory and Methods* 47 (11), pp. 2643–2663.

Flouris, Ioannis; Giatrakos, Nikos; Deligiannakis, Antonios; Garofalakis, Minos; Kamp, Michael; Mock, Michael (2017): Issues in complex event processing: Status and prospects in the big data era. In *Journal of Systems and Software* 127, pp. 217–236.

Gavalas, Damianos; Konstantopoulos, Charalampos; Mastakas, Konstantinos; Pantziou, Grammati (2014): Mobile recommender systems in tourism. In *Journal of network and computer applications* 39, pp. 319–333.

Gendreau, Michel; Potvin, Jean-Yves (2010): Handbook of Metaheuristics. Boston, MA: Springer US (146).

Georgantas, Nikolaos (2018): Service Oriented Computing in Mobile Environments: Abstractions and Mechanisms for Interoperability and Composition. Sorbonne Université.

Hariri, Reihaneh H.; Fredericks, Erik M.; Bowers, Kate M. (2019): Uncertainty in big data analytics: survey, opportunities, and challenges. In *Journal of Big Data* 6 (1), pp. 1–16.

Hashem, Ibrahim Abaker Targio; Yaqoob, Ibrar; Anuar, Nor Badrul; Mokhtar, Salimah; Gani, Abdullah; Khan, Samee Ullah (2015): The rise of "big data" on cloud computing: Review and open research issues. In *Information Systems* 47, pp. 98–115.

Heinrich, Bernd; Hopf, Marcus; Lohninger, Daniel; Schiller, Alexander; Szubartowicz, Michael (2021): Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. In *Electron Markets* 23 (2–3), p. 169. DOI: 10.1007/s12525-019-00366-7.

Heinrich, Bernd; Hopf, Marcus; Lohninger, Daniel; Schiller, Alexander; Szubartowicz, Michael (2022): Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems. In *Inf Syst Front* 24 (1), pp. 267–286. DOI: 10.1007/s10796-020-10071-y.

Heinrich, Bernd; Huber, Maximilian; Krapf, Thomas; Schiller, Alexander (2023): The Currency of Wiki Articles – A Language Model-based Approach. In *International conference on information systems*. Available online at https://aisel.aisnet.org/icis2023/dab_sc/dab_sc/14.

Heinrich, Bernd; Klier, Mathias (2011): Assessing data currency—a probabilistic approach. In *Journal of Information Science* 37 (1), pp. 86–100.

Heinrich, Bernd; Klier, Mathias; Lewerenz, Lars; Mayer, Michael (2015): Enhancing decision support in multi user service selection. In *36th International Conference on Information Systems (ICIS)*.

Heinrich, Bernd; Klier, Mathias; Schiller, Alexander; Wagner, Gerit (2018): Assessing data quality – A probability-based metric for semantic consistency. In *Decision Support Systems* 110, pp. 95–106. DOI: 10.1016/j.dss.2018.03.011.

Heinrich, Bernd; Lewerenz, Lars (2015): Decision support for the usage of mobile information services: A context-aware service selection approach that considers the effects of context interdependencies. In *Journal of decision systems* 24 (4), pp. 406–432. DOI: 10.1080/12460125.2015.1080498.

Heinrich, Bernd; Mayer, Michael (2018): Service selection in mobile environments: considering multiple users and context-awareness. In *Journal of decision systems* 27 (2), pp. 92–122. DOI: 10.1080/12460125.2018.1513223.

Heinrich, Bernd; Schön, Dominik (2015): Automated Planning of Context-aware Process Models. In *European conference on information systems*. DOI: 10.18151/7217352.

Hidri, Wafa; M'tir, Riadh Hadj; Saoud, Narjès Bellamine Ben; Ghedira-Guegan, Chirine (2019): A Meta-model for context-aware adaptive Business Process as a Service in collaborative cloud environment. In *Procedia Computer Science* 164, pp. 177–186.

Hifi, Mhand; Michrafy, Mustapha; Sbihi, Abdelkader (2004): Heuristic algorithms for the multiple-choice multidimensional knapsack problem. In *Journal of the Operational Research Society* 55 (12), pp. 1323–1332.

Hohenstein, Jess; Khan, Hani; Canfield, Kramer; Tung, Samuel; Perez Cano, Rocio (2016): Shorter Wait Times. In Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris, Juan Pablo Hourcade (Eds.): Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. San Jose California USA, 2016. New York, NY, USA: ACM, pp. 3084–3090.

Hohmann, Christoph; Posselt, Tim (2019): Design challenges for CPS-based service systems in industrial production and logistics. In *International Journal of Computer Integrated Manufacturing* 32 (4-5), pp. 329–339. DOI: 10.1080/0951192X.2018.1552795.

Hu, Shigang; Kumar, Akshi; Al-Turjman, Fadi; Gupta, Shivam; Seth, Simran; Shubham (2020): Reviewer Credibility and Sentiment Analysis Based User Profile Modelling for Online Product Recommendation. In *IEEE Access* 8, pp. 26172–26189. DOI: 10.1109/ACCESS.2020.2971087.

Idrissi, Nouhaila; Zellou, Ahmed (2020): A systematic literature review of sparsity issues in recommender systems. In *Soc. Netw. Anal. Min.* 10 (1), p. 5101. DOI: 10.1007/s13278-020-0626-2.

Ilarri, Sergio; Hermoso, Ramón; Trillo-Lado, Raquel; Rodr\'\ia del Carmen (2015): A review of the role of sensors in mobile context-aware recommendation systems. In *International Journal of Distributed Sensor Networks* 11 (11), p. 489264.

Jannach, Dietmar; Adomavicius, Gediminas (2016): Recommendations with a Purpose. In Shilad Sen, Werner Geyer (Eds.): Proceedings of the 10th ACM Conference on Recommender Systems. Boston, Massachusetts, USA. ACM. New York, NY, USA: Association for Computing Machinery, pp. 7–10.

Knight, Frank H. (1921): Risk, uncertainty, and profit. Library of Economics and Liberty. In *Acceso* 31, p. 2017.

Ko, Hyeyoung; Lee, Suyeon; Park, Yoonseo; Choi, Anna (2022): A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. In *Electronics* 11 (1), p. 141. DOI: 10.3390/electronics11010141.

Konstan, Joseph A.; Riedl, John (2012): Recommender systems. From algorithms to user experience. In *User Model User-Adap Inter* 22 (1-2), pp. 101–123. DOI: 10.1007/s11257-011-9112-x.

Koskela-Huotari, Kaisa; Patrício, Lia; Zhang, Jie; Karpen, Ingo Oswald; Sangiorgi, Daniela; Anderson, Laurel; Bogicevic, Vanja (2021): Service system transformation through service design: Linking analytical dimensions and service design approaches. In *Journal of Business Research* 136 (4), pp. 343–355. DOI: 10.1016/j.jbusres.2021.07.034.

Kum, Deuk Kyu (2020): Extended CEP Model for Effective Enterprise Systems Service Monitoring. In *KSII Transactions on Internet and Information Systems (TIIS)* 14 (2), pp. 807–825.

Li, Seth Siyuan; Karahanna, Elena (2015): Online recommendation systems in a B2C E-commerce context: a review and future directions. In *Journal of the Association for Information Systems* 16 (2), pp. 72–107.

Li, Shasha; Chen, Chien-Hsiung (2019): The effects of visual feedback designs on long wait time of mobile application user interface. In *Interacting with Computers* 31 (1), pp. 1–12. DOI: 10.1093/iwc/iwz001.

Li, Shasha; Chen, Chien-Hsiung; Lin, Zhongzhen (2022): Evaluating the impact of wait indicators on user visual imagery and speed perception in mobile application interfaces. In *International Journal of Industrial Ergonomics* 88 (2), p. 103280. DOI: 10.1016/j.ergon.2022.103280.

Li, Zhi; Zhao, Hongke; Liu, Qi; Huang, Zhenya; Mei, Tao; Chen, Enhong (2018): Learning from History and Present. In Yike Guo, Faisal Farooq (Eds.): Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London United Kingdom, 2018. New York, NY, USA: ACM, pp. 1734–1743.

Lika, Blerina; Kolomvatsos, Kostas; Hadjiefthymiades, Stathes (2014): Facing the cold start problem in recommender systems. In *Expert Systems with Applications* 41 (4), pp. 2065–2073. DOI: 10.1016/j.eswa.2013.09.005.

Liu, Shaofeng; Zaraté, Pascale (2014): Knowledge based decision support systems: A survey on technologies and application domains. In *Group Decision and Negotiation*. A Process-Oriented View: Joint INFORMS-GDN and EWG-DSS International Conference, GDN 2014. Springer, pp. 62–72.

Lohninger, Daniel (2024): User-based Event Handling Strategy for Multi User Context-Aware Service Systems in Mobile Environments. Working Paper, University of Regensburg.

Madadipouya, Kasra; Chelliah, Sivananthan (2017): A literature review on recommender systems algorithms, techniques and evaluations. In *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* 8 (2), pp. 109–124.

Mauro, Andrea de; Greco, Marco; Grimaldi, Michele (2015): What is big data? A consensual definition and a review of key research topics. In *AIP conference proceedings*, vol. 1644. American Institute of Physics, pp. 97–104.

Mayer, Michael (2017): Multi-user Service Re-Selection: React Dynamically to Events occuring at process Execution. In *25th European Conference on Information Systems (ECIS)*.

Mayer, Michael (2019): Quality-of-Service-Aware Service Selection in Mobile Environments. Available online at https://epub.uni-regensburg.de/38182/.

Mehta, Rachana; Rana, Keyur (2017): A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems*, Computing and IT Applications (CSCITA), pp. 269–274.

Moghaddam, Mahboobeh; Davis, Joseph G. (2014): Service Selection in Web Service Composition: A Comparative Review of Existing Approaches. In Athman Bouguettaya, Quan Z. Sheng, Florian Daniel (Eds.): Web Services Foundations, vol. 33. New York, NY: Springer New York, pp. 321–346.

Mu, Ruihui (2018): A Survey of Recommender Systems Based on Deep Learning. In *IEEE Access* 6, pp. 69009–69022. DOI: 10.1109/ACCESS.2018.2880197.

Muhammad, Syed Sardar; Dey, Bidit Lal; Weerakkody, Vishanth (2018): Analysis of Factors that Influence Customers' Willingness to Leave Big Data Digital Footprints on Social Media: A Systematic Review of Literature. In *Inf Syst Front* 20 (3), pp. 559–576. DOI: 10.1007/s10796-017-9802-y.

Neville, Karen; O'Riordan, Sheila; Pope, Andrew; Rauner, Marion; Rochford, Maria; Madden, Martina et al. (2016): Towards the development of a decision support system for multi-agency decision-making during cross-border emergencies. In *Journal of decision systems* 25 (sup1), pp. 381–396.

Patel, Krupa; Patel, Hiren B. (2020): A state-of-the-art survey on recommendation system and prospective extensions. In *Computers and Electronics in Agriculture* 178, p. 105779.

Phillips-Wren, Gloria; Adya, Monica (2020): Decision making under stress: the role of information overload, time pressure, complexity, and uncertainty. In *Journal of decision systems* 29 (sup1), pp. 213–225. DOI: 10.1080/12460125.2020.1768680.

Picault, Jérome; Ribiere, Myriam; Bonnefoy, David; Mercer, Kevin (2011): How to get the Recommender out of the Lab? In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 333–365.

Pipino, Leo L.; Lee, Yang W.; Wang, Richard Y. (2002): Data quality assessment. In *Commun. ACM* 45 (4), pp. 211–218. DOI: 10.1145/505248.506010.

Power, Daniel J.; Sharda, Ramesh; Burstein, Frada (2015): Decision support systems. Hoboken, New Jersey, USA: John Wiley & Sons, Ltd.

Ray, Biswarup; Garain, Avishek; Sarkar, Ram (2021): An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. In *Applied Soft Computing* 98 (6088), p. 106935. DOI: 10.1016/j.asoc.2020.106935.

Rötzel, Peter Gordon; Fehrenbacher, Dennis D. (2019): On the Role of Information Overload in Information Systems (IS) Success: Empirical Evidence from Decision Support Systems. In *International conference on information systems*.

Saha, Barna; Srivastava, Divesh (2014): Data quality: The other face of big data. In *2014 IEEE 30th international conference on data engineering*. IEEE, pp. 1294–1297.

Saxena, Deepak; Lamest, Markus (2018): Information overload and coping strategies in the big data context: Evidence from the hospitality sector. In *Journal of Information Science* 44 (3), pp. 287–297.

Schnabel, Tobias; Bennett, Paul N.; Joachims, Thorsten (2018): Improving recommender systems beyond the algorithm. In *arXiv preprint arXiv:1802.07578*.

Smirnova, Elena; Vasile, Flavian (2017): Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In Balázs Hidasi, Alexandros Karatzoglou, Oren Sar-Shalom, Sander Dieleman, Bracha Shapira, Domonkos Tikk (Eds.): Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems. Como Italy, 2017. New York, NY, USA: ACM, pp. 2–9.

Srifi, Mehdi; Oussous, Ahmed; Ait Lahcen, Ayoub; Mouline, Salma (2020): Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey. In *Information* 11 (6), p. 317. DOI: 10.3390/info11060317.

Statista (2017): 229 Millionen Produkte auf Amazon.de. Available online at https://de.statista.com/infografik/7849/bei-amazon-deutschland-gelistete-produkte/, updated on 2017, checked on 4/4/2024.

Statista (2019): Size of mobile professional services market by segment worldwide in 2014 and 2018. Edited by Statista. Available online at https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/, checked on 3/10/2023.

Statista (2023): Volumen der jährlich generierten/replizierten digitalen Datenmenge weltweit von 2010 bis 2022 und Prognose bis 2027. Edited by Statista. Available online at https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/, checked on 4/3/2024.

TripAdvisor LLC (2024): About Tripadvisor. Available online at https://tripadvisor.mediaroom.com/us-about-us, checked on 4/2/2024.

Vanaja, R.; Mukherjee, Saswati (2019): Novel Wrapper-Based Feature Selection for Efficient Clinical Decision Support System. In Leman Akoglu, Emilio Ferrara, Mallayya Deivamani, Ricardo Baeza-Yates, Palanisamy Yogesh (Eds.): Advances in Data Science, vol. 941. Third International Conference on Intelligent Information Technologies. Chennai, India. ICIIT Conference Group. Singapore: Springer Singapore (Communications in Computer and Information Science, 941), pp. 113–129.

Ventola, C. Lee (2014): Mobile devices and apps for health care professionals: uses and benefits. In *Pharmacy and Therapeutics* 39 (5), p. 356.

Verma, Rohit; Srivastava, Abhishek (2018): A dynamic web service registry framework for mobile environments. In *Peer-to-peer Networking and Applications* 11, pp. 409–430.

Wand, Yair; Wang, Richard Y. (1996): Anchoring data quality dimensions in ontological foundations. In *Commun. ACM* 39 (11), pp. 86–95. DOI: 10.1145/240455.240479.

Wang, Hongbing; Peng, Shunshun; Yu, Qi (2019): A parallel refined probabilistic approach for QoS-aware service composition. In *Future Generation Computer Systems* 98 (02), pp. 609–626. DOI: 10.1016/j.future.2019.03.053.

Woodall, Philip; Borek, Alexander; Gao, Jing; Oberhofer, Martin; Koronios, Andy (2015): An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics. In Richard Wang (Ed.): Big data. 19th International Conference on Information Quality. Xi'an, China. International Conference on Information Quality; ICIQ. Red Hook, NY: Curran (Management and data quality), pp. 24–33.

Xu, Hu; Liu, Bing; Shu, Lei; Yu, Philip (2019): BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2324–2335. Available online at https://www.aclweb.org/anthology/N19-1242.

Ye, Hua; Kankanhalli, Atreyi (2020): Value Cocreation for Service Innovation: Examining the Relationships between Service Innovativeness, Customer Participation, and Mobile App Performance. In *JAIS*, pp. 294–312. DOI: 10.17705/1jais.00602.

Zhang, Dongsong; Adipat, Boonlit; Mowafi, Yaser (2009): User-centered context-aware mobile applications—The next generation of personal mobile computing. In *Communications of the Association for Information Systems* 24 (1), p. 3.