Quality-of-Service-Aware Service Selection in Mobile Environments

Dissertation zur Erlangung des Grades eines

Doktors der Wirtschaftswissenschaft

eingereicht an der Fakultät für Wirtschaftswissenschaften

der Universität Regensburg

vorgelegt von:

Michael Mayer, M.Sc. with Honors

Berichterstatter:

Prof. Dr. Bernd Heinrich

Prof. Dr. Mathias Klier

Tag der Disputation: 13.12.2018

Dedicated to my family and friends

And especially, to Andrea

Acknowledgements

I would like to express my deep gratitude to Prof. Dr. Bernd Heinrich and Prof. Dr. Mathias Klier for their guidance and great support during the last four years. In addition, I would also like to thank my further co-authors Lars Lewerenz and Michael Bortlik for their contribution, ideas and constructive feedback during our collaboration. Furthermore, I want to thank my colleagues and the many students who worked on the topic with me and thus supported my research.

Finally, I would like to thank my family and friends for their ongoing support and encouragement.

Michael Mayer

Contents

Explanatory note: The references for the papers are listed at the end of each paper, the references for Section 1 and 6 are listed at the end of the thesis (Section 7).

List of Figures

List of Abbreviations

AHP	Analytic Hierarchy Process
CA	context-aware
ComWSCs	common world-state-service-object combinations
FC	Focus Class
IUR	Inter-User-Request
MMKP	multi-choice multidimensional knapsack problem
NCA	non-context-aware
NFP	non-functional property
QoS	Quality-of-Service
RQ	research question
SAW	Simple Additive Weighting
SLA	service-level agreement
WSC	world-state-service-object combination

1 Introduction

The introduction section is structured as follows: First, the motivation for the work is presented, which is followed by a discussion of the considered research questions. Subsequently, the research methodology of the thesis is briefly described. And finally, the structure and contents of the study are outlined.

1.1 Motivation

The last decade is characterized by the rise of mobile technologies (UMTS, LTE, WLAN, Bluetooth, SMS, etc.) and devices (notebooks, tablets, mobile phones, smart watches, etc.). In this rise, mobiles phones have played a crucial role because they paved the way for mobile pervasion among the public. Although the first mobile phone calls (via phones integrated in cars) were possible in the middle of the 20th century (cf. Messmer 2008), it was the early years of this millennium that saw a breakthrough in terms of the large scale introduction of third-generation digital mobile communications networks (3G) and the possibility of mobile phones connecting to data networks such as WLAN, WiMAX, Bluetooth, and NFC (cf. Kamal Bashah et al. 2012; Temple 2014). Today, more than 63% of the global population uses a mobile phone (Statista 2017a). Furthermore, according to Gartner (2017), the number of mobile phones shipped annually is expected to remain near 2 billion constantly from 2017 to 2019.

In addition, the emergence of mobile technologies has led to the development and provision of mobile services. This has resulted in the rapid growth of the mobile service/application market. For instance, the app store hosted by Apple counted 0.3 million applications in 2010, whereas in 2016, the number of available applications exceeded 2 million (Statista 2017b). Mobile applications could refer to transaction (e.g., banking, shopping, and auctions), communication (e.g., email, and instant messaging), and information services (e.g., navigation, traffic, and tourist guides).

As a consequence, users nowadays find themselves in a mobile environment, with (almost) unlimited access to information and services from anywhere through the Internet and can connect to other people at any time (cf. Deng et al. 2016; Newman 2015). Furthermore, mobile devices allow for instant communication and reaction (e.g., through messenger or video chat) and by this foster user interaction and collaboration. In this respect, modern mobile devices offer the opportunity to select the services or information that best fit to a user's current situation. More precisely, these devices possess a variety of physical sensors that allow the capture of current user and environment contexts, such as their current location (i.e., GPS position), time of day, weather conditions (e.g., temperature and humidity), and even their medical condition (e.g., heartbeat, respiration, and perspiration) (cf. Hu et al. 2012; Lin et al. 2012; Raij et al. 2011; Yu and Reiff-Marganiec 2009b). Dey (2001) defines context in general as follows (p. 5):

> *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the*

> *interaction between a user and an application, including the user and applications themselves."*

In this regard, mobile information services support users in retrieving context and non-context information, such as about the current traffic situation, public transport options, flight connections, weather forecast, and hospital patients, as well as about real-world entities, such as sights, museums, and restaurants (cf. Deng et al. 2016; Heinrich and Lewerenz 2015; Ventola 2014). For example, healthcare professionals in hospitals use hospital information services on mobile devices to gather information about patients (e.g., medical data, previous diseases, and personal circumstances), and also information about operating and treatment rooms with respect to context information such as time schedule and location (cf. Boruff and Storie 2014; Marynissen and Demeulemeester 2016; Ventola 2014). This information can then be used for assigning healthcare professionals with certain skills to patients subject to their special needs, to maximize treatment quality and minimize overall duration (i.e., treatment period and waiting time) for patients. Thus, considering context information facilitates the detection of nearby available rooms for treatment or surgery. Furthermore, for certain treatments, healthcare professionals must conduct multiple actions in a row (cf. Măruşter et al. 2002; Marynissen and Demeulemeester 2018; Vries et al. 1999) while for some of these actions it could be more beneficial when they are conducted together by several healthcare professionals who possess various skills (e.g., complex surgery), thereby resulting in a process with multiple participating healthcare professionals.

A further example of the application of mobile information services is several users planning a joint city day trip. Here, the users could utilize information retrieved about real-world entities for their planning. Such a trip constitutes a process with multiple participating users and may encompass actions such as visiting a museum, having lunch, visiting a sight, and going to a café. For each action, mobile information services (e.g., Yelp, TripAdvisor, Google Places) can help locate available alternatives that differ only in attributes such as price, average length of stay (i.e., duration), or recommendations published by previous visitors. In addition, context information can be used to more effectively support the users in their decisions, for instance, in terms of business hours of the real-world entities or distance between real-world entities of succeeding actions (e.g., the distance between a museum and a restaurant). Moreover, because multiple users are participating in the same trip, some users want to or must conduct certain actions together. For example, one user might prefer to have lunch with some of the other users on that trip.

However, decision-makers (e.g., mobile users) attempting to determine the optimal solution for such processes (i.e., optimal city trip tours or optimal healthcare professional allocations) – meaning the best alternative for each action and each participating user – are confronted with several challenges, as shown by means of the city trip example: First, each user most likely has his or her own preferences and requirements regarding attributes such as price and duration, which all must be considered. Furthermore, for each action of the day trip, a huge number of alternatives probably exist (e.g., Yelp lists almost 8,000 restaurants[1] and 600 museums[2] in Berlin, Germany). Thus, users might face difficulties selecting the optimal alternatives because of an information overload problem (Zhang et al. 2009). Second, taking multiple users into account may require the coordination of their actions because of potential dependencies among different users' tours, which, for example, is the case when users prefer to conduct certain actions together. This turns the almost sophisticated decision problem at hand into a problem of high complexity. The problem complexity is increased further when considering context information, because this causes dependencies among different actions of a user that must be taken into account. For instance, the distance to cover by a user to reach a certain restaurant depends on the location of the previously visited museum. In conclusion, it might be impossible for a user to determine an optimal city trip tour for all users, making decision support by an information system necessary. Because the available alternatives for each action of the process can be denoted as (information) service objects (cf. Dannewitz et al. 2008; Heinrich and Lewerenz 2015; Hinkelmann et al. 2013), the decision problem at hand is a *Quality-of-Service (QoS)-aware service selection* problem, which is described as follows.

QoS-aware service selection problems can be originally found in the field of service-oriented computing (cf. Barry 2012; Weinhardt et al. 2011) where they refer to the selection of suitable (web) services (i.e., modular designed applications) to realize complex business processes in service-oriented architectural settings (cf., e.g., Alrifai et al. 2012; Alrifai and Risse 2009; Ardagna and Mirandola 2010; Ardagna and Pernici 2007; Canfora et al. 2005; Yu et al. 2007; Zeng et al. 2004). Similarly, QoS-aware service selection problems in mobile environments refer to processes that encompass several actions, wherein each action can be realized by multiple alternative service objects that differ only in their non-functional properties (NFP), represented by QoS attributes such as price, duration (or response time), and recommendations of other users (e.g., by ratings). In the basic case of a single user process and without considering context information, the decision problem can be formulated as follows: What is the optimal service (object) composition based on the user's preferences (i.e., target weights) and requirements in the sense of global end-to-end constraints (e.g., the upper limit for overall budget) regarding these NFP?

In general, a QoS-aware service selection problem can be understood as a knapsack problem, which is a combinatorial optimization problem (cf. Alrifai et al. 2012; Ardagna and Pernici 2006; Cao et al. 2007; Jaeger et al. 2005; Lin et al. 2011; Strunk 2010; Yu et al. 2007). Knapsack optimization models are also used for solving capacity planning problems in fields such as manufacturing, healthcare and network systems, production planning problems, capital budgeting, cargo loading problems, and cutting stock problems (cf. Bretthauer and Shetty 2002; Camargo et al. 2012; Lai and Barkan 2011; Martello and Toth 1987). To be exact, the basic QoS-aware service selection problem is a multi-choice, multidimensional knapsack problem (MMKP) (cf. Moser et al. 1997; Strunk 2010): The available items (= service objects) classified in multiple groups (= actions) are characterized by a specific value (= utility) where multiple

[1] https://www.yelp.com/search?find_desc=Restaurants&find_loc=Berlin,+Germany, accessed August 2018
[2] https://www.yelp.com/search?find_desc=Museums&find_loc=Berlin,+Germany, accessed August 2018

resources (e.g., duration and budget) constrain the knapsack (= user's service composition) (Ardagna and Pernici 2006).

Referring to the aforementioned city trip example, the alternative service objects for each action of the process are characterized by the individual values for certain NFP (e.g., price, duration, or recommendation) of the corresponding real-world entities. Based on all users' preferences and requirements regarding these NFP, the optimal set of service objects for each user can be determined by applying a suitable QoS-aware service selection approach. Similarly, QoS-aware service selection can be used to determine the optimal (mobile) service objects for other processes in mobile environments, such as the aforementioned example of healthcare professionals.

The purpose of this thesis is to develop novel concepts and optimization approaches for service selection regarding processes with multiple users and context information, focusing on scenarios in mobile environments. In this respect, a sophisticated multi user context-aware service selection approach must be able to deal with dependencies among different users' service compositions, which result from the consideration of multiple users, as well as dependencies within a user's service composition, which result from the consideration of context information. These approaches are expected to provide suitable support for decision-makers, such as mobile users.

1.2 Research Questions

This section introduces the three basic research questions of this thesis. These refer to the dimensions of *point in time of service selection* and *type of considered dependencies* (to deal with multiple users and context information), as illustrated in Figure 1. More precisely, the first research question (RQ 1) addresses QoS-aware service selection for multi user processes at planning time, whereas the second (RQ 2) targets the time of process execution. The third research question (RQ 3) then deals with dependencies resulting from both multiple users and context information.

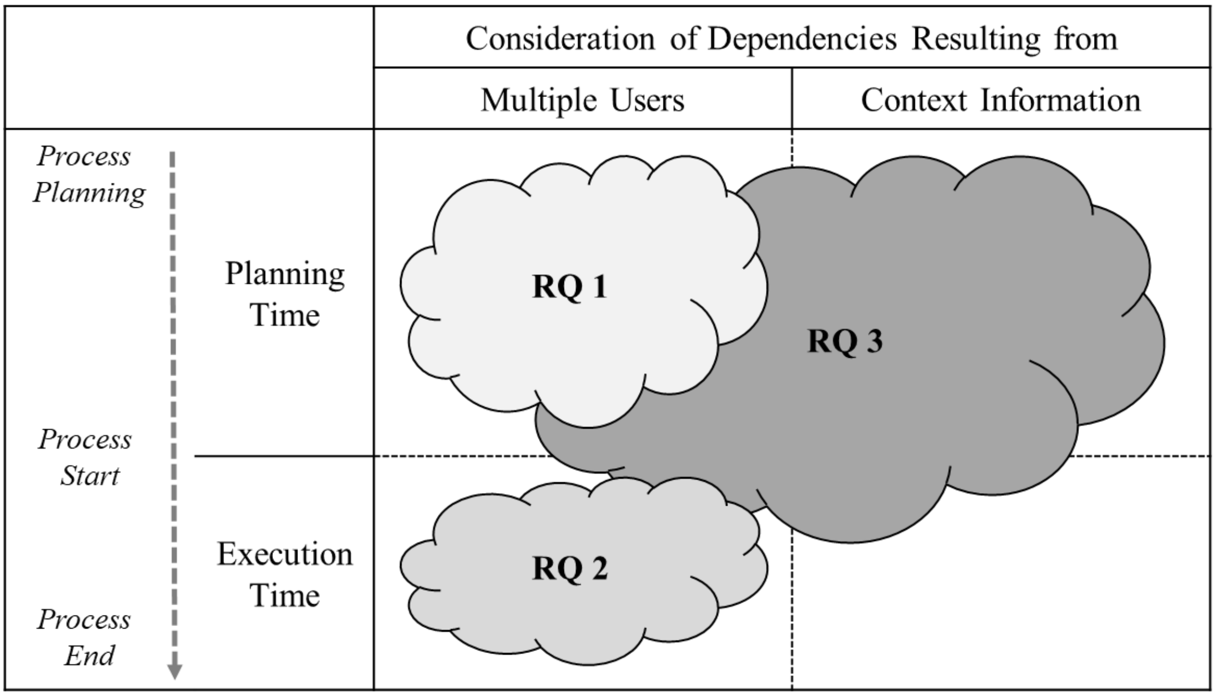

Figure 1. Focus of the Research Questions

When considering processes with multiple participating users in QoS-aware service selection, the preferences and requirements regarding the NFP of each user must be taken into account. Furthermore, there could be situations in which the simultaneous use of a certain service object is limited, for example, in the case of capacity limits of service objects (e.g., a restricted number of available seats in a restaurant) (cf., e.g., He et al. 2012; Kang et al. 2011; Zhu et al. 2017), or in which the mutual use of a specific service object is mandatory (cf., e.g., Benouaret et al. 2012; Wanchun et al. 2011; Wang et al. 2010). Apart from these hard restrictions, some users may have requests that refer to other users because of interpersonal relationships (cf. Heider 1958; Schutte et al. 2001). Such requests could entail a user who prefers using a certain service object or conducting a certain action together with other users, or even a user who does not want to use a certain service object with other users. These user-defined requests can be denoted as *Inter-User-Requests* (IUR) because they usually affect the optimal service composition of each user.

As a result, such IUR cause dependencies between the service objects of different users. Thus, the actual utility of a service object for a user depends on whether one or more certain other service objects are selected or not. These dependencies could also be of temporal nature (e.g., a user requests to use a certain service object simultaneously together with other users), which additionally requires the temporal coordination of the users' actions. Therefore, the first research question is specified as follows:

- **RQ 1:** How to define and model user requests that refer to other users (which means IUR) in a methodically well-founded way and how to integrate them in a multi user service selection approach?

Usually, service objects for a process are selected at the time of planning, which means before the execution of the process begins. Thus, the aim of an ex-ante multi user service selection approach is to determine the optimal service composition for each user based on the service objects available at that time as well as their NFP values.

However, particularly in a mobile environment, these determined service compositions may not be optimal when it comes to the execution of the process: Service objects selected at planning time may, for example, take (significantly) longer than expected, and may in reality not be available or fail during their execution (cf. Canfora et al. 2008; Sheng et al. 2014; Zheng et al. 2014). A reason for this is the special characteristics of mobile environments because, for instance, constant mobility of the users may lead to non-predetermined service performance (Deng et al. 2016). An example is the response time of service objects, which can differ depending on the time and location of invocation (cf. Wang et al. 2015; Zheng et al. 2014).

In terms of QoS-aware service selection for processes with multiple participating users, potential events occurring at execution time could be:

- Actual NFP values considerably differing from those determined at planning time
- Failing or newly available service objects
- Users leaving or joining the process

The occurrence of such events can have a significant impact on the utility or feasibility of an ex-ante planned service composition as well as on the realization of planned IUR. In sum, multi user service selection must somehow deal with disruptive events.

Various strategies exist for how decision-makers can react to (potential) process disruptions. For example, proactive strategies include anticipating possible disruptions, building robust models, and employing rule-based supervision (cf. Ardagna et al. 2011; Pinedo 2005). Proactive strategies in QoS-aware service selection could be considering potential service failures already at planning time (cf. Heinrich et al. 2015; Yu and Lin 2005) or following a fault-tolerant strategy (cf. Shen et al. 2012b; Stein et al. 2009; Zheng and Lyu 2010).

However, because disruptive events could still occur at execution time despite the deployed proactive strategies, reactive disruption management in terms of dynamic service re-selection is additionally required. Therefore, this work examines the following research question:

➢ **RQ 2:** How to design a multi user service re-selection approach that is capable of handling disruptive events occurring at execution time?

As described in Section 1.1, considering context information can significantly enhance the decision support provided for multi user processes. For instance, referring to the city trip example, by considering the business hours of restaurants and museums, a user will most likely not find a closed restaurant upon arrival. In addition to business hours (i.e., daytime-dependent availability of service objects), several other types of context information can be regarded in service selection, for example, price discounts on a certain set of service objects (cf., e.g., Xu and Jennings 2010; Yu and Reiff-Marganiec 2009a; Zhang et al. 2013a), distance between different service providers (cf., e.g., Heinrich and Lewerenz 2015; Zhang et al. 2013b) or devices (cf., e.g., Shen et al. 2012a), provider relations (cf., e.g., Zhang et al. 2013a), and user favorites (cf., e.g., Lewerenz 2015).

When considering context information in service selection, its dynamic characteristic must be addressed (cf. Damián-Reyes et al. 2011; Kirsch-Pinheiro et al. 2008; Vanrompay et al. 2009), which leads to the following three effects (cf. Heinrich and Lewerenz 2015):

(1) The actual value regarding a specific piece of context information for a certain service object depends on a user's initial context (e.g., starting time and location) and previously selected service objects. This means that the actual manifestation (value) of a context information can differ for different service compositions. For instance, the value for the "distance" of a certain restaurant in the city trip example depends not only on its location but also on the location of the museum visited by the user before.
(2) As a consequence, the utility of a certain service object usually differs for different service compositions.
(3) Furthermore, the selection of a service object can affect the feasibility of other service objects (e.g., regarding business hours, a certain restaurant selected for lunch may already be closed if too much time was spent in a museum prior to arriving).

Thus, these effects could cause dependencies between different service objects of a user. Therefore, considering both multiple users and context information in service selection requires dealing with dependencies within one user's service composition and among different users' service compositions. This results in the third research question of this thesis:

- **RQ 3:** How to model and consider dependencies resulting from both multiple users and context information in QoS-aware service selection?

1.3 Research Methodology

In the following, the research methodology applied to address the aforementioned three research questions is briefly discussed.

According to Bertrand and Fransoo (2002), quantitative model-based research "is based on the assumption that we can build objective models (…) that can capture (part of) the decision-making problems" (p. 249). Here, the authors distinguish one dimension through empirical versus axiomatic research, and another dimension through descriptive versus normative research. Whereas empirical quantitative model-based research focuses on the fit between a model defined to represent the reality and empirical findings or measurements, axiomatic research rather aims to determine solutions for existing problems. Furthermore, whereas descriptive research is considered to study a procedure or progress, normative research aims to contribute to a problem solution.

Because the purpose of this thesis is the development of novel concepts and optimization approaches for QoS-aware service selection problems in mobile environments, this work follows an axiomatic normative model-based quantitative research approach, through utilizing analytical and heuristic techniques (cf. Bertrand and Fransoo 2002; Meredith et al. 1989). Hence, the focus lies in the research phases of *conceptualization*, *modeling*, and *model solving*, as specified by Mitroff et al. (1974) (cf. Figure 2). Moreover, the evaluation methods applied in this work to measure the quality of the proposed models include lab and simulation experiments (both include real-world data) as well as benchmarking (i.e., comparing developed models with existing ones).

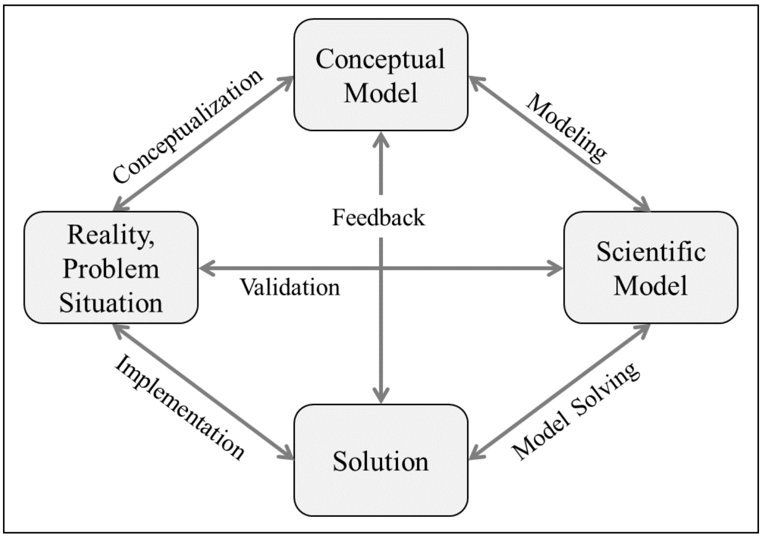

Figure 2. Phases in Normative Model-based Research (cf. Mitroff et al. 1974)

1.4 Thesis Content and Structure

This thesis consists of four papers, which address the three previously specified research questions:

- Paper 1: Enhancing Decision Support in Multi User Service Selection (RQ 1)
- Paper 2: Multi-User Service Re-Selection: React Dynamically to Events Occurring at Process Execution (RQ 2)
- Paper 3: Service Selection in Mobile Environments: Considering Multiple Users and Context-Awareness (RQ 3)
- Paper 4: Multi User Context-Aware Service Selection for Mobile Environments – A Heuristic Technique (RQ 3)

Figure 3 illustrates the focal points of each paper regarding the dimensions of *point in time of service selection* and *type of considered dependencies*.

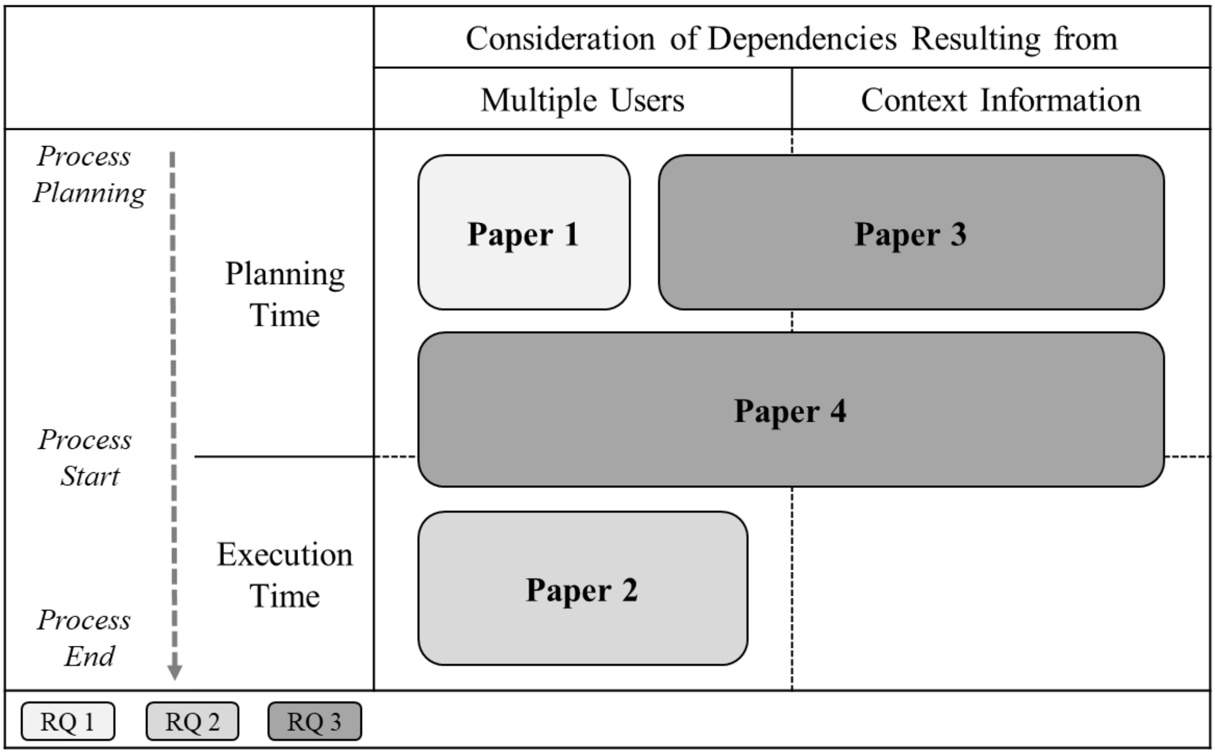

Figure 3. Focal Points of the Four Papers

Paper 1 – which addresses RQ 1 – develops a service selection approach enabling the consideration of multiple users with their individual preferences and requirements regarding the NFP as well as IUR. Therefore, the paper proposes concepts to define and model various types of IUR as well as to enable the temporal coordination of users. Subsequently, a knapsack optimization model is presented that integrates these concepts and allows the consideration of dependencies resulting from multiple users and IUR. Furthermore, the correctness, practical applicability, and performance of the approach is evaluated (e.g., by means of a lab experiment). This approach addresses multi user service selection at planning time, which means at the time the process is planned, and thus before its execution (for details about automated process planning see Heinrich et al. 2012; Henneberger et al. 2008; Hoffmann et al. 2009).

Based on the concepts and optimization model developed in Paper 1, Paper 2 presents an approach for multi user service re-selection that allows reacting dynamically to disruptive events occurring at process execution time (RQ 2). More precisely, it provides a novel optimization model that is able to consider dependencies caused by multiple users and IUR. This model also incorporates a continuous time concept required for the temporal coordination of the users at execution time. Thus, the approach enables provision of the optimal feasible solution for all users and the remaining part of the process after process disruption, which is demonstrated through an efficacy evaluation of the approach.

Papers 3 and 4 address RQ 3, and thus, they consider both multiple users and context information in QoS-aware service selection. First, Paper 3 identifies and categorizes various types of IUR and context information, and subsequently provides a unified modeling concept for dependencies resulting from multiple users and context information. Based on this, a stateful and a stateless optimization model for multi user context-aware service selection are presented and evaluated. The approaches proposed in Paper 3 (as well as in Papers 1 and 2) are exact service selection approaches, which means they apply exact solving methods (e.g., integer programming) to determine the optimal service compositions. By contrast, Paper 4 focuses on the development of a heuristic technique for tackling the computation time issues that come with exact approaches, caused by the general NP-hardness of the service selection problem (cf. Abu-Khzam et al. 2015). The presented heuristic technique comprises two stages and is able to consider dependencies resulting from context information, multiple users, and the simultaneous mandatory use of the same service object by several users.

The remainder of this thesis is structured as follows: Sections 2, 3, 4, and 5 present the four abovementioned papers. The thesis concludes with a summary of the major findings, a discussion of the limitations of the work, and suggestions for possible further research.

2 Paper 1: Enhancing Decision Support in Multi User Service Selection

Status	Published	Full Citation
Accepted	12/2015	Heinrich, B., Klier, M., Lewerenz, L., and Mayer, M. 2015. "Enhancing Decision Support in Multi User Service Selection," in *36th International Conference on Information Systems (ICIS)*, Fort Worth, USA.

Post publication changes:

- The format of the keywords was changed for consistency reasons
- Section numbering was added for consistency reasons
- In the whole paper, the format of the references was changed for consistency reasons
- In the whole paper, the comma setting for "e.g.", "i.e." and "cf." was adjusted for consistency reasons
- "modelling" changed to "modeling" in Section 3.3, 4.3 and 5
- "interdependencies" changed to "dependencies" for consistency reasons in Section 4.3
- "SODSS" changed to "service-oriented decision support systems" in Section 5
- "i.e." changed to "that means" in Section 2.1 and 4.3 for consistency reasons

----

Abstract

In service-oriented systems, the execution of processes can be supported by composing a variety of different services. In this context, an important research question concerns the selection of the optimal services while taking multiple users and their individual goals into account. Existing multi user service selection approaches focus on restrictions like fixed capacity restrictions of services. However, due to inter-user relations there may also be user requests that refer to other users, like for example that some users may prefer to conduct certain services together. Such Inter-User-Requests (IUR) – have not been addressed in research yet. To address this issue, we propose a novel multi user service selection approach taking into account IUR. We evaluate our approach with respect to correctness and performance. In addition, we examine the practical applicability by means of a real-world example and show that considering IUR in multi user service selection can considerably enhance decision support.

Keywords: Decision support, Service-oriented systems, Service selection, Multiple users

1 Introduction

In service-oriented systems, the execution of processes can be supported by composing a variety of different services. In this context, Quality-of-Service (QoS)-aware service selection is a widely known and discussed problem (cf. Alrifai et al. 2012; Ardagna and Mirandola 2010;

Ardagna and Pernici 2007; Canfora et al. 2008; Yu et al. 2007; Zeng et al. 2004). In a situation with a set of functional equivalent services – referred to as a service class – for each action of a process, non-functional properties (NFP) of services – represented by QoS attributes (e.g., price, response time, availability) – become the main decision criteria to select a suitable service composition. To provide decision support, existing approaches usually map the NFP onto a single utility value, while taking the preferences of the user concerning the different NFP into account. On this basis, the optimal service composition is determined by maximizing the overall utility of the included services, while satisfying global end-to-end constraints for the QoS attributes (e.g., an upper limit concerning the end-to-end price).

We argue that QoS-aware service selection can also be used in the context of service-oriented decision support systems (cf. Delen and Demirkan 2013; Demirkan and Delen 2013; Dong and Srinivasan 2013; Vescoukis et al. 2012). Indeed, processes from various domains such as logistics (cf. Tao et al. 2010), crisis management (cf. Vescoukis et al. 2012), or tourism (cf. Gavalas et al. 2014) are beneficiaries of service-oriented decision support systems, as the execution of each action can be supported by services which store, provide and subsequently analyze information relevant to the action. More precisely, the provided information can be understood as an information respectively service object (cf. Dannewitz et al. 2008; Hinkelmann et al. 2013) representing a real-life entity which is characterized by NFP (cf. O'Sullivan et al. 2002). Focusing, for instance, on the tourism domain, the information services *Yelp, TripAdvisor*, and *Google Places* can be used to support the execution of the action "*visiting museum*" by providing feasible service objects (e.g., *museum a, museum b,* etc.) in combination with their respective NFP (e.g., entrance fees, durations, recommendation values, etc.). Hence, analyzing and selecting the provided information (e.g., service objects) using QoS-aware service selection approaches (cf. Alrifai et al. 2012; Ardagna and Mirandola 2010; Ardagna and Pernici 2007; Canfora et al. 2008; Yu et al. 2007) can offer a promising means to support decision making.

In service-oriented decision support systems, the analyzed information can be provided cross-platform (e.g., laptop, desktop PC, mobile devices, etc.), which makes it possible to support processes with multiple participating users. Especially in the light of the emerging technology of mobile devices (e.g., smartphones and tablets) (cf. Google 2013; Picoto et al. 2014), adequate decision support for multi user processes becomes more and more important. Examples can be found in the coordination of field workers in engine repairing, relief field workers in disaster management (cf. Fajardo and Oppus 2009; Kartiwi and Gunawan 2013), fleets in forwarding companies, the actions of fire workers in emergency situations (cf. Monares et al. 2011), field health workers (cf. DeRenzi et al. 2011) or in the field of tourism (cf. Nagata et al. 2006). In the latter, a comprehensible use case for the support of multi user processes is a city day trip (cf. Figure 1). Here, it is likely that the participating users conduct some actions (e.g., Dinner, Sight, Museum, etc.) together (e.g., user 1 visits the 'Hofbraeuhaus' for dinner together with user 2) whereas other actions are rather conducted alone (e.g., visiting different museums due to personal predilections). A similar example can be found in the field of emergency situations/disaster management. Here, situations can exist where the conduction of certain actions (e.g., free persons trapped in cars/buildings) of the process is more beneficial but not mandatory, when certain users (e.g., firefighters with common professional experiences) conduct these actions together.

To adequately support multi user processes, a service selection approach must be capable of taking multiple users and their preferences and requirements (in terms of global end-to-end constraints) regarding the NFP into account. Literature already provides first approaches (cf. Jin et al. 2012a; Kang et al. 2011; Wang et al. 2010) which aim to maximize the accumulated utility over all participating users while considering restrictions like fixed capacity restrictions of services, meaning that two or more users must not select the same service. Such restrictions are not defined by the users themselves but usually by the service providers. Thus we call them *non-user-defined restrictions* in the following. However, taking solely non-user-defined restrictions into account disregards aspects of possible relations between users. Indeed, in the examples provided above (e.g., coordination of workers, city day trip) it is very likely that, due to interpersonal relations of any kind (cf. Heider 1958; Schutte et al. 2001), some individuals may prefer to conduct certain actions together whereas others might rather not encounter each other. These expressions can be understood as user requests, as according to Forgas (1999) requests are commonly used in social interaction, for instance to manage relationships (Fletcher and Fitness 1995; Holmes and Rempel 1989), to negotiate and bargain (Pruitt and Carnevale 1993), or to obtain help from others (Dovidio 1984; Salovey et al. 1991). This term is also used by Martial (1992) to represent a corresponding type of causal relation between actions of agents in multi agent systems. As such a request affects other users' decisions in a sense that there exist dependencies, we will henceforth use the term *Inter-User-Request* (IUR).

In this paper, we aim at a service selection approach providing decision support for multi user processes. To the best of our knowledge, not a single service selection approach exists which considers user-defined requests in terms of IUR yet. Therefore, we develop a novel approach considering multiple users and – in particular – IUR. The contribution of our paper is threefold:

❶ We define and model IUR. We distinguish thereby four fundamental forms of requests – mutual vs. simultaneous and complementary vs. conflicting cases.

❷ The simultaneous case requires a concept to consider temporal relations, especially waiting times, to coordinate users' actions. Hence, we provide a modeling concept to address this issue.

❸ Finally, we present an optimization model for multi user service selection. Besides the preferences and global end-to-end constraints regarding the NFP of the participating users, the concepts of ❶ and ❷ are taken into account accordingly.

Addressing ❶ to ❸, we find that decision support in a multi user context can be enhanced considerably.

The remainder of this paper is structured as follows: In the next section, we discuss the related literature and our contribution concerning the identified research gap. In addition, we introduce our model setup as well as a real-world example. The latter is used to illustrate the problem of service selection considering NFP and serves as a basis for the evaluation of our approach later on. In the third section, our multi user service selection approach, which addresses the aspects ❶ to ❸, is presented. In the fourth section, we provide an evaluation of our approach in respect of correctness, practical applicability, and performance. Finally, we conclude our paper with a short discussion on limitations and an outlook on further research.

2 Background

The following subsection provides an overview of the literature related to our research and a discussion of our contribution in terms of the identified research gap, ensued by the presentation of our model setup and a real-world example to illustrate our approach.

2.1 Related Literature

Our research is related to the literature on (1) QoS-aware service selection and contributes in particular to the literature on (2) multi user service selection.

In the literature, (1) QoS-aware service selection has been widely discussed for a single user context (cf., e.g., Han et al. 2011). A common way is to conceptualize the respective problem as an optimization problem, where the optimal service composition is obtained by solving an optimization model under consideration of the user's preferences and global end-to-end constraints regarding different NFP (e.g., Alrifai et al. 2012; Alrifai and Risse 2009; Ardagna and Pernici 2007; Canfora et al. 2005; García et al. 2008; Lin et al. 2005; Yu et al. 2007; Zeng et al. 2004). Alrifai and Risse (2009) and Yu et al. (2007), for instance, regard the service selection problem as multi-choice, multidimensional knapsack problem (MMKP), whereas García et al. (2008) and Lin et al. (2005) utilize a constraint satisfaction model to solve the underlying optimization problem. Approaches on QoS-aware service selection generally consider only a single user. Indeed, while they still could be applied on problem definitions with multiple users (by conducting the service selection separately for each user), any *dependencies* among the single users' decisions and thus the service compositions of the users would have to be neglected.

The approaches in the literature on (2) multi user service selection particularly aim at a consideration of such *dependencies*. In general, dependencies among the decisions and service compositions of the users can originate from two possible sources: First, *user-defined* requests referring to other users – which we defined as *IUR* and what will be the focus of this paper. The main characteristic of *user-defined* requests is that they are not necessarily hard restrictions in the sense that they must be satisfied in a feasible solution of the corresponding service selection problem. They rather reflect that a user associates a particular (positive or negative) value with the realization of the IUR. In consequence, IUR can influence the optimal service composition. Second, *non-user-defined* restrictions which have been addressed by several approaches that either focus on situations, where the mutual use of a certain service is mandatory, or capacity restrictions during service selection. In the following, we discuss the approaches in greater detail that address only *non-user-defined* restrictions in multi user service selection, as to the best of our knowledge, no approaches exist that aim at a consideration of *user-defined* requests.

Wang et al. (2010), Wanchun et al. (2011) and Benouaret et al. (2012) aim at selecting an optimal service for several users and a single service class, where the mutual use of a service by different users is mandatory. Wang et al. (2010) consider a situation, where the management of a company prescribes that two departments have to use a single storage service, which results in potential conflicts regarding the NFP of a service. To resolve such dependencies, they use a concept called CP-nets, whereas Wanchun et al. (2011) refer to an AHP (Analytical Hierarchy Process) approach. Benouaret et al. (2012) determine a pareto-optimal front of services by means of a calculated Jaccard-coefficient.

Jin et al. (2012b), Kang et al. (2011), Liang et al. (2013) and Wang et al. (2014) aim at selecting the optimal services for several users and a single service class, while taking capacity restrictions concerning the services into account. Kang et al. (2011) consider situations, where service provider define an upper restriction regarding the concurrent service invocation. If more users than processible are requesting the same service (as this service is optimal for them), some users have to evade to another functional equivalent, but not optimal service. Similar approaches can be found in (Jin et al. 2012b), (Liang et al. 2013) and (Wang et al. 2014), as their work is based upon Kang et al. (2011). He et al. (2012), Jin et al. (2012a) and Shen et al. (2012) focus on the same objective (i.e., capacity restrictions), but with the aim to consider several service classes and thus a service composition or process, respectively. Shen et al. (2012), for instance, suggest an approach, where the service selection is performed separately for each user. In case any conflicts prior or during the execution of the process arise, a particular error handling mechanism is invoked, where the affected users conduct an auction-based negotiation, followed by a re-selection mechanism. Thus, occurring dependencies among the users due to capacity restrictions are handled apart from the actual service selection. In the approaches of Jin et al. (2012a) and He et al. (2012), the common single user optimization model as proposed, for instance, in (Yu et al. 2007) is extended by the dimension “user”. By this, the authors maximize the utility over all users while taking capacity restrictions – as described above – into account. Moreover, He et al. (2012) deal with the case that every service can only be selected for one user, that means the capacity is set to 1.

2.2 Research Gap and Contribution to Research

Besides considering user preferences regarding certain NFP, similar to single user approaches (cf. (1)), existing multi user service selection approaches (cf. (2)) account for the fact that dependencies exist when multiple users are considered. However, these approaches do not aim to model user-defined requests referring to other users by means of *IUR*. Rather, only *non-user-defined* restrictions are taken into account, which are implemented straightforwardly as constraints. Hence, we aim for modeling and organizing *user-defined* requests by means of *IUR* (cf. ❶).

Moreover, several approaches address capacity restrictions while determining the optimal service composition for a process and multiple users. Here, users usually are forced to select a minor good service (conflicting case), instead of, for instance, wait until the optimal service is available again, which could potentially be more beneficial for the user. In the opposite, complementary case, a user could also wait for another user in order to be able to conduct a service together. To address both cases, a concept to coordinate user decisions regarding temporal relations is required. This is not in the focus of existing multi user service selection approaches so far. Hence, we aim for modeling a concept to consider temporal relations among users by means of simultaneous *IUR* (cf. ❷).

To conclude, dependencies in multi user service selection have not been sufficiently addressed so far. To address this research gap, we contribute to the current body of knowledge in multi user service selection by providing a novel optimization model (cf. ❸), where user-defined requests (cf. ❶) and a concept to coordinate users regarding temporal relations (cf. ❷) are considered in a well-founded way. By this, we aim for an enhanced decision support for multiple users conducting the same process.

2.3 Model Setup

In the following, we introduce our model setup in line with existing works, which means those definitions and modeling elements that can serve as a *common* knowledge base. Later on, when proposing our approach, this allows us to comprehensibly explicate and differentiate between existing knowledge and our contribution ❶ to ❸.

We consider a sequential process that consists of a number of actions i (with $i = 1$ to I) that contribute to achieve an intended goal. Each action i is represented by a service class S_i that includes all functional equivalent services – which we refer to as service objects – s_{ij} (with $j = 1$ to J_i) that are able to implement the corresponding action. Moreover, we focus on a number of attributes n (with $n = 1$ to N) describing the NFP of a service object $s_{ij} \in S_i$. Thus, we introduce $q_{ij} = \left[q_{ij}^1, \dots, q_{ij}^N\right]^T$ as the NFP vector for service object s_{ij} including the values for each single attribute n.

For the selection of service objects where several NFP values have to be considered, we use in line with the existing literature (cf. Alrifai et al. 2012; Ardagna and Mirandola 2010; Ardagna and Pernici 2007; Cui et al. 2011; Heinrich et al. 2015a; Sun and Zhao 2012; Surianarayanan et al. 2015) a utility function U. The purpose of this function is to map the different NFP values of a service object onto a single utility value. We divide the set of attributes thereby into two subsets. The first subset contains all attributes where the corresponding NFP value needs to be minimized (e.g., price of a service object). These attributes will be denoted as n^- in the following. The second subset contains all attributes, where the corresponding value needs to be maximized (e.g., recommendation value of a service object). Those attributes will be denoted as n^+ in the following. To determine the utility value of a service object, without loss of any generality, we apply simple additive weighting (SAW). In a first step, the values of the NFP of the service objects are normalized in the interval [0; 1] to ensure comparability between different scaled NFP values. Similar to Alrifai et al. (2012), this is achieved by using the (possible) maximum and minimum NFP values over all service classes S_i. For the attributes n the aggregated values are defined as follows:

$$Pmin(n) = \sum\nolimits_{i=1}^{I} \big(Pmin(i,n)\big);\ Pmin(i,n) = \min_{s_{ij} \in S_i} q_{ij}^{n} \tag{1}$$

$$Pmax(n) = \sum\nolimits_{i=1}^{I} \big(Pmax(i,n)\big);\ Pmax(i,n) = \max_{s_{ij} \in S_i} q_{ij}^{n} \tag{2}$$

In a second step, the normalized NFP values of the attributes are weighted with the preferences of the user. Hence, the utility U_{ij} of a service object s_{ij} is defined as follows:

$$U_{ij} = \sum_{n^-=1}^{N^-} \left(\frac{Pmax(i,n^-) - q_{ij}^{n^-}}{Pmax(n^-) - Pmin(n^-)}\right) * w^{n^-} + \sum_{n^+=1}^{N^+} \left(\frac{q_{ij}^{n^+} - Pmin(i,n^+)}{Pmax(n^+) - Pmin(n^+)}\right) * w^{n^+} \tag{3}$$

Here, $q_{ij}^{n^-}$ and $q_{ij}^{n^+}$ are the NFP values for each single attribute n of the NFP vector of service object s_{ij}. The user can set up preferences (i.e., w^{n^-}, w^{n^+}) for each attribute n, where $0 < w^{n^-}, w^{n^+} < 1$ and $\sum_{n^-=1}^{N^-} w^{n^-} + \sum_{n^+=1}^{N^+} w^{n^+} = 1$ hold. Based on this, the utility of a service composition can be computed by aggregating the utility of the selected services. In order to consider user requirements regarding the aggregated NFP values of a service composition, we introduce a global end-to-end constraints vector $Q_c = [Q_c^1, \dots, Q_c^N]^T$ including the values for each attribute n. These constraints can be defined either as lower (for attributes n^+) or upper bounds (for attributes n^-).

Based on the notation presented above, the service selection problem is represented as a 0-1 multi-choice multidimensional knapsack problem (MMKP) (cf. Alrifai et al. 2012; Yu et al. 2007; Zeng et al. 2004) and thus the optimization model is defined as follows[3]:

$$\max_{x_{ij}} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} U_{ij} * x_{ij}$$

$$s.t. \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} q_{ij}^{n} * x_{ij} \leq Q_c^n \ \ \forall n = 1 \ to \ N$$

$$\sum_{s_{ij} \in S_i} x_{ij} = 1 \ \forall i = 1 \ to \ I; x_{ij} \in \{0,1\} \tag{4}$$

Considering the service classes S_i included in the process as well as the respective service objects $s_{ij} \in S_i$, the optimization model determines for a single user the decision variables x_{ij} ($x_{ij} = 1$ indicates that service object s_{ij} is selected, $x_{ij} = 0$ that it is not) to maximize the accumulated utility of the selected service objects. For each service class S_i exactly one service object has to be selected. At the same time the aggregated NFP values of the service composition need to satisfy the global end-to-end constraints $Q_c = [Q_c^1, \dots, Q_c^N]^T$ for every attribute n.[4]

[3] Please notice that for attributes n^+, the corresponding constraint has to be multiplied by -1 so that it holds that the aggregated NFP value needs to be less than the given constraint.

[4] Obviously, the model in (4) can straightforwardly be extended by non-user-defined restrictions as discussed in '*Related Literature*' (cf., e.g., Jin et al. 2012a; Wang et al. 2010). As such restrictions have already been addressed by existing works and are not part of our contribution ❶ to ❸, we do not consider this further.

2.4 Real-world Example

In this section, we introduce our real-world example which is used to illustrate how a service selection regarding NFP can be addressed. Without loss of generality (w.l.o.g.), we focus on the tourism domain, as this domain in particular has shown to support the willingness to use (cf. Gerpott and Berg 2011) and the value of decision support (cf. Vos et al. 2008), for instance, by means of mobile apps. Moreover, this domain has its practical relevance. *ProgrammableWeb*, for instance, offers more than 1,000 (information) services that can potentially be used for the provision of information by means of service objects. We used the information services *Google Places*[5] and *TripAdvisor*[6] to determine the set of available service objects as well as their *duration (D)*, *costs (C)*, and *recommendation value (R)* (ratings by former users that are denoted in numbers from 1 to 5) as NFP[7]. The considered process with its corresponding actions or service classes, respectively, can be constructed in an automated way (cf., e.g., Heinrich et al. 2015b; Heinrich and Schön 2015; Henneberger et al. 2008) and is illustrated in Figure 1.

The goal of the process is to support users in conducting a city day trip (in our case Munich, Germany) and comprises typical actions such as "*CityTour*", "*Lunch*", and "*Museum*". Figure 1 illustrates, that users can choose between different actions they may want to accomplish (cf. S_3 vs. S_4 vs. S_5; S_7 vs. S_8 vs. … vs. S_{11}; S_{13} vs. S_{14} vs. S_{15}; for pick construct cf. Wan et al. 2008; Yu et al. 2007). For instance, a user can either visit a museum or a zoo/aquarium or choose to do wellness (cf. S_3 vs. S_4 vs. S_5). To deal with the pick construct within existing optimization models (cf. Term 4), the functional equivalent service classes of the pick construct (e.g., S_3 vs. S_4 vs. S_5) are merged into a single service class. This can be done, as merging of functional equivalent service classes does not affect the selection process or the optimal service composition for each user, respectively.

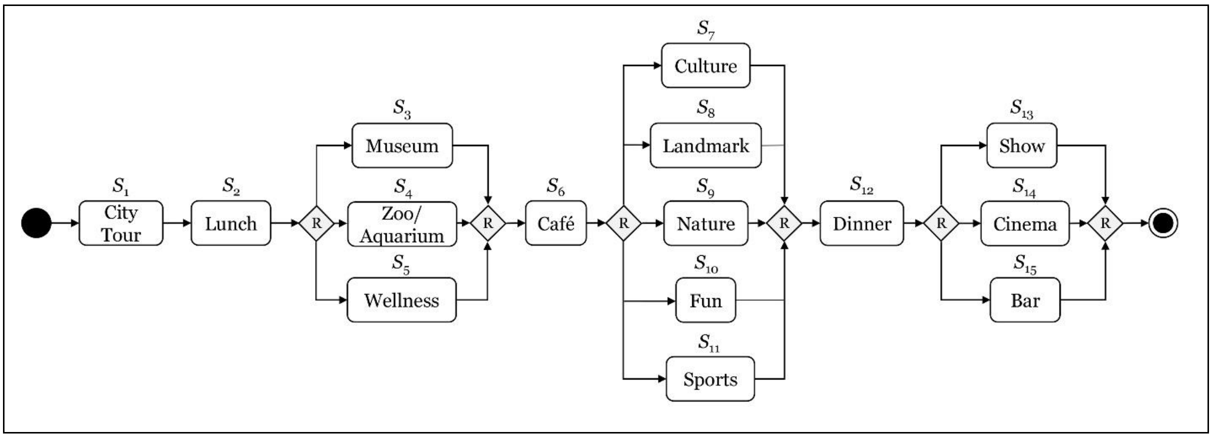

Figure 1. Real-world Process

In our real-world example, we further consider five different users, where each user has its individual preferences as well as global end-to-end constraints regarding the different NFP. For each user, the service selection is – in line with existing works – conducted separately, which

[5] http://www.programmableweb.com/api/google-places, accessed September 2015
[6] http://www.programmableweb.com/api/tripadvisor, accessed September 2015
[7] The entire data can be made available on request.

means no dependencies between users of any kind are taken into account. Here, w.l.o.g. we used integer programming (cf. Zeng et al. 2004) as an analytical approach to solve the corresponding optimization model. Table 1 contains the results for each user. Please notice, that so far the results solely depend on the individual preferences regarding the NFP and the global end-to-end constraints of the users (cf. Term 4).

User	Optimal service composition	Costs (€)	Duration (min)	Recommendation Value	Utility
1	$s_{1\,4}, s_{2\,9}, s_{3\,1}, s_{6\,9}, s_{7\,4}, s_{12\,27}, s_{15\,17}$	44.00	600	33.23	0.8433
2	$s_{1\,4}, s_{2\,9}, s_{3\,13}, s_{6\,2}, s_{7\,5}, s_{12\,26}, s_{14\,4}$	34.00	600	32.59	0.9086
3	$s_{1\,5}, s_{2\,7}, s_{3\,2}, s_{6\,7}, s_{7\,6}, s_{12\,27}, s_{15\,17}$	46.50	540	33.73	0.6571
4	$s_{1\,2}, s_{2\,9}, s_{3\,1}, s_{6\,2}, s_{7\,4}, s_{12\,27}, s_{15\,3}$	28.50	600	30.15	0.8967
5	$s_{1\,4}, s_{2\,9}, s_{3\,13}, s_{6\,8}, s_{7\,5}, s_{12\,27}, s_{15\,49}$	40.00	600	33.09	0.4677

Table 1. Optimal Service Composition per User

Focusing, for instance, on user 3, the optimal service composition is determined to $s_{1\,5}, s_{2\,7}, s_{3\,2}, s_{6\,7}, s_{7\,6}, s_{12\,27}, s_{15\,17}$ with end-to-end costs of €46.50, an end-to-end duration of 540 min and an end-to-end recommendation value of 33.73 (points). The results depicted in Table 1 serve as a reference base later on.

However, besides preferences and global end-to-end constraints regarding the NFP, in the context of multi user processes users might have requests referring to other users. An example for this could be that *user 3 requests to go into the Bavarian restaurant 'Hofbraeuhaus' for dinner together with user 2* – which could be understood as a complementary, simultaneous IUR. Such IUR – and particularly such time-dependent IUR – are addressed by the approach presented in the following.

3 A novel multi user service selection approach

In the following, we present our multi user service selection approach, where user-defined requests referring to other users by means of *IUR* are considered. We therefore first show how different types of IUR (mutual and simultaneous, complementary and conflicting) can be modeled and organized (cf. ❶). Based on this, we present our concept to coordinate users by means of temporal relations including waiting times and thus enabling simultaneous IUR (cf. ❷). Taking the *Model Setup* as a foundation, we finally present our optimization model (cf. ❸) that addresses both ❶ and ❷.

3.1 Modeling Inter-User-Requests (IUR)

Modeling IUR is one of the core challenges of our paper. An IUR encompasses a set of users and actions, where an *action* can refer to a whole service class or to a specific service object. Further, an IUR can comprise – if required – different actions/service classes for each participating user. When specifying an IUR, a user has to associate a particular value with the realization of that IUR. Moreover, a set of users can probably define the same or (very) similar IUR which can be interpreted as "group requests". IUR extend significantly non-user-defined restrictions (cf. section '*Related Work*'). Indeed, they shall represent preferences comparing different alternatives (e.g., perform an action together with other users or not) instead of hard restrictions.

To conceptualize IUR, we consider two dimensions: Regarding the first dimension, there can be actions (implemented by service classes and service objects) which a user requests to do together with one or more other users. In contrast, there might be actions the user does not want to conduct with particular other users. Thus, in accordance with literature in coordination research, we differentiate between complementary and conflicting situations (cf. Martial 1992). Regarding the second dimension, the complementary or conflicting usage of actions can refer to simultaneous usage (e.g., two users want to conduct an action together at the same time) or be time-independent (mutual case). Overall, we therefore distinguish four fundamental categories, which are presented in Table 2:

	Complementary	**Conflicting**
Mutual (time-independent)	{1} *Complementary mutual usage*: – A user requests to perform an action *together* with one or more other users. – A *positive* value is associated with this IUR.	{2} *Conflicting mutual usage*: – A user requests *not* to perform an action together with one or more other users. – A *negative* value is associated with this IUR.
Simultaneous (time-dependent)	{3} *Complementary simultaneous usage*: – A user requests to perform and thus to start an action *together* with one or more other users at *the same time*. – Potential occurrence of *waiting times* for users. – A *positive* value is associated with this IUR.	{4} *Conflicting simultaneous usage*: – A user requests *not* to perform an action together with one or more other users at *any given moment of time*. – Potential occurrence of *waiting times* for users. – A *negative* value is associated with this IUR.

Table 2. Categories of IUR

According to the traditional service selection problem (cf. section '*Model Setup*') users are able to define their preferences and global end-to-end constraints regarding the NFP. Now, we extend these possibilities by allowing a user a (with $a = 1$ to A) to specify K_a different IUR (with $k = 1$ to K_a). In order to address this, the user has to define the participating users, for every participating user a particular action and whether the IUR is restricted to mutual usage (cases {1} and {2}) or simultaneous usage (cases {3} and {4}). S/he then specifies a certain request value V for the realization of that IUR, which represents how important an IUR is compared to other IUR. Thus, it is similar to other NFP like costs or duration, which allow us to compare and valuate different service objects against each other based on their NFP and request values, respectively. Besides comparing different IUR against each other, IUR have to be set in relation to the other NFP. Therefore, a user a also specifies a preference w_a^{IUR} for IUR in general (cf. also the preferences for the other NFP (e.g., w_a^-, w_a^+)). To receive a normalized and weighted utility for IUR, the same utility function (cf. Term 3) as for the utility calculation of service objects is applied to the request value V. For a mutual case the utility is represented by $\widehat{U}$ and for a simultaneous case by $\overline{\overline{U}}$. This leads to the following formal definition of IUR:

$$IUR_k = \left(a', \widehat{U}_k, \overline{\overline{U}}_k, A_k, X_k\right) \tag{5}$$

A 5-tuple IUR_k consists of the user a', who specified the IUR, the possible positive or negative utility for the mutual usage case $\widehat{U}_k$, the possible positive or negative utility for the simultaneous usage case $\overline{\overline{U}}_k$, the set of users A_k which are participating in that IUR, and the set X_k including the decision variables x_{aij} for the service objects s_{ij} of user a affected by the IUR. Here, the binary decision variables x_{aij} are defined as follows: x_{aij} is 1, if the service object s_{ij} is selected for user a, and 0, if not. The latter implies that every participating user in a particular IUR is assigned only a single service object. If an IUR is specified for a whole service class consisting of several service objects, multiple 5-tuples IUR_k are defined to realize that IUR. If IUR_k represents one of the cases $\{1\}$ or $\{2\}$, $\widehat{U}_k$ is different from null and $\overline{\overline{U}}_k$ is null. If IUR_k represents one of the cases $\{3\}$ or $\{4\}$, it is the other way round.

The IUR cases $\{3\}$ and $\{4\}$ induce temporal dependencies among the service compositions of the participating users. Thus, waiting times may occur. The next section shows how such waiting times can be modeled to coordinate the participating users.

3.2 Enabling Temporal Coordination to Represent Simultaneous Cases of IUR

To address the temporal coordination of multiple users, we have to consider the selected service objects of the IUR's participating users in terms of their duration as well as potentially occurring waiting times. Such a coordination, for instance, should let users wait if the overall additional utility achieved by a simultaneous usage outweighs the loss of utility by waiting. With the IUR, the additional utility for simultaneous usage has already been defined, however, so far a concept to model and consider *waiting times* and especially the resulting loss of utility is still missing.

According to our model setup, we focus on a process that consists of different actions/service classes. As a result, users have the possibility to wait right between two succeeding actions in order to start the particular action with other users (cf. $\{3\}$), or to prevent that a particular action is conducted together with other users (cf. $\{4\}$). To straightforwardly model waiting times without increasing the complexity of the general character of the optimization model, we propose to conceptualize *waiting times* as special service classes consisting of several service objects that represent different waiting times (denoted as ordinary NFP).

More precisely, we introduce waiting time WT (with $WT \in N^-$) as a NFP of a service object. Further, we extend our model setup by so called *waiting service classes*, which we denote as S_i^* in the following. These service classes are characterized by the fact that for each $s_{ij} \in S_i^*$ the NFP are only used to represent *WT*. To consider S_i^* within a process, they are placed right before a service class representing an action which can potentially be delayed by a user deciding to prefer waiting. This fact is illustrated in Figure 2.

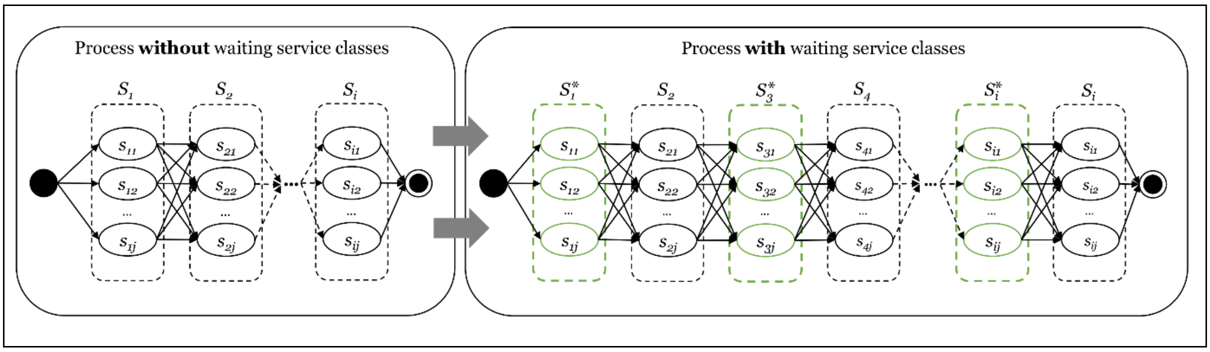

Figure 2. Considering Waiting Service Classes

For instance, let us focus on a situation in which two users want to simultaneously conduct the service object s_{41} together, which implies that they both want to start this action at the same time. Potential waiting times depend on the duration q_{ij}^{D} of the service objects s_{ij} the users have already conducted so far (e.g., service objects selected for S_2). As a result, there exist three possible scenarios:

(1) Waiting is not necessary (e.g., if the duration of the selected service compositions till s_{41} is the same for both users), or
(2) it is proposed to wait for one of the two users (e.g., if the duration of the selected service compositions till s_{41} for both users is a different one), or
(3) waiting is dispensable (e.g., the IUR will not be realized).

To decide which case is the most beneficial, an optimization model should evaluate if the additional utility $\overline{\overline{\overline{U}}}_k$ achieved by simultaneous usage outweighs the utility loss caused by waiting. Please notice, that this can also affect the selection of preceding and succeeding service objects of s_{41}. To be able to support this decision within an optimization model, the question arises how to determine the value of q_{ij}^{WT} for different service objects of waiting service classes. To resolve this issue, we model attributes representing "time" – waiting time and duration – as discrete variables, such that $q_{ij}^{WT}, q_{ij}^{D} \in \{k * c | k \in \mathbb{N}_0\}$, with $c \in \mathbb{R}$. With the aim of determining the optimal service composition at planning time (in line with traditional QoS-aware service selection approaches), we argue that the use of discrete variables seems appropriate, as the parameter c can be adjusted to every purpose or need. In that way, waiting times can be determined and straightforwardly considered within our optimization model (cf. next section).

3.3 Optimization Model for Multi User Service Selection

Similar to the service selection for the single user case, the multi user service selection problem can be formulated as knapsack problem and in our case more precisely as a non-linear 0-1 knapsack problem.

As already noted, we realize the consideration of multiple users by user-dependent decision variables x_{aij} for the individual service objects. Each decision variable x_{aij} is associated with a utility U_{aij} taking the user's preferences w_a^n regarding the attribute n (with $n = 1$ to N) into account. By this, the utility values for the same service object can be completely different for two users. Additionally, as described in the previous section IUR and thus temporal relations including potential wait times have to be considered. For this purpose, each decision variable x_{aij} is associated with a utility $\widehat{U}_k$ or $\overline{\overline{U}}_k$ taking the user's preferences w_a^{IUR} regarding the IUR k (with $k = 1$ to K) into account. By this, the utility values even for the same IUR (e.g., two users request to use the same service object with the other user at the same time) can be completely different for two users. To sum it up: modeling the user's preference regarding the NFP and IUR as described above allows for a high flexibility. This concerns the determination of different or same utility scores – whatever is needed or specified by the users – when different users and the same service objects are considered.

Therefore, our optimization model for a multi user service selection is defined as follows:

$$\max_{x_{ij}, s_k} \sum_{a \in A} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} U_{aij} x_{aij} + \sum_{\{k \in K | \widehat{U}_k \neq 0\}} \widehat{U}_k \prod_{x_{aij} \in X_k} x_{aij} + \sum_{\{k \in K | \overline{\overline{U}}_k \neq 0\}} \overline{\overline{U}}_k s_k \prod_{x_{aij} \in X_k} x_{aij} \tag{6}$$

$$s.t. \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} q_{ij}^n \, x_{aij} \leq Q_a^n \quad \forall \, n = 1 \; to \; N \; ; \; \forall a = 1 \; to \; A \tag{7}$$

$$\left[\max_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum_{i=1}^{i'-1} \sum_{s_{ij} \in S_i} \sum_{n \in \{D, WT\}} q_{ij}^n \, x_{aij} \right) - \min_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum_{i=1}^{i'-1} \sum_{s_{ij} \in S_i} \sum_{n \in \{D, WT\}} q_{ij}^n \, x_{aij} \right) \right] * s_k = 0 \tag{8}$$

$\forall \, k = 1 \; to \; K \mid \overline{\overline{U}}_k > 0$

$$\left[\begin{array}{l} \max\limits_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum\limits_{i=1}^{i'-1} \sum\limits_{s_{ij} \in S_i} \sum\limits_{n \in \{D, WT\}} q_{ij}^n x_{aij} \right) \\ - \min\limits_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum\limits_{i=1}^{i'-1} \sum\limits_{s_{ij} \in S_i} \sum\limits_{n \in \{D, WT\}} q_{ij}^n x_{aij} + q_{i'j'}^d x_{ai'j'} \right) \end{array} \right] * (1 - s_k) \geq 0 \tag{9}$$

$\forall \, k = 1 \; to \; K \mid \overline{\overline{U}}_k < 0$

$$\sum_{s_{ij} \in S_i} x_{aij} = 1 \quad \forall \, i = 1, \ldots, I \; ; \forall \, a = 1 \; to \; A \tag{10}$$

$$with \; x_{aij} \in \{0, 1\}; \; s_k \in \{0,1\} \tag{11}$$

Considering the service classes S_i as well as the respective service objects $s_{ij} \in S_i$, the optimization model determines for all users $a \in A$ the decision variables x_{aij}, s_k ($x_{aij} = 1$ indicates that service object s_{ij} is selected for user a, $x_{aij} = 0$ that it is not; $s_k = 1$ indicates that $\overline{\overline{U}}_k$ is realized for user a, $s_k = 0$ that it is not), to maximize the accumulated utility of the selected service objects as well as the corresponding IUR (represented by $\widehat{U}_k$ and $\overline{\overline{U}}_k$). For each user $a \in A$ and for every service class S_i, exactly one service object has to be selected. At the same time, the aggregated NFP of a service composition need to satisfy the given global end-to-end constraints Q_a^n for each attribute n (with $n = 1$ to N) and user a (with $a = 1$ to A).

Focusing on the IUR specified by $\widehat{U}_k$ and $\overline{\overline{U}}_k$: If the product $\prod_{x_{aij} \in X_k} x_{aij}$ is 1, the associated $\widehat{U}_k$ of a mutual IUR will be realized and part of the solution. The same holds for $\overline{\overline{U}}_k$ of a simultaneous IUR but only if the associated binary auxiliary variable s_k is 1, too. The distinction between mutual ({1}, {2}) and simultaneous ({3}, {4}) IUR is necessary, since the simultaneous IUR – unlike the mutual IUR – have to take temporal relations into account. For the mutual usage cases {1} and {2} there is no need for additional constraints, since they can be implemented directly in the objective function by multiplication of the utility $\widehat{U}_k$ with the corresponding decision variables x_{aij} of the affected service objects.

The simultaneous usage cases, however, require particular constraints to incorporate the temporal relations and thus allow for a coordination of the participating users. Therefore, each complementary simultaneous IUR {3} is considered in equation (8) and each conflicting simultaneous IUR {4} in equation (9). Both equations are linked to their associated utilities $\overline{\overline{U}}_k$ in the objective function through binary auxiliary variables s_k. Since $\overline{\overline{U}}_k$ for the complementary case {3} is a positive value, the user a' who specified that IUR wants $\overline{\overline{U}}_k$ to be realized. But a particular $\overline{\overline{U}}_k$ is only intended to be realized if for all participating users $a \in A_k$ the corresponding services $x_{ai'j'} \in X_k$ are selected and start at the same time. This is represented by constraint (8). Since we model all time-dependent NFP (i.e., *duration* and *waiting time*) as discrete and waiting time as ordinary service classes, this can be determined when solving the model by comparing the selected service objects of the different users regarding *duration* and *waiting time*. The optimization model adjusts the decision variables to meet the constraints and hence enables a coordination between the different users. Therefore, it is sufficient to check whether the difference in time between the user $a \in A_k$ with the maximum time (*duration* plus *waiting time*) spent until that point in time and the user $a \in A_k$ with the minimum time spent is equal to 0. If the condition of starting at the same time is not already met, it is assessed, whether it is more favorable to renounce the utility $\overline{\overline{U}}_k$ or to fulfill the condition for each user by changing a service object prior in the process or by selecting an appropriate waiting service object somewhere before.

Contrary to the complementary case, $\overline{\overline{U}}_k$ for the conflicting case {4} is negative and thus the user wants $\overline{\overline{U}}_k$ not to be realized. To avoid the realization of $\overline{\overline{U}}_k$ there must not be a single moment where all affected users $a \in A_k$ conduct their particular services $x_{ai'j'} \in X_k$ simultaneously. The necessary analysis for this purpose is done in (9) by comparing – analogical to (8) – the amount of time of the user $a \in A_k$ with the maximum time spent to the user $a \in A_k$ with the minimum time spent but here, including the duration of the service object $x_{ai'j'} \in X_k$. If the difference is greater or equal to null, the negative $\overline{\overline{U}}_k$ in the objective function can be avoided (for s_k set to 0). This again implicitly determines whether accepting the negative utility $\overline{\overline{U}}_k$, choosing other service objects or accepting waiting times is the optimal option.

To determine the optimal service objects (i.e., the service compositions) for each user the prescribed optimization model has to be solved exactly. To achieve this, integer programming can be used (Nemhauser and Wolsey 1988) as we will do in our evaluation, since an exhaustive enumeration is – due to the NP-hardness of the problem (Pisinger 1995) – only possible for very small problem instances. But in order to use integer programming our non-linear optimization model has to be transformed as usual into a linear one at first. Particularly, this refers to our objective function (6) and constraints for the simultaneous IUR cases (8) and (9).

4 Evaluation

In this section, we evaluate our approach with respect to correctness, practical applicability, and performance. The general evaluation design, and in particular the design for the evaluation of correctness and performance, follows evaluation designs of many state-of-the art service selection approaches (cf., e.g., Alrifai et al. 2012; Ardagna and Mirandola 2010; Yu et al. 2007; Zeng et al. 2004). In doing so, the findings of our performance evaluation can basically be compared to other selection approaches. As our approach aims to provide decision support for multi user processes in real-world scenarios, we further focused on its practical applicability. We thereby decided to conduct an experiment with graduated students (who represent the users) based on both a real-world process and real-world data from *GooglePlaces* and *TripAdvisor*. In so doing, we are able to illustrate the benefits of our approach in comparison to existing service selection approaches in a comprehensive way.

4.1 Correctness

To analyze the correctness of our approach, we implemented the non-linear optimization model as well as the linearized version in Java. The non-linear model was solved by an enumerative approach, whereas for solving the linear model, Gurobi[8], an integer programming solver, was used. To verify these implementations and ensure a correct and consistent implementation, a manual analysis of the source code was done by other persons than the programmers and we made a series of tests using the JUnit Framework, including runs with extreme values, JUnit regression tests, and unit tests. The implementation did not show any errors at the end of the test phase. In a further step, we evaluated over 10,000 randomly generated scenarios (with up to 2,985,984 possible combinations of service objects) on the basis of both the non-linear and the linear implementation in order to examine their consistency. As a result, the solutions were the same for each assessed scenario and implementation.

Moreover, we tested whether both models deliver correct results in comparison to Excel calculations. The results of 17 artificial scenarios (two service classes á two service objects, each waiting service class á three waiting service objects, and three users[9]) with different IUR provided by the two specified models were compared to the results of Excel calculations. The solutions were again the same for each assessed scenario and implementation. Thus, we conclude that our non-linear and linear optimization models are consistent and seem to work correctly.

[8] http://www.gurobi.com/, accessed September 2015

[9] The exact solving of greater problem sizes is not feasible with Excel calculations.

4.2 Practical Applicability

The aim of this evaluation step is to demonstrate the practical applicability of our approach based on our real-world example. To do so, we use the following setting:

Process: We consider a tourism city day trip (cf. Figure 1), which consists of 15 service classes. For each of these service classes, we use the information services *Google Places* and *TripAdvisor* to determine feasible service objects as well as their NFP. Furthermore, service objects with no fixed duration like restaurants were defined multiple times each with a different specification for duration. This resulted in 332 service objects and 89 billion possible service compositions per user.[10]

Non-functional properties: We consider *costs* (*C*), *recommendation value* (*R*), *waiting times* (*WT*) and *duration* (*D*) as NFP of a service object. The values of *WT* and *D* are represented by the following discrete values $\{0,15,30,45,60,75,90,105,120,135,150\}$ which seems to be suitable for a day trip – particularly as we focus in this paper on the selection of services at planning time instead of runtime.

Users and IUR: In total we consider five different users who are graduated students. We conducted an experiment with those students in order to receive their individual preferences w_a^n and IUR w_a^{IUR}, their global end-to-end constraints regarding the NFP and three IUR each, which led to 15 IUR with five cases $\{1\}$, one case $\{2\}$, eight cases $\{3\}$ and one case $\{4\}^8$. Only the global end-to-end constraint for *D* was predefined to a maximum of ten hours which seems to be realistic for a day trip. In this context it is not reasonable that the calculation of the service objects' utility aims as usual on minimizing *D* in an absolute manner. Therefore, we apply user-defined target values for each service class, for instance, two hours for dinner, and thus the utility calculation is more realistically based upon the interval between the service object's *D* and that target value.

Based on these inputs, we apply our approach. The results are illustrated in Table 3. Here, for each user the optimal service composition considering IUR is presented.

[10] A detailed list of input and output data is left out due to length restrictions. The data can be made available on request.

User	Optimal service composition	Costs (€)	Duration (min)	Wait Time (min)	Recommendation Value	Utility
1	$s_{1\,4}, s_{2\,9}, s_{3\,1}, s_{6\,9}, s_{7\,4}, s_{12\,27}, s_{15\,17}$ ($s_{1\,4}, s_{2\,9}, s_{3\,1}, s_{6\,9}, s_{7\,4}, s_{12\,27}, s_{15\,17}$)	44.00	600	0	33.23	0.8433
2	$s_{1\,4}, s_{2\,9}, s_{3\,14}, s_{6\,2}, s_{7\,4}, s_{12\,26}, s_{15\,50}$ ($s_{1\,4}, s_{2\,9}, s_{3\,13}, s_{6\,2}, s_{7\,5}, s_{12\,26}, s_{14\,4}$)	34.00	600	0	32.59	0.9086
3	$s_{1\,4}, s_{2\,7}, s_{3\,3}, s_{6\,7}, s_{9\,6}, s_{12\,27}, s_{14\,4}$ ($s_{1\,5}, s_{2\,7}, s_{3\,2}, s_{6\,7}, s_{7\,6}, s_{12\,27}, s_{15\,17}$)	44.00	600	30	33.16	0.6477
4	$s_{1\,2}, s_{2\,6}, s_{3\,1}, s_{6\,2}, s_{7\,4}, s_{12\,27}, s_{15\,3}$ ($s_{1\,2}, s_{2\,9}, s_{3\,1}, s_{6\,2}, s_{7\,4}, s_{12\,27}, s_{15\,3}$)	28.50	600	0	29.65	0.9034
5	$s_{1\,4}, s_{2\,9}, s_{3\,13}, s_{6\,8}, s_{9\,5}, s_{12\,27}, s_{15\,49}$ ($s_{1\,4}, s_{2\,9}, s_{3\,13}, s_{6\,8}, s_{7\,5}, s_{12\,27}, s_{15\,49}$)	40.00	600	0	33.02	0.4997

Table 3. Optimal Service Compositions by our Multi User Approach (In Brackets: Optimal Service Compositions of Section 'Real-word Example')

The contribution of our multi user service selection can be illustrated by analyzing the optimal service composition, for instance, of user 3 and compare it to the solution of traditional single user service selection without IUR (as illustrated in section '*Real-world Example*'). As shown in Table 3 the optimal service composition of user 3 differs in four service objects ($s_{1\,4}$ instead of $s_{1\,5}$, $s_{3\,3}$ instead of $s_{3\,2}$, $s_{9\,6}$ instead of $s_{7\,6}$, $s_{14\,4}$ instead of $s_{15\,17}$) and additionally, a wait time of 30 minutes is indicated. These facts could be directly ascribed to the following complementary simultaneous IUR – specified by the students – which is realized in the optimal solution: *User 5 requests to visit 'English Garden' together with user 3*.

Precisely, this IUR is fulfilled in case users 3 and 5 go to the same attraction 'English Garden' of service class '9) Nature' – which is true for the optimal service compositions of both users (service objects $s_{9\,6}$ (user 3) and $s_{9\,5}$ (user 5) both represent 'English Garden', but with different manifestations of *duration*). This also corresponds to an adaptation of the selected service class for both users: Instead of going to 'Asam Church' (service class '7) Culture') as indicated by the service compositions in the single user solution, 'English Garden' and hence service class '9) Nature' is selected by our approach – as requested by user 5. Additionally, they both arrive at 'English Garden' at the same time, which implies that the time period elapsed up to that point of the day trip has to be the same. This solution was proposed by our optimization model. As a consequence, user 3 has to wait 30 minutes in total for user 5 in order to visit 'English Garden' together with him/her. These 30 minutes equal user 3's global end-to-end constraint regarding *waiting time*, but the total duration of user 5's actual service composition until the visit of 'English Garden' is still another 60 minutes higher. Therefore, the solution of our approach proposes user 3 to take a longer city tour and a longer stay at 'Die Pinakotheken' in comparison to the previous solution presented in section '*Real-world Example*'. Because of the high value defined by user 5 for this IUR, the corresponding utility $\overline{\overline{U}}_k$=0.03 justifies these changes. Furthermore, it compensates the decrease in user 3's utility caused by selecting 'English Garden' instead of 'Asam Church'.

Moreover, another fulfilled IUR is *user 2 requests that not all users 1-5 should go to 'Andechser am Dom'* since the negative utility user 2 associates with this IUR is not realized in the optimal solution. On the contrary, some other IUR like *user 3 requests to dine together with user 2 in 'Hofbraeuhaus'* are not fulfilled because, for instance, some global end-to-end constraints regarding NFP would be violated or the associated value by user 3 was not high enough to compensate the utility reduction caused by the necessary selection of other service objects.

As the application of our approach on this real-world example demonstrated, by considering IUR the outcome of the service selection could serve as enhanced decision support for its users. This was also supported by the students of the experiment when being asked to compare the results of our multi user approach to the results of the single user approach presented in section '*Real-world Example*'. They pointed out that both considering IURs in general as well as the temporal coordination and the inherent suggestions to potentially wait for other users in particular create substantial added value when planning a day trip with multiple users. The students, for example, found the option to conduct preceding actions longer (e.g., staying longer in a restaurant) for purpose of coordination especially helpful.

4.3 Performance

This section provides an evaluation of the performance regarding the computation time of our approach, in particular the linear model using Gurobi. The computation time for solving the traditional service selection problem (cf. Term 4) depends on the *number of service classes*, the *number of service objects per class* and the *number of NFP* (cf. Alrifai et al. 2012). The influence of these variables on computation time has intensively been investigated in prior research (cf. Alrifai et al. 2012; Alrifai and Risse 2009; Ardagna and Pernici 2007; Canfora et al. 2005). In our analyses, we observed similar results. Indeed, the computation time increases in an over-proportional fashion in case one of these variables is increased. In the following, we focus on the variables which have been added to address ❶ and ❷ in our approach, that means the *number of waiting service objects* per waiting class, the *number of users*, the *number of simultaneous* IUR and *the number of mutual IUR*. Our analyses base on the following setting:

Key Figure: Computation time is measured in seconds [sec].

Initial Problem Size: Our performance analysis addresses a larger service selection problem compared to evaluating model correctness (cf. above). The problem encompasses 20 service classes (i.e., actions) á 20 service objects, 20 waiting service classes, 4 NFP (including *WT*), 5 users, 5 mutual IUR and 5 simultaneous IUR as basic setting. Furthermore, each of the 20 waiting service classes consists of 10 waiting services, where the waiting time is increased from 0 to 90 in steps of 10 time units.

Data Set: We use an artificial data set, where the values have been generated randomly.

Scenarios: To evaluate the computation time of our approach, we use four different scenarios, where we change one parameter while keeping all other parameters constant (ceteris paribus):

a) The number of mutual IUR is increased from 0 to 15 in steps of 1
b) The number of users is increased from 2 to 20 in steps of 1

c) The number of waiting services per waiting class is increased from 2 to 20 in steps of 1
d) The number of simultaneous IUR is increased from 0 to 15 in steps of 1

We simulate each scenario 1,000 times and determine the average computation time. For the initial problem size this is on average 0.13 seconds. All evaluations are performed on a desktop PC with an Intel Core I7 processor with 3.4 GHz, 16 GB RAM, Win7 64bit, Gurobi 6.0 and Java 1.8. The results are summarized in Figure 3.

Computation time depending on the number of waiting service objects per waiting service class. The results show that the computation time increases apparently as the number of waiting service objects is increased (cf. Figure 3, Scenario a)). As waiting service objects are characterized by the fact that they affect only one single NFP (waiting time), their direct influence on computation time is minor compared to regular service objects (cf. traditional service selection). But on the other side, the number of waiting service objects determine the "granularity of time" inherent to the specified model and therefore influence the runtime complexity as with finer "time granularity" the solution space is increased. For example, for 20 service objects per waiting service class, the computation time on average is 0.315 sec, which seems sufficient when selecting service objects at planning time (instead of runtime). This supports our proposed concept of modeling waiting time as service objects and service classes.

Computation time depending on the number of users. With every additional user our optimization model has to consider potentially additional dependencies caused by IUR – besides the respective variables and constraints. Hence, we generally expect an over-proportional increase of computation time in case the number of users increases. Indeed, this is actually observed in our simulation experiment (cf. Figure 3, Scenario b)). Yet, for 20 users the computation time is on average 1.06 sec., which we argue to be practical especially at planning time.

Computation time depending on the number of simultaneous IUR. For the number of simultaneous IUR, our simulation experiment reveals a very similar effect, that means an over-proportional increase in computation time (cf. Figure 3, Scenario c)). This does not seem surprising, as the number of simultaneous IUR directly affects the number of constraints in the optimization model. For 15 simultaneous IUR, the computation time, however, is only about 1.08 sec. on average.

Computation time depending on the number of mutual IUR. The graph for Scenario d) in Figure 3 shows that the computation time hardly increases depending on the number of mutual IUR. This may be due to the fact that the definition of mutual IUR has only minor effect on the number of constraints in the linearized optimization model that is used for performance evaluation. Indeed, mutual IUR can directly be considered within the objective function of the non-linear model.

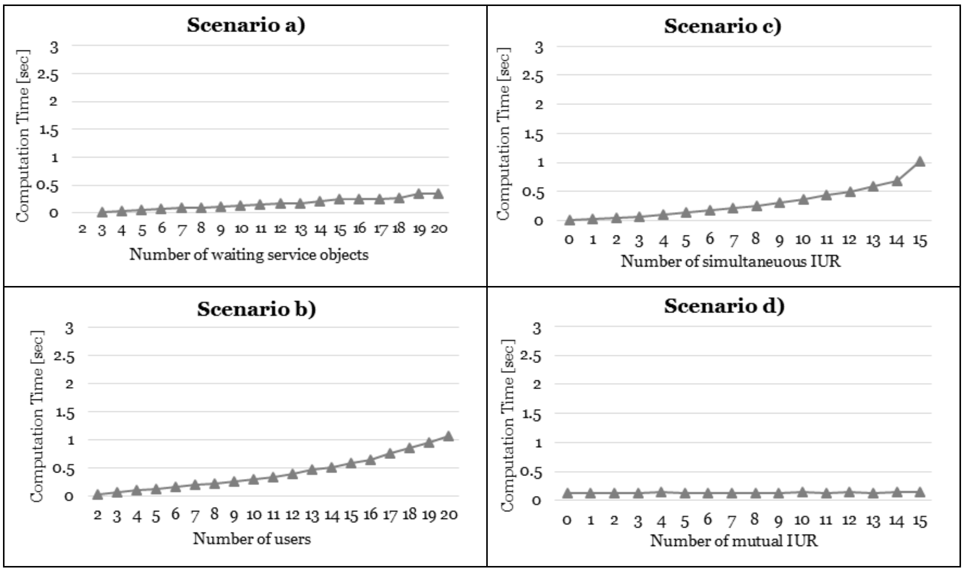

Figure 3. Performance Analysis

Summing up, we do not aim to present a computation time optimized (heuristically) approach, but rather a first approach that is capable of considering user-defined requests. Against this background, the computation times seem quite acceptable. However, it still has to be noted that our optimization model is NP-hard and generally requires exponential solving time (Nemhauser and Wolsey 1988).

5 Conclusion, Limitations, and Further Research

In this paper, we presented a service selection approach for multiple users that is able to consider dependencies between user decisions that originate from user-defined requests (i.e., *Inter-User-Requests*). For the participating users, this approach determines the optimal service compositions under the consideration that some users may prefer to conduct certain actions together, whereas others may rather not encounter each other. Existing multi user service selection approaches do not aim to address such a decision support but focus on non-user-defined restrictions. Consequently, they neither consider user-defined requests nor provide a concept to coordinate users regarding temporal dependencies.

Our service selection approach addresses this research gap (cf. ❶ - ❸). To do so, we first defined and modeled four different types of IUR – complementary mutual, conflicting mutual, complementary simultaneous, and conflicting simultaneous. We further proposed a concept to enable the consideration of simultaneous IUR. As part of that, we modeled waiting service classes before each regular service class and introduced the additional NFP *waiting time*. This approach allows a straightforward integration in our optimization model. Based on that, we feel confident that it provides an enhanced decision support for multi user service selection problems. This was reinforced by our evaluation. Indeed, we were able to demonstrate the correctness of our optimization model as well as the applicability in a real-world scenario. Furthermore, our model could be solved in acceptable time for realistic problem sizes. Hence, we contribute to the current body of knowledge in multi user service selection by not only extending existing approaches, but rather by providing a first approach (cf. ❸) which considers ❶ and ❷.

Moreover, the application in the real-world scenario demonstrates that our approach can yield significant practical benefit. Indeed, taking relevant IUR into account, the results of the multi user service selection approach can provide enhanced decision support for its users. Such IUR are relevant in situations where dependencies between the service compositions of the participating users exist. For instance, in our real-world example of the tourism domain, user 5 requested to visit ‘English Garden’ together with user 3. This requires to consider possible waiting times in users’ service compositions to allow for a simultaneous start of the corresponding service object (i.e., ‘English Garden’). Indeed, such cases can be observed in our results (cf. section ‘*Practical Applicability*’). Here, the realization of the IUR led to an improved overall utility, while some participating users were confronted with waiting times or a change of the optimal service composition as compared to a situation without IUR. In conclusion, the consideration of such dependencies by means of IUR constitutes an important step to support processes where several users want to achieve their requests in a best possible way. For instance, in the field of emergency situations or disaster management, the optimal management of the users/resources available (e.g., fire fighters) is crucial to rescue people in need in a best possible way. Here, our approach can provide decision support concerning the coordination of fire fighters and their actions. In that sense, it can be more beneficial if certain actions (e.g., free persons trapped in a car) are conducted by fire fighters with common professional experiences which can be represented by IUR. Another example can be found in the field of fleets in forwarding companies. Here, situations exist where it is more valuable if two users (i.e., long distance drivers in this context) deliver their goods to the same client at the same time which can be also represented by IUR. Here, our approach can provide decision support as it allows for a realization of such IUR, precisely, a simultaneous start of a service object for two or more users while taking possible dependencies between the users’ service compositions into account. Besides the possibility to consider such dependencies by means of IUR, our findings reveal another important practical benefit. Depending on the problem size, the number of possible service compositions can get extremely large (cf. section ‘*Evaluation*’). In combination with dependencies between the users’ service compositions, this large number lead to great complexity when users need to decide which service composition should be used to execute a process. Our approach help to deal with this complexity as it determines the optimal service composition for each participating user considering his/her personal preferences and IUR. To sum up, we feel confident that our approach can serve as a profound decision support in order to support multi user processes in service-oriented decision support systems by means of service selection.

However, our approach is also subject to limitations. First, the application of our approach for the real-world scenario has shown that its computation time indeed seems practical at planning time instead of runtime. But as already noted, situations may exist where our approach goes along with an unacceptable computation time, due to the NP-hardness of the service selection problem. However, the goal of our study is not to present a runtime-optimized approach or heuristic. It is rather about the question how IUR can be considered within multi user service selection in a well-founded way. Nevertheless, future work is needed and intended to evaluate how existing heuristic techniques (cf. Alrifai et al. 2012; Li and Yan-xiang 2012; Yu et al. 2007) can be combined with our idea to consider IUR in multi user service selection. Furthermore, we have to recognize that there may be scenarios during runtime, where time (in terms of NFP *duration* and *waiting time*) needs to be considered on a fine-grained or quasi-continuous level. While we focused on planning time and modeled time as discrete, our approach would still be able to consider such runtime scenarios by adjusting the factor *c* accordingly (e.g., to the level of seconds). This, however, would increase the size of the selection problem (e.g., number of waiting services) and affect its computation time. In this case, time may also be modeled by means of continuous variables. This would require only one waiting service per waiting service class while potential waiting times are calculated dynamically. Further research has to examine, whether this kind of modeling has advantages regarding time complexity. Second, like other single and multi user service selection approaches (cf. Alrifai et al. 2012; Yu et al. 2007; Zeng et al. 2004), our approach requires certain input data with respect to parameterization (i.e., preferences regarding the different QoS attributes, values for the global end-to-end constraints, and IUR). Such input data may not always be available or known prior to service selection. To deal with this challenge, various estimation methods, historical data, and/or user surveys can be used. For instance, a simple estimation of the input data which does not require much effort nor data, can be conducted based on the rule of thumb or expert interviews (e.g., users in the tourism domain are usually willing to pay up to €80 for a city day trip, which can be used to recommend or determine an upper limit for the global end-to-end price). Indeed, using such estimation methods the applicability of the approach can be supported. Historical data may for example be applied in case several process realizations of the users are available (e.g., user profiles in a mobile application). Based on the respective data, it is for example possible to determine the users' preference values (e.g., to recommend or determine the willingness to pay for a city day trip based on previous realizations of similar processes). Finally, another possibility to deal with cold-start problems is to query or survey the necessary input data among the participating users. We used the latter possibility to determine the required input data for our real-world example (cf. section '*Practical Applicability*'). Third, our approach maximizes the accumulated utility of all users considering the individual users' preferences and global end-to-end constraints regarding the NFP. Thus, it has to be noted that for a user the resulting service composition could be inferior in terms of utility as compared to the solution of a service selection in a single user context. The reasons can be found in the realization of other users' IUR (cf. user 3 of our real-word example) resulting in a higher overall utility. However, to address this issue, an appropriate compensation is needed, which constitutes an encouraging starting point for further research.

Moreover, we see two further promising starting points for future research. First, in our model setup we focused on sequential processes. Dealing with multiple user processes, however, it is very likely that users either conduct different actions at the same time, which requires for a consideration of a parallel construct (cf. Alrifai et al. 2012; Ardagna and Pernici 2007), or need to decide which action to conduct, which requires for a pick (cf. Wan et al. 2008; Yu et al. 2007) or conditional construct (cf. Alrifai et al. 2012; Yu et al. 2007). In future research, we will work on considering these control flow structures. The use of execution routes (cf., e.g., Alrifai et al. 2012; Ardagna and Pernici 2007; Yu et al. 2007; Zeng et al. 2004) is a first promising step in this direction. Nevertheless, adaptions have to be made in order to make them work properly for multi user service selection. Second, in recent years, emerging mobile technologies enabled the capturing and processing of context information (e.g., GPS position) in a convenient way. In combination with user-defined requests, context information could be used to further enhance decision support. Here, methods of context-aware service selection (cf., e.g., Heinrich and Lewerenz 2015; Tao et al. 2010; Yu and Reiff-Marganiec 2009) can serve as a sound methodical foundation. Hence, in future research we will work on combining multi user service selection with context-aware service selection.

Overall, our research constitutes a first important step to consider user-defined requests in multi user service selection. Beyond that, we hope that it will open doors for further research in this exciting area.

References

Alrifai, M., and Risse, T. 2009. "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proceedings of the 18th International Conference on World Wide Web,* J. Quemada, G. León, Y. Maarek and W. Nejdl (eds.), Madrid, Spain, pp. 881–890.

Alrifai, M., Risse, T., and Nejdl, W. 2012. "A hybrid approach for efficient Web service composition with end-to-end QoS constraints," *ACM Transactions on the Web* (6:2), pp. 1–31.

Ardagna, D., and Mirandola, R. 2010. "Per-flow optimal service selection for Web services based processes," *Journal of Systems and Software* (83:8), pp. 1512–1523.

Ardagna, D., and Pernici, B. 2007. "Adaptive Service Composition in Flexible Processes," *IEEE Transactions on Software Engineering* (33:6), pp. 369–384.

Benouaret, K., Benslimane, D., and Hadjali, A. 2012. "Selecting Skyline Web Services for Multiple Users Preferences," in *IEEE 19th International Conference on Web Services (ICWS),* Honolulu, HI, USA, pp. 635–636.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2005. "An approach for QoS-aware service composition based on genetic algorithms," *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 1069–1075.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2008. "A framework for QoS-aware binding and re-binding of composite web services," *Journal of Systems and Software* (81:10), pp. 1754–1769.

Cui, L., Kumara, S., and Lee, D. 2011. "Scenario Analysis of Web Service Composition based on Multi-Criteria Mathematical Goal Programming," *Service Science* (3:4), pp. 280–303.

Dannewitz, C., Pentikousis, K., Rembarz, R., Renault, É., Strandberg, O., and Ubillos, J. 2008. "Scenarios and research issues for a network of information," in *4th International Mobile Multimedia Communications Conference*.

Delen, D., and Demirkan, H. 2013. "Data, information and analytics as services," *Decision Support Systems* (55:1), pp. 359–363.

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

DeRenzi, B., Borriello, G., Jackson, J., Kumar, V. S., Parikh, T. S., Virk, P., and Lesh, N. 2011. "Mobile phone tools for field-based health care workers in low-income countries," *The Mount Sinai journal of medicine, New York* (78:3), pp. 406–418.

Dong, C.-S. J., and Srinivasan, A. 2013. "Agent-enabled service-oriented decision support systems," *Decision Support Systems* (55:1), pp. 364–373.

Dovidio, J. F. 1984. "Helping Behavior and Altruism: An Empirical and Conceptual Overview," in *Advances in experimental social psychology*, L. Berkowitz (ed.), New York: Academic Press, pp. 361–427.

Fajardo, J. T. B., and Oppus, C. M. 2009. "A Mobile Disaster Management System Using the Android Technology," *International Journal of Communications* (3:3), pp. 78–86.

Fletcher, G. J. O., and Fitness, J. 1995. *Knowledge structures in close relationships*: *A social psychological approach*, Hillsdale, NY: Erlbaum .

Forgas, J. P. 1999. "Feeling and Speaking: Mood Effects on Verbal Communication Strategies," *Personality and Social Psychology Bulletin* (25:7), pp. 850–863.

García, J. M., Ruiz, D., Ruiz-Cortés, A., and Parejo, J. A. 2008. "QoS-Aware Semantic Service Selection: An Optimization Problem," *Proceedings of the 2008 IEEE Congress on Services - Part I*, pp. 384–388.

Gavalas, D., Konstantopoulos, C., Mastakas, K., and Pantziou, G. 2014. "Mobile recommender systems in tourism," *Journal of Network and Computer Applications* (39), pp. 319–333.

Gerpott, T. J., and Berg, S. 2011. "Determinants of the Willingness to Use Mobile Location-Based Services," *Business & Information Systems Engineering* (3:5), p. 279.

Google. 2013. *Our Mobile Planet: United States of America*. http://services.google.com/fh/files/misc/omp-2013-us-en.pdf. Accessed 15.01.16.

Han, X., Liu, Y., Xu, B., and Zhang, G. 2011. "A Survey on QoS-Aware Dynamic Web Service Selection," in *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Wuhan, China, pp. 1–5.

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

Heider, F. 1958. *The psychology of interpersonal relations*, New York: Wiley.

Heinrich, B., Klier, M., Lewerenz, L., and Zimmermann, S. 2015a. "Quality-of-Service-Aware Service Selection: A Novel Approach Considering Potential Service Failures and Nondeterministic Service Values," *Service Science* (7:1), pp. 48–69.

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

Heinrich, B., and 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* (24:4), pp. 406–432.

Heinrich, B., and Schön, D. 2015. “Automated Planning of context-aware Process Models,” in *European Conference on Information Systems (ECIS),* Münster, Germany. May 26-29.

Henneberger, M., Heinrich, B., Lautenbacher, F., and Bauer, B. 2008. “Semantic-Based Planning of Process Models,” *Proceedings of the 2008 Multikonferenz Wirtschaftsinformatik*, pp. 1677–1689.

Hinkelmann, K., Maise, M., and Thonssen, B. 2013. “Connecting enterprise architecture and information objects using an enterprise ontology,” in *Enterprise Systems Conference (ES),* Cape Town, South Africa, pp. 1–11.

Holmes, J. G., and Rempel, J. K. 1989. “Trust in close relationships,” in *Review of Personality and Social Psychology,* C. Hendrick (ed.), London: Sage Ltd, pp. 187–220.

Jin, H., Zou, H., Yang, F., Lin, R., and Shuai, T. 2012a. “A Novel Method for Optimizing Multi-User Service Selection,” *Journal of Convergence Information Technology* (7:5), pp. 296–310.

Jin, H., Zou, H., Yang, F., Lin, R., and Shuai, T. 2012b. “Using Bipartite Graph for Resolving Multiple Requests Conflicts,” in *International Joint Conference on Service Sciences (IJCSS),* Shanghai, China, pp. 46–50.

Kang, G., Liu, J., Tang, M., Liu, X., and Fletcher, K. K. 2011. “Web Service Selection for Resolving Conflicting Service Requests,” in *IEEE International Conference on Web Services,* Washington, DC, pp. 387–394.

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

Li, W., and Yan-xiang, H. 2012. “A Web Service Composition Algorithm Based on Global QoS Optimizing with MOCACO,” in *Proceedings of the 2011, International Conference on Informatics, Cybernetics, and Computer Engineering*: *Volume 2: Information Systems and Computer Engineering*, L. Jiang (ed.), Berlin, Heidelberg: Springer-Verlag GmbH, pp. 79–86.

Liang, Z., Zou, H., Guo, J., Yang, F., and Lin, R. 2013. “Selecting Web Service for Multi-user Based on Multi-QoS Prediction,” in *IEEE International Conference on Services Computing (SCC),* Santa Clara, CA, USA, pp. 551–558.

Lin, M., Xie, J., Guo, H., and Wang, H. 2005. “Solving Qos-Driven Web Service Dynamic Composition as Fuzzy Constraint Satisfaction,” in *IEEE International Conference on e-Technology, e-Commerce and e-Service,* Hong Kong, China, pp. 9–14.

Martial, F. von. 1992. *Coordinating Plans of Autonomous Agents,* Berlin, Heidelberg: Springer Berlin Heidelberg.

Monares, Á., Ochoa, S. F., Pino, J. A., Herskovic, V., Rodriguez-Covili, J., and Neyem, A. 2011. “Mobile computing in urban emergency situations: Improving the support to firefighters in the field,” *Expert Systems with Applications* (38:2), pp. 1255–1267.

Nagata, M., Murata, Y., Shibata, N., Yasumoto, K., and Ito, M. 2006. “A Method to Plan Group Tours with Joining and Forking,” in *Simulated evolution and learning*: *6th international conference, SEAL 2006, Hefei, China, October 15 - 18, 2006 ; proceedings,* T.-D. Wang (ed.), Berlin: Springer, pp. 881–888.

Nemhauser, G. L., and Wolsey, L. A. 1988. *Integer and combinatorial optimization*, New York: Wiley.

O'Sullivan, J., Edmond, D., and ter Hofstede, A. 2002. "What's in a Service? Towards Accurate Description of Non-Functional Service Properties," *Distributed and Parallel Databases* (12:2/3), pp. 117–133.

Picoto, W. N., Bélanger, F., and Palma-dos-Reis, A. 2014. "An organizational perspective on m-business: Usage factors and value determination," *European Journal of Information Systems* (23:5), pp. 571–592.

Pisinger, D. 1995. *Algorithms for Knapsack Problems*. Ph.D. dissertation, Copenhagen.

Pruitt, D. G., and Carnevale, P. J. 1993. *Negotiation in social conflict*, Buckingham: Open University Press.

Salovey, P., Mayer, J. D., and Rosenhan, D. L. 1991. "Mood and helping: Mood as a motivator of helping and helping as a regulator of mood," in *Prosocial behavior*, M. S. Clark (ed.), Newbury Park, CA: Sage Ltd, pp. 215–237.

Schutte, N. S., Malouff, J. M., Bobik, C., Coston, T. D., Greeson, C., Jedlicka, C., Rhodes, E., and Wendorf, G. 2001. "Emotional intelligence and interpersonal relations," *The Journal of social psychology* (141:4), pp. 523–536.

Shen, Y., Yang, X., Wang, Y., and Ye, Z. 2012. "Optimizing QoS-Aware Services Composition for Concurrent Processes in Dynamic Resource-Constrained Environments," in *IEEE 19th International Conference on Web Services (ICWS),* Honolulu, HI, USA, pp. 250–258.

Sun, S. X., and Zhao, J. 2012. "A decomposition-based approach for service composition with global QoS guarantees," *Information Sciences* (199), pp. 138–153.

Surianarayanan, C., Ganapathy, G., and Ramasamy, M. S. 2015. "An approach for selecting best available services through a new method of decomposing QoS constraints," *Service Oriented Computing and Applications* (9:2), pp. 107–138.

Tao, F., Zhao, D., Yefa, H., and Zhou, Z. 2010. "Correlation-aware resource service composition and optimal-selection in manufacturing grid," *European Journal of Operational Research* (201:1), pp. 129–143.

Vescoukis, V., Doulamis, N., and Karagiorgou, S. 2012. "A service oriented architecture for decision support systems in environmental crisis management," *Future Generation Computer Systems* (28:3), pp. 593–604.

Vos, H. de, Haaker, T., and Teerling, M. 2008. "Consumer Value of Context Aware and Location Based Mobile Services," *BLED 2008 Proceedings*, pp. 50–62.

Wan, C., Ullrich, C., Chen, L., Huang, R., Luo, J., and Shi, Z. 2008. "On Solving QoS-Aware Service Selection Problem with Service Composition," in *Seventh International Conference on Grid and Cooperative Computing*, J. Fan (ed.), Shenzhen, Guangdong, China, Los Alamitos, California: IEEE Computer Society, pp. 467–474.

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

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., and Li, P. 2010. "Web Service Selection for Multiple Agents with Incomplete Preferences," in *IEEE/ACM International*

Conference on Web Intelligence-Intelligent Agent Technology, Toronto, AB, Canada, pp. 565–572.

Wang, S., Hsu, C.-H., Liang, Z., Sun, Q., and Yang, F. 2014. “Multi-user web service selection based on multi-QoS prediction,” *Information Systems Frontiers* (16:1), pp. 143–152.

Yu, H. Q., and Reiff-Marganiec, S. 2009. “A Backwards Composition Context Based Service Selection Approach for Service Composition,” in *IEEE International Conference on Services Computing*, Bangalore, India, pp. 419–426.

Yu, T., Zhang, Y., and Lin, K.-J. 2007. “Efficient algorithms for Web services selection with end-to-end QoS constraints,” *ACM Transactions on the Web* (1:1).

Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., and Chang, H. 2004. “QoS-aware middleware for Web services composition,” *IEEE Transactions on Software Engineering* (30:5), pp. 311–327.

3 Paper 2: Multi-User Service Re-Selection: React Dynamically to Events Occurring at Process Execution

Status	Published	Full Citation
Accepted	06/2017	Mayer, M. 2017. “MULTI-USER SERVICE RE-SELECTION: REACT DYNAMICALLY TO EVENTS OCCURRING AT PROCESS EXECUTION,” in *25th European Conference on Information Systems (ECIS)*, Guimarães, Portugal.

Post publication changes:

- The format of the keywords was changed for consistency reasons
- In the whole paper, “e.g.” was changed to “for example” for consistency reasons when used within a sentence
- In the whole paper, the comma setting for “e.g.”, “i.e.” and “cf.” was adjusted for consistency reasons

----

Abstract

Considering service-based processes, the problem of determining the service candidates that fit best to a user’s target weights and requirements regarding certain non-functional properties is known as QoS-aware service selection problem. Referring to multi-user processes, this requires taking into account several users with their individual goals. In this regard, users could also have preferences in the sense of user-defined requests referring to other users, so-called Inter-User-Requests (IUR). Such IUR result in dependencies among different users’ service compositions that have to be taken into account when selecting services. However, due to the dynamic environment in which services are used certain events – like the failure of a service – may occur during process execution that require service re-selection at runtime. In this work, we provide such a service re-selection approach in terms of an optimization model that considers multiple users and dependencies resulting from IUR. Moreover, for the temporal coordination of the users – necessary for time-dependent IUR – we further propose a continuous time concept and integrate that in our model. Supported by our evaluation, we feel confident that this approach can serve as a first step for a comprehensive multi-user service re-selection approach where dependencies among users exist.

Keywords: Decision support, Service re-selection, Multi-user processes, Dependencies.

1 Introduction

Service-orientation as IT-architectural paradigm has received great attention in the last decade (Barry 2012; Weinhardt et al. 2011). Encapsulating clearly defined functionalities to modular designed services enables more flexible systems (Papazoglou et al. 2007) and allows to compose single services in order to realize or support complex business processes (Sheng et al. 2014). In this regard, the continuing increase in the number of available services (e.g., currently almost 16,000 services available only on *programmableweb.com* and over 3,000 services on *appexchange.salesforce.com*) results in a growing number of functional equivalent services that differ only in their non-functional properties (NFP). These properties are represented by Quality-of-Service (QoS) criteria such as costs, duration or availability (Alrifai et al. 2012). This development might be additionally reinforced by the recent rise of the microservice architectural style which postulates to design applications as independently deployable services (Lewis and Fowler 2014). As a consequence, there exists a decision problem that is related to the question which services fit best to each single action (i.e., service class) of the underlying process for a certain user (Alrifai et al. 2012). For this, the user's individual target weights and requirements (e.g., constraints with respect to the end-to-end process) regarding the NFP can be taken into account (Zeng et al. 2004).

The problem of determining the optimal service composition is known as QoS-aware service selection problem and has been widely discussed in literature for single user processes (e.g., Alrifai and Risse 2009; Ardagna and Mirandola 2010; Yu et al. 2007; Zeng et al. 2004) as well as for multi-user processes (e.g., Benouaret et al. 2012; He et al. 2012; Heinrich et al. 2015a; Kang et al. 2011; Wang et al. 2010). Multi-user processes require to deal with several users and their individual target weights and requirements. Furthermore, there may also exist dependencies among the service compositions of different users that have to be taken into account in multi-user service selection. In this respect, existing multi-user approaches consider dependencies resulting from hard restrictions (such as predetermined capacity limits) on the one side (e.g., Benouaret et al. 2012; He et al. 2012; Kang et al. 2011; Wang et al. 2010) and user preferences on the other side. Such user preferences could be user-defined requests referring to other users, so-called Inter-User-Requests (IUR) (Heinrich et al. 2015a), for instance, a user wants to use (or not) a specific service together with certain other users.

However, due to the dynamic environment in which services are used (cf., e.g., Sheng et al. 2014), services selected at planning time may, for instance, take longer than expected, become unavailable or even fail during their execution (cf. Canfora et al. 2008; Sheng et al. 2014; Zheng et al. 2014). Therefore, there may exist new optimal service compositions at runtime or the initially planned service compositions may even be infeasible. For instance in case of a service failure, there may possibly exist numerous alternative service candidates a user could select as substitute. The user would then also need to consider the already executed part of the process as well as the remaining part to ensure the new service composition is still feasible (e.g., due to the user's NFP constraints). As a result, the user might see her-/himself being confronted with an information overload problem (cf. Shen et al. 2012a) – where decision support could be provided by a suitable service re-selection approach. Although there exist some service selection approaches which consider the effects of potential service failures already at planning time (e.g., Heinrich et al. 2015b; Yu and Lin 2005), a dynamic re-selection approach that reacts to service failures and other events at runtime is still necessary. Additionally, in the context of multi-user processes, events occurring for one user may also influence other users' process execution due to existing dependencies among the users. However, existing optimization-based service re-selection approaches consider only single user processes and therefore no such dependencies (e.g., Berbner et al. 2007; Canfora et al. 2008; Li et al. 2011; Sandionigi et al. 2013).

Thus, the aim of this work is to propose a novel multi-user service re-selection approach in terms of an optimization model that allows to react dynamically to events occurring at process execution under consideration of dependencies among different users' service compositions (contribution ❶). Here, we focus on dependencies resulting from user preferences in the sense of IUR. Moreover, we distinguish mutual (time-independent) and simultaneous (time-dependent) IUR. For the consideration of the latter a temporal coordination of the users' actions is required. For this purpose, we need to develop a concept dealing with time as continuous (contribution ❷) since in service re-selection at runtime events could occur at any time. By this, our work refers to the following research questions:

How to design a dynamic optimization-based multi-user service re-selection approach that is capable of considering the effects of events occurring at process execution? How to integrate a continuous time concept within this multi-user service re-selection?

The remainder of this paper is structured as follows: In the next section, we analyze and discuss the existing literature related to the identified research gap and our contribution. This is followed by the introduction of our model setup including the definition of IUR. Based on that, we develop the continuous time concept and integrate this in our novel multi-user service re-selection approach in Section 5. In Section 6, we then provide an evaluation of this approach. We conclude our work with a short discussion on limitations and future research.

2 Related Literature, Research Gap and Contribution

Since our research is related to the consideration of dependencies among different users in multi-user service selection problems, we first analyze the existing approaches in that field, before we discuss the identified research gap and our contribution with respect to literature on QoS-aware service re-selection.

As already described, with multiple users participating in a process the common single user service selection problem is extended by the consideration of potential dependencies among different users' service compositions. Existing works could be divided in approaches focussing on hard restrictions and approaches considering user preferences when determining the (near) optimal service compositions for all users under consideration of their individual target weights and requirements regarding the NFP. In terms of hard restrictions, which have to be satisfied in a feasible service composition, this particularly refers to the consideration of capacity limits of services (e.g., He et al. 2012; Jin et al. 2012; Kang et al. 2011; Shen et al. 2012b) as well as the mandatory mutual use of a certain service by several users (e.g., Benouaret et al. 2012; Wanchun et al. 2011; Wang et al. 2010). On the other side, user preferences obviously affect the utility and thus the optimality of a service composition. Here, to the best of our knowledge only Heinrich et al. (2015a) provide an approach that enables to consider user preferences, so-called user-defined requests referring to other users (IUR). The multi-user service selection approach proposed by them utilizes an optimization model formulated as knapsack problem for selection of the optimal service compositions for all users *at planning time* while taking into account the dependencies resulting from IUR. Furthermore, for consideration of simultaneous IUR they suggest a concept dealing with *time as discrete* by introducing special waiting service classes and waiting services.

In this work, we focus on multi-user service re-selection at runtime, which means the consideration of events occurring during execution of the initially planned service compositions. Re-selection may be required or appropriate, for instance, after the failure of services or the appearance of new services (cf. Ardagna and Pernici 2007; Canfora et al. 2008), the deviation of realized from expected NFP values (cf. Canfora et al. 2008; Shen et al. 2012b), or users redefining their target weights (cf. Ardagna and Pernici 2007) and requirements (cf. Shen et al. 2012b). Referring to multi-user processes, such events could also be users leaving the process or the participation of new users. Besides approaches that consider potential service failures already at planning time (e.g., Heinrich et al. 2015b; Yu and Lin 2005) or following a certain fault-tolerant strategy (e.g., Shen et al. 2012b; Stein et al. 2009; Zheng and Lyu 2010), several optimization-based service re-selection approaches have been proposed to deal with unforeseen events occurring at runtime. In this respect, they aim at enabling successful process completion by determining a new feasible, (near) optimal service composition for a single user. While Ardagna and Pernici (2007), Sandionigi et al. (2013) and Zeng et al. (2004) simply suggest to apply their proposed optimization model regarding service selection at planning time on the remaining part of the process in case of an event at runtime, Berbner et al. (2007), Canfora et al. (2008), Li et al. (2011) and Lin et al. (2010) provide independent service re-selection approaches. Li et al. (2011) and Lin et al. (2010), for instance, propose an iterative approach which gradually expands the part of the process considered in re-selection and thus trying to find a solution that does not violate the constraints regarding the NFP. On the other side, both Berbner et al. (2007) and Canfora et al. (2008) provide a heuristic approach applied on the whole remaining process that determines the new near optimal service composition for the user. Regarding this, the algorithm H1_RELAX_IP of Berbner et al. (2007) uses the LP relaxation of the original mixed integer problem combined with a backtracking algorithm, while Canfora et al. (2008) developed an approach based on a genetic algorithm. Besides that, in the field of semantic web services, existing approaches (e.g., Klusch and Kapahnke 2012; Li et al. 2008; Rodriguez-Mier et al. 2012) could possibly support the QoS-aware service re-selection by automatically discovering new functionally equivalent services, for instance, in case the currently executed service fails.

Research Gap and Contribution to Research

According to Ardagna and Pernici (2007), a valid re-selection to determine the new optimal service composition after occurrence of a (runtime) event in single user problems seems to be to simply apply a service selection approach again on the remaining part of the process. However, using this idea especially in the context of multi-user processes and IUR would not necessarily lead to a feasible and optimal solution: On the one hand, the impact of the occurred event itself could not be considered correctly which may lead to infeasible service compositions (e.g., realized NFP of a failed service affect users' constraints). On the other hand, dependencies existing between different users' services located in the already executed and the remaining part of the process would be disregarded. Thus, when considering IUR and the resulting dependencies, the entire initial process has to be taken into account. Furthermore, events – although directly related only to one user – might also affect other users' service compositions (e.g., a user leaving the process). In this respect, there could be IUR that are planned to be realized in the initial service compositions but will not be realized due to unforeseen events, or the other way round. We therefore aim at providing a service re-selection approach that – after occurrence of an event at runtime – considers for all users the already executed (or currently executing) services and realized NFP as well as the still unexecuted actions of the remaining part of the process (contribution ❶).

Furthermore, using a discrete time concept – as proposed in (Heinrich et al. 2015a) – for the temporal coordination of the users' actions might be sufficient in many situations, but in service re-selection at runtime this would be accompanied by some serious weaknesses regarding flexibility and performance (see Section 4.1). Because of this, we propose a concept that enables to consider time as continuous (contribution ❷). As to the best of our knowledge, there exists no work within service science describing an optimization model that would allow to consider multiple users with dependencies among them where the model also contains a continuous time concept for temporal coordination of the users, we develop such an optimization model in this work.

3 Model Setup

In the following, we introduce our model setup in terms of those definitions and modeling elements that can serve as common knowledge base. This allows for a better differentiation between existing knowledge and our contribution later on.

In this work, we consider a process with multiple participating users $a \in A$. More precisely, a process consists of a number of actions or service classes i (with $i = 1$ to I), respectively, that contribute to achieve an intended goal. Each service class encompasses a set of functional equivalent service candidates s_{ij} that only differ in their non-functional properties (NFP) represented by Quality-of-Service (QoS) attributes (e.g., costs, duration, availability). Furthermore, a service composition is defined as a concrete implementation of a process in terms of a set of services with exactly one service candidate out of each service class of the process.

3.1 NFP and Utility Function

We further define NFP as the set of attributes N that have to be considered in service selection or re-selection, respectively. This set N can be divided into the subset of attributes N^- that need to be minimized, the subset of attributes N^+ that need to be maximized, and the subset of attributes N^{tv} that refer to a target value tv. Further, we introduce the vector $q_{ij} = [q_{ij}^1, \dots, q_{ij}^N]^T$ of the quantified NFP values of a service candidate s_{ij}. When selecting service candidates with several NFP, we use – in line with the existing literature – a utility function which aggregates the values of the different attributes N to a single utility value U (cf., e.g., Ai and Tang 2008; Alrifai and Risse 2009; Ardagna and Pernici 2007; Yu et al. 2007). In our work, we apply the utility function described, for example, by Alrifai and Risse (2009), which is based upon the simple additive weighting (SAW) technique: First, the values $q_{ij}^{\propto}$ of all attributes $\propto \in N$ are normalized in the interval [0;1] to enable comparability between different attributes. For this, the aggregated maximum $P_{max}^{\propto}$ and minimum $P_{min}^{\propto}$ of the attributes N^- and N^+ – and $P_{max^*}^{\propto}$, $P_{min^*}^{\propto}$ for N^{tv} – over all service classes S_i are used, which can be calculated as follows:

$$P_{max}^{\propto} = \sum_{i=1}^{I} P_{i,max}^{\propto} = \sum_{i=1}^{I} \max_{s_{ij} \in S_i} q_{ij}^{\propto}\,; \quad P_{min}^{\propto} = \sum_{i=1}^{I} P_{i,min}^{\propto} = \sum_{i=1}^{I} \min_{s_{ij} \in S_i} q_{ij}^{\propto} \tag{1}$$

$$P_{max^*}^{\propto} = \sum_{i=1}^{I} P_{i,max^*}^{\propto} = \sum_{i=1}^{I} \max_{s_{ij} \in S_i} \left(\left| q_{ij}^{\propto} - tv^{\propto} \right| \right); \quad P_{min^*}^{\propto} = \sum_{i=1}^{I} P_{i,min^*}^{\propto} = \sum_{i=1}^{I} \min_{s_{ij} \in S_i} \left(\left| q_{ij}^{\propto} - tv^{\propto} \right| \right) \tag{2}$$

The utility score of a single service candidate could then be determined by taking the weighted sum over all attributes under consideration of user-defined target weights regarding the attributes N. With multiple participating users each user $a \in A$ may possibly have its individual target weights $w_a^{\propto}$ (with $\sum_{\propto=1}^{N} w_a^{\propto} = 1$), leading to varying utility scores $U_{a_{ij}}$ for the same service candidate s_{ij} but different user a.

$$\begin{aligned} U_{a_{ij}} = & \sum_{\propto \in N^-} \left(\frac{P_{i,max}^{\propto} - q_{ij}^{\propto}}{P_{max}^{\propto} - P_{min}^{\propto}} \right) * w_a^{\propto} + \sum_{\propto \in N^+} \left(\frac{q_{ij}^{\propto} - P_{i,min}^{\propto}}{P_{max}^{\propto} - P_{min}^{\propto}} \right) * w_a^{\propto} \\ & + \sum_{\propto \in N^{tv}} \left(\frac{P_{i,max^*}^{\propto} - \left(\left| q_{ij}^{\propto} - tv \right| \right)}{P_{max^*}^{\propto} - P_{min^*}^{\propto}} \right) * w_a^{\propto} \end{aligned} \tag{3}$$

By summing up the utility scores of all selected services by all users the overall utility value of a service composition could be determined. In line with existing optimization-based approaches, we formulate our optimization model provided in Section 4 as knapsack problem (e.g., Alrifai et al. 2012; Strunk 2010; Yu et al. 2007). Thus, it consists of an objective function determining the overall utility value and several constraints, for instance, to integrate the users' global end-to-end requirements regarding the NFP, which can be described by the vector $Q_a = [Q_a^1, \dots, Q_a^N]^T$. In this respect, we use decision variables $x_{a_{ij}}$ for each user $a \in A$ and every service candidate s_{ij}, with $x_{a_{ij}} = 1$ indicating that service candidate s_{ij} is selected for user a, and $x_{a_{ij}} = 0$ that is not.

3.2 Considering Inter-User-Requests

According to (Heinrich et al. 2015a), IUR are specified as user-defined requests referring to other users. In contrast to hard restrictions as considered, for example, in (Benouaret et al. 2012; He et al. 2012; Kang et al. 2011; Wang et al. 2010), IUR represent user preferences assessing different alternatives, for example, using a certain service together with defined other users or not. Generally, an IUR is defined by a user who determines the service or service class related to that IUR and the set of participating users. In scenarios where a user does not know all other users in the process, the user could instead describe the participating users of an IUR by certain characteristics such as age, gender or interests in terms of persons as users and industry branch, country or company size in terms of companies. Based on the described characteristics, the corresponding group of users could be identified and connected to that IUR. Furthermore, a user associates a certain (positive or negative) value with the realization of an IUR. In line with Heinrich et al. (2015a), we distinguish four basic types of IUR, regarding the two dimensions relation and time:

	Complementary	**Conflicting**
Mutual (time-independent)	*Complementary mutual usage:* • A user requests to perform an action **together** with one or more other users. • A **positive value** is associated with this IUR.	*Conflicting mutual usage:* • A user requests **not** to perform an action together with one or more other users. • A **negative value** is associated with this IUR.
Simultaneous (time-dependent)	*Complementary simultaneous usage:* • A user requests to perform and thus to start an action **together** with one or more users **at the same time**. • Potential occurrence of **waiting times** for users. • A **positive value** is associated with this IUR.	*Conflicting simultaneous usage:* • A user requests **not** to perform an action together with one or more other users at **any given moment of time**. • Potential occurrence of **waiting times** for users. • A **negative value** is associated with this IUR.

Table 1. Fundamental types of IUR (Heinrich et al. 2015a)

Based on that, each user may specify a set of IUR E_a^{IUR}, where a single IUR $e \in E_a^{IUR}$ could be formally defined by the following quadruplet:

$$e = (\widehat{U}_e, \overline{\overline{U}}_e, A_e, X_e) \quad (4)$$

An IUR e, thereby, is defined by means of the utility $\widehat{U}_e$ (which is distinct from 0 in case e is a mutual IUR), the utility $\overline{\overline{U}}_e$ (which is distinct from 0 in case e is a simultaneous IUR), the set of participating users A_e and the set X_e of corresponding decision variables $x_{a_{ij}}$.

Furthermore, for the consideration of simultaneous (i.e., time-dependent) IUR a temporal coordination of the users' actions is needed. In this respect, it may possibly be more beneficial for a user to wait a certain amount of time, for instance, to realize the positive utility associated with a complementary simultaneous IUR or to avoid the negative utility associated with a conflicting simultaneous IUR. This requires a concept to consider potential waiting times as well as the loss of utility caused by waiting in the corresponding optimization model. More precisely, there has to be the possibility for a user to wait right between two succeeding actions. For this, Heinrich et al. (2015a) propose a concept which considers time (in terms of duration, response time, etc.) and waiting time of a service selection problem as discrete. Particularly, they introduce waiting time WT as additional NFP and special waiting service classes S_i^* right in front of each regular service class S_i, with each waiting service class encompassing a set of waiting services s_{ij}^*. Attributes representing "time" (i.e., duration/ response time Dur, waiting time WT) are modeled as discrete values $q_{ij}^{Dur}, q_{ij}^{WT} \in \{k * c | \; c \in \mathbb{R}^+\}$ with $k \in \{0, 1, \ldots, K\}$ and thus in discrete steps: Every waiting service class consists of a defined number of waiting services, each being described by a different specific waiting time q_{ij}^{WT} and thus related to a different utility score as the utility is calculated similar to regular service candidates. However, values q_{ij}^{Dur} representing NFP duration, response time, etc. must also fit to these discrete time steps. The parameter c specifies the fixed length of each time interval. For example, making c smaller results in more discrete steps necessary to cover the same overall time range in the corresponding optimization model. Thereby, the optimization model evaluates the different alternatives (e.g., realization of $\bar{\bar{\bar{U}}}_e$ vs. loss of utility through waiting) and determines the right waiting service candidate, that means the right (discrete) amount of waiting, for each waiting service class and user of the process.

Moreover, as IUR affect more than one user, the consideration of IUR results in dependencies between different users' service compositions. In the case of simultaneous IUR, these dependencies are also of temporal nature. Regarding the formulation of a service selection problem as optimization model, the dependencies resulting from mutual IUR can be integrated directly in the objective function (of a non-linear model), whereas temporal-based dependencies related to simultaneous IUR require additional constraints (cf. Heinrich et al. 2015a). By solving such an optimization model, the initial optimal service compositions for all users regarding a certain multi-user service selection problem can be determined.

4 Novel Multi-User Service Re-Selection Approach

In this section, we present our service re-selection approach which enables to consider multiple users with their IUR when re-optimizing the users' initial service compositions after occurrence of a certain event during process execution. Subject to the event, there can be distinguished three general complementary goals for performing service re-selection (cf. Berbner et al. 2007):

(1) *Recovery*: To enable successful process completion for a user (e.g., after failure of a service).
(2) *Feasibility*: To ensure the selected service composition for each user is feasible (e.g., in case the realized NFP differ significantly from the ex-ante expected values).

(3) *Optimality*: To ensure the optimal service composition is selected for each user (e.g., after failure of a service or a user leaving the process).

Thus, the overall objective is to make certain that after occurrence of an event the process execution is still possible and feasible for all users – under the consideration that possibly new optimal service compositions for the individual users may exist. In order to take all global end-to-end constraints of the users and potential dependencies resulting from mutual and simultaneous IUR into account, it is necessary to adopt a global perspective, that means to consider the entire process, for re-optimization of the remaining part.

In the next paragraph, we therefore propose a continuous time concept enabling the consideration of simultaneous IUR and the resulting temporal-based dependencies in re-selection (contribution ❷). Based on that, we then describe how to integrate multiple users, IUR and this concept in an optimization model for service re-optimization of the remaining part of the process and all users (contribution ❶). For a discussion on the technical aspects regarding detection and triggering service re-selection we refer, for example, to Canfora et al. (2008), Lin et al. (2010) or Shen et al. (2012b).

4.1 Concept for Continuous Consideration of Time

For the temporal coordination of the users' actions required for consideration of simultaneous IUR we theoretically could adopt the existing discrete time concept described in Section 3.2. In this case, for each service selection problem a suitable length for the discrete time intervals (i.e., parameter c) would have to be defined when building the optimization model. In terms of service selection at planning time rather large time intervals seem to be sufficient in most cases (cf. Heinrich et al. 2015a). But the premises change when considering service re-selection at runtime as the execution of services or waiting could be disrupted at any given moment, for example, a service could fail at any time. Furthermore, the actual realized execution time for a service and actual waiting times do generally differ from the discrete values used for service selection at planning time. Thus, applying the described discrete time concept in a re-selection approach would require to specify much smaller intervals compared to service selection at planning time. However, this would result in problems regarding

Flexibility: What is the optimal choice for the length of a time interval (i.e., parameter c)?

Performance: Smaller time intervals normally correspond to larger problem sizes and thus higher computation times for calculating the optimal solution.

In the following, we therefore propose a concept to integrate (waiting) time as continuous in an optimization model: Usually, the utility of a user's service composition is determined by adding up the a-priori calculated utilities of the selected service candidates in the objective function of the optimization model (cf. Section 3.1). Using the described utility function the utility of a user's service composition can also be calculated during solving of the optimization model based on the aggregated NFP values of the service composition as the following equation illustrates:

$$\sum_{a \in A} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} U_{a_{ij}} x_{a_{ij}} = \sum_{a \in A} \sum_{\propto \in N} w_a^{\propto} \frac{P_{max}^{\propto} - \left(\sum_{i=1}^{I} \sum_{s_{ij} \in S_i} q_{ij}^{\propto} x_{a_{ij}} \right)}{P_{max}^{\propto} - P_{min}^{\propto}} \tag{5}$$

In contrast to calculate the utility of a service candidate a-priori based on predetermined NFP values, the term on the right allows to use a variable instead of a fixed value $q_{ij}^{\propto}$ for an attribute $\propto$ where the optimal value is then determined by the optimization model. Because of that, waiting time can be modeled as continuous by using variables $wt_{a_i} \in \mathbb{R}_0^+$ for NFP waiting time and let the optimization model dynamically determine the right amount of waiting time wt_{a_i} and the corresponding utility during solving of the model. Particularly, we connect a waiting time variable wt_{a_i} with each service class i and each user a (cf. Figure 1).

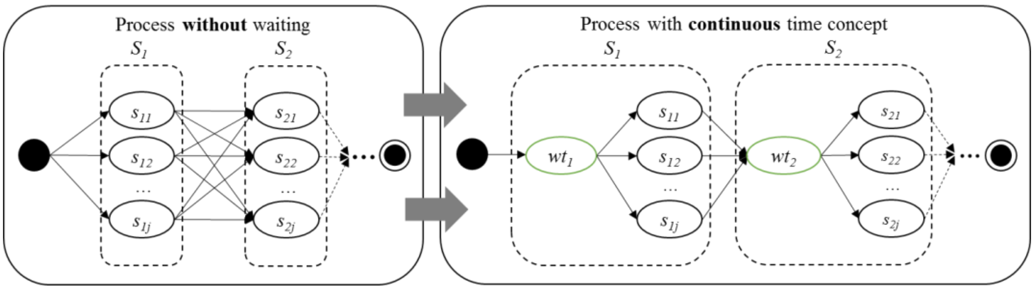

Figure 1. Illustration of process with continuous time concept for a single user

By this, there is no need to prescribe fixed, discrete values for waiting when building the optimization model. Consequently, there is also no restraint on certain discrete values for attributes representing duration/ response time, and thus $q_{ij}^{Dur} \in \mathbb{R}^+$. Therefore, the continuous concept overcomes the problem regarding *flexibility* related to the discrete time concept. Furthermore, it generally also *outperforms* the existing concept in terms of computation time required for calculating the optimal solution (cf. Section 5.1).

4.2 Optimization Model for Service Re-Selection

Hereafter, we introduce our optimization model for multi-user service re-selection. In case the re-selection is triggered (after occurrence of an event), the process can be divided into three regions for each user (cf. Zeng et al. 2004): Region (A) of already completely executed services and waited times, region (B) of currently being executed services or waiting, and region (C) of still unexecuted but planned services and waiting times. To be able to consider dependencies resulting from IUR that may exist among users' services/ service classes in different regions, we formulate the corresponding optimization model for the entire process taking into account regions (A)-(C).

Moreover, as a service could fail during its execution, we also have to take a possible "consume" of NFP (e.g., time) into consideration. Focusing, for instance, on attributes representing time this means that although the failed service could not be executed completely, the amount of time until detection of the failure is nevertheless consumed. As this affects the global end-to-end NFP constraints and also the temporal coordination of users' unexecuted actions, we therefore add to our model the variables $CA_{a_i}^{\propto}$ for each user a and each service class i that holds the consumed amount for each attribute $\propto \in N \backslash \{WT\}$[11].

In the following, we describe our non-linear optimization model for the consideration of multiple users and IUR in service re-selection which also integrates the proposed concepts for continuous time and consumed NFP. It is formulated as knapsack problem, consisting of an objective function and several constraints: The objective function of our model determines the accumulated maximum utility over all users $a \in A$, all service classes S_i and all service candidates s_{ij} by taking into account the binary decision variables $x_{a_{ij}}$ and s_e as well as the continuous waiting time variables wt_{a_i}:

$$\max_{x_{a_{ij}};wt_{a_i};s_e} \sum_{a\in A} \sum_{\substack{\propto \in N \\ \backslash\{WT\}}} w_a^{\propto} \frac{P_{max}^{\propto} - \left(\sum_{i=1}^{I}\left(\sum_{s_{ij}\in S_i} q_{ij}^{\propto} x_{a_{ij}} + CA_{a_i}^{\propto}\right)\right)}{P_{max}^{\propto} - P_{min}^{\propto}} + \sum_{a\in A} w_a^{WT} \frac{P_{a,max}^{WT} - \sum_{i=1}^{I} wt_{a_i}}{P_{a,max}^{WT} - P_{a,min}^{WT}}$$
$$+ \sum_{a\in A} \sum_{e\in E_a^{IUR}} \widehat{U}_e \prod_{x_{a_{ij}}\in X_e} x_{a_{ij}} + \sum_{a\in A} \sum_{e\in E_a^{IUR}} \overline{\overline{U}}_e s_e \prod_{x_{a_{ij}}\in X_e} x_{a_{ij}} \tag{6}$$

More precisely, the first summand calculates the utility of the users' service compositions based on the aggregated NFP values – including possibly consumed NFP $CA_{a_i}^{\propto}$, but without waiting time WT – of the selected service candidates ($x_{a_{ij}} = 1$ indicates that service candidate s_{ij} is selected for user a, $x_{a_{ij}} = 0$ that is not). The second summand determines the utility subject to the amount of waiting time WT for all users, the third and fourth summand refer to determining the utility regarding IUR. Moreover, the utility $\widehat{U}_e$ of a mutual IUR $e \in E_a^{IUR}$ is realized if – and only if – all decision variables $x_{a_{ij}} \in X_e$ are 1, that means, all corresponding service candidates s_{ij} have to be part of the solution. For the realization of $\overline{\overline{U}}_e$ of a simultaneous IUR in addition the indicator variable s_e has to be 1. This variable s_e is linked to the following constraints which evaluate whether the temporal conditions associated with a complementary (7) or conflicting (8) simultaneous IUR are fulfilled or not:

$$\left[\begin{array}{l} \max\limits_{\substack{a\in A_e| \\ \{x_{a_{i'j'}}\in X_e\}}} \left(\sum_{i=1}^{i'-1}\left(wt_{a_i} + \sum_{s_{ij}\in S_i} q_{ij}^{Dur} x_{a_{ij}} + CA_{a_i}^{Dur}\right) + wt_{a_{i'}}\right) - \\ \min\limits_{\substack{a\in A_e| \\ \{x_{a_{i'j'}}\in X_e\}}} \left(\sum_{i=1}^{i'-1}\left(wt_{a_i} + \sum_{s_{ij}\in S_i} q_{ij}^{Dur} x_{a_{ij}} + CA_{a_i}^{Dur}\right) + wt_{a_{i'}}\right) \end{array}\right] * s_e \leq 0 \quad \forall\, e \in E_a^{IUR}, \overline{\overline{U}}_e > 0 \tag{7}$$

[11] Whether there has to be considered consumed NFP of an attribute $\propto \in N$ depends on the specific type of attribute or, for instance, on the contractual agreements (e.g., SLA) of user and service provider.

$$\left[\begin{array}{l}\max\limits_{\left\{\substack{a\in A_e|\\ x_{a_{i'j'}}\in X_e}\right\}}\left(\sum_{i=1}^{i'-1}\left(wt_{a_i}+\sum_{s_{ij}\in S_i}q_{ij}^{Dur}x_{a_{ij}}+CA_{a_i}^{Dur}\right)+wt_{a_{i'}}\right)-\\ \min\limits_{\left\{\substack{a\in A_e|\\ x_{a_{i'j'}}\in X_e}\right\}}\left(\sum_{i=1}^{i'-1}\left(wt_{a_i}+\sum_{s_{ij}\in S_i}q_{ij}^{Dur}x_{a_{ij}}+CA_{a_i}^{Dur}\right)+wt_{a_{i'}}+q_{i'j'}^{Dur}x_{a_{i'j'}}\right)\end{array}\right]*(1-s_e)\geq 0 \tag{8}$$

$$\forall\, e\in E_a^{IUR}, \overline{\overline{U}}_e<0$$

In terms of constraints (7), which refer to complementary simultaneous IUR, s_e is 1 if the service compositions of the users $a \in A_e$ all possess the same duration until the point in time right before the potential invocation of the considered service candidates in X_e. Regarding constraints (8) and conflicting simultaneous IUR, s_e could get 0 – to avoid the associated negative utility – if there exists no point in time where the execution of all service candidates $x_{a_{i'j'}} \in X_e$ is overlapping. For the calculation of the duration of a user's service composition until a certain service class $S_{i'}$ both terms (7) and (8) take into account the duration/ response time q_{ij}^{Dur} of the selected services, the waiting time wt_{a_i} and possibly consumed time $CA_{a_i}^{Dur}$. By adjusting the users' decision variables for the individual service candidates and waiting time variables the optimization model enables the temporal coordination of the users' actions in order to achieve the overall optimal solution.

$$\sum_{i=1}^{I}\left(\sum_{s_{ij}\in S_i}q_{ij}^{\propto}x_{a_{ij}}+CA_{a_i}^{\propto}\right)\leq\ Q_a^{\propto}\quad \forall\propto\in N\backslash\{WT\};\ \forall\, a\in A \tag{9}$$

$$\sum_{i=1}^{I}wt_{a_i}\ \leq\ Q_a^{WT}\quad \forall\, a\in A \tag{10}$$

Moreover, the users' global end-to-end constraints regarding the NFP are taken into account by means of term (9) – for all attributes $\propto\in N\backslash\{WT\}$ and under consideration of possibly consumed NFP $CA_{a_i}^{\propto}$ – and term (10) for waiting time WT.

Finally, constraints (11) make certain that exactly one service candidate s_{ij} is selected for each service class S_i and each user a:

$$\sum_{s_{ij}\in S_i}x_{a_{ij}}=1\quad \forall\, i=1,\dots,I;\ \forall\, a\in A \tag{11}$$

However, to ensure correct service re-optimization, our hitherto described basic optimization model has to be adjusted subject to the specific characteristics of the event causing the re-selection – which concerns the following elements:

- The impact of the event itself (failed service, left user, diverging NFP values, etc.)
- Region (A): The already completely executed services and waited times of each user
- Region (B): The current state of each user

We integrate these into our model by modifying existing and adding additional constraints. First, the consideration of the event itself highly depends on the type of the event. For instance, a failure of service s_{ij} for user a can be taken into account by setting the corresponding decision variable $x_{a_{ij}} = 0$. If the failure occurs during execution of the service and thus NFP are consumed, in addition the values $CA_{a_i}^{\propto}$ have to be set accordingly. This is also true in case a user a leaves the process during execution of a service. Even though the user leaves, we do not completely extract her/him from the optimization model as there may still exist dependencies to other users' service compositions, for example, between region (A) of the leaving user and region (C) of other users. Indeed, to model that a user leaves right before or during service class i' and thus is not participating in the process any further, we set constraint (11) and all waiting time variables wt_{a_i} to zero for all service classes $i = i'$ to I. In terms of a service candidate's NFP values differing from the expected values, the model is adjusted by updating the affected $q_{ij}^{\propto}$ regarding that service. Considering region (A), already completely executed services and waited times are integrated by setting the corresponding decision variables $x_{a_{ij}} = 1$ and waiting time variables $wt_{a_i} = V$, with V as the actual waited time. Besides that, the ex-ante expected NFP values $q_{ij}^{\propto}$ can be replaced by the actual realized ones. With regard to region (B), if a user is currently executing a service, the related decision variable $x_{a_{ij}}$ is set to 1. In case a user is currently waiting, s/he could either continue or stop waiting. This is considered by integrating the corresponding waiting time variable $wt_{a_i} \geq V$ with V as the already waited time. For region (C) – that means the unexecuted actions of the process – no further adjustments to the basic optimization model are necessary.

Based on this optimization model tailored to the associated event, the optimal solution for the service re-selection problem in hand could be determined, for instance, by applying mixed integer programming (cf. Nemhauser and Wolsey 1988). Further, if a new optimal service composition is found for one or more users and the remaining part of the process, it may then be deployed and executed.

5 Evaluation

In this section, we evaluate our approach. We first compare our continuous time concept with the existing discrete concept in terms of computation time. By this, we want to analyze whether our concept can overcome the performance issues that would occur when using the existing concept in service re-selection. Second, we want to demonstrate the efficacy of our multi-user service re-selection approach based on a real-world scenario. In this regard, our evaluation design follows the compositional styles simulation- and metric-based benchmarking of artefacts and demonstration (cf. Prat et al. 2015).

To enable the application of mixed integer programming in order to obtain the optimal solution for our optimization model, we first had to linearize our presented non-linear model (using the guidelines as proposed by, for example, Williams 2013). Moreover, we implemented this linearized version in Java, and – to ensure a correct implementation – we further conducted intensive testing of the source code (i.e., manual analysis by other persons than the programmers, unit tests, regression tests, runs with extreme values). For solving the model we use the mathematical programming solver Gurobi[12].

5.1 Performance

As described in Section 4.1, using the existing discrete time concept in multi-user service re-selection would result in performance issues since this requires small time intervals. In the following, we want to analyze these performance issues and whether our novel continuous time concept can overcome them.

For this purpose, we evaluate the computation time needed for solving an exemplary multi-user service selection problem with our approach (this equals re-selection without an event and thus our basic optimization model) and an approach using the discrete time concept, and compare the results. More precisely, for the discrete time approach we consider several settings, each increasing the number of regarded time intervals by reducing the parameter c (i.e., the fixed length of each time interval) while keeping all other parameters unchanged (ceteris paribus). Our representative problem (referred to as scenario *S1* in the following) encompasses 20 service classes á 20 service candidates, 3 NFP (duration, waiting time, costs), and 5 users with 2 IUR each. For each setting we conduct 1,000 simulation runs and determine the average computation time (measured in milliseconds [ms]) Gurobi needs for solving each of both optimization model. For all simulation runs, we use a machine with an Intel Xeon E5-2470 v2 processor with 2.40 GHz, 32 GB RAM, Win7 64bit, Java 1.8, and Gurobi Optimizer 6.5.

Using our continuous time concept the computation time required for solving the problem *S1* is 120 ms, which holds true for all settings as the parameter change only concerns the discrete time approach. As the left diagram of Figure 2 illustrates, the discrete time approach needs much less than 120 ms for settings with a single-digit number of time intervals but – on the other side – more than 1,000 ms for settings with more than 100 time intervals. When conducting this experiment with other problem settings – for example, different number of service classes (scenario *S2*), service candidates (scenario *S3*), users (scenario *S4*) or NFP – we achieve similar results (cf. Figure 2): The continuous approach is superior regarding computation time above a certain small number of time intervals. Therefore, in scenarios which would require a fine granular time concept (resulting in a high number of time intervals) – as it is the case with service re-selection at runtime – the continuous time concept greatly outperforms the discrete time concept in terms of computation time.

[12] http://www.gurobi.com/, accessed August 2016

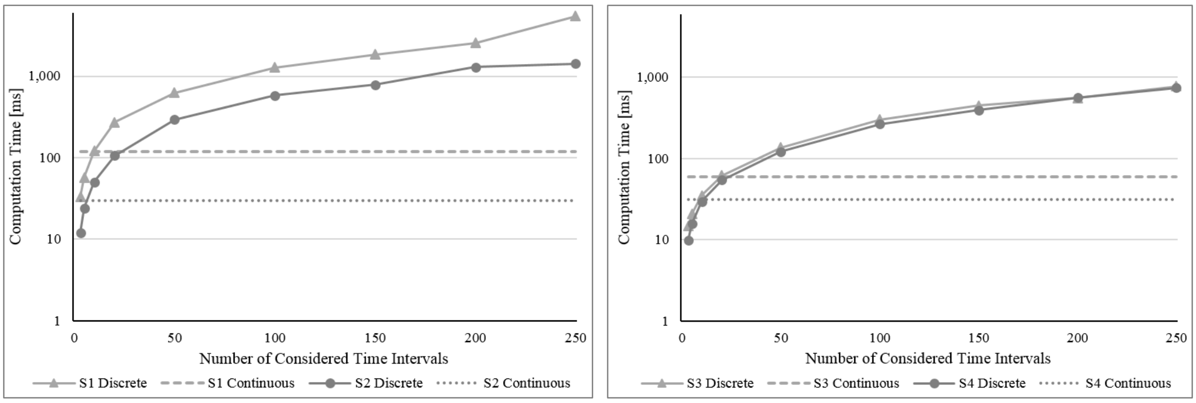

Figure 2. Performance evaluation of continuous vs. discrete time concept

5.2 Efficacy

For the evaluation of efficacy, we build upon a real-world scenario provided by Heinrich et al. (2015a): This scenario refers to a tourism city day trip by five individual persons (cf. Figure 3). In this context, services are understood as service objects (cf. Dannewitz et al. 2008; Hinkelmann et al. 2013) representing real-world entities (e.g., sight, museum, restaurant) that are determined by certain information services (TripAdvisor[13], GooglePlaces[14]). The considered process consists of 15 service classes or actions, respectively, and each action could be realized by a suitable service object – where a service object is characterized by its NFP *costs*, *recommendation value* and *duration*. Using the discrete time concept service objects with no fixed duration (e.g., restaurants, sights) would have to be integrated multiple times, each with a different possible manifestation of duration which have to fit to the considered discrete time intervals. As our approach allows for continuous consideration of time, we are able to leave it to the optimization model to determine the optimal duration of such a service object when solving the problem. More precisely, each of the five users has specified his/her individual target value for duration for each of the 15 actions – in addition to his/her personal weights and requirements regarding all NFP. Based on that, our problem setting encompasses 132 service objects allocated to the 15 actions of the process. Both *duration* and *waiting time* are considered as continuous variables according to our continuous time concept. Furthermore, each user participating in the process has defined three different IUR, which in total results in five mutual complementary, one mutual conflicting, eight simultaneous complementary, and one simultaneous conflicting IUR.

[13] http://www.programmableweb.com/api/tripadvisor, accessed September 2016

[14] http://www.programmableweb.com/api/google-places, accessed September 2016

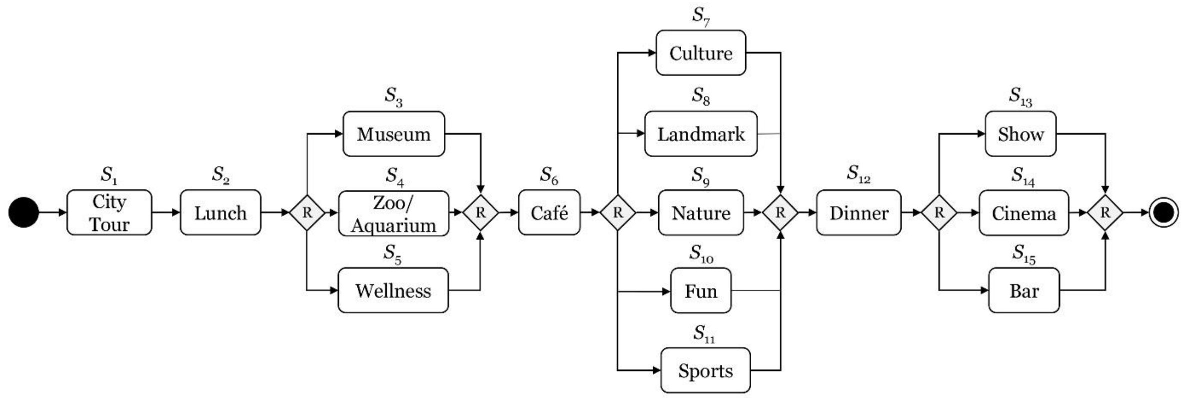

Figure 3. Process of the city day trip (Heinrich et al. 2015a)

During the city day trip there may occur various events that would require a re-selection of the initially planned service compositions. To demonstrate the efficacy of our approach we simulate the occurrence of three exemplary events and analyze the results:

a) Failing service (object): User 4 leaves the restaurant *Sababa* after 9 minutes waiting for a free table to have *2) lunch* elsewhere
b) User leaving the process: User 3 leaves the day trip after having coffee at the *6) café Livingroom*
c) Deviating NFP values: User 5's visit of the restaurant *Ringlers* for having *2) lunch* took 11 minutes longer than initially planned

For this purpose, we use our re-selection approach to obtain the new optimal service compositions of all users and compare them to the initially planned service compositions determined by the existing approach. Due to space restrictions, we focus in the following on discussing the differences in the initial vs. re-optimized service compositions for users 4 and 5 only.

Regarding event a), that means the "failure" of restaurant *Sababa* after 9 minutes, user 4 would not be able to have *2) lunch* without re-selection. After re-selection, the optimal solution for user 4 is having *2) lunch* at the restaurant *Ringlers* for 75 minutes. Since the 9 minutes at *Sababa* also count for the total duration of the day trip, the succeeding stay at *BMW World* (action *3) museum*) is reduced by this amount of time (from 75 min to 66 min) in order to fulfill user 4's constraint regarding the NFP duration. On the other hand, the failure of restaurant *Sababa* and the resulting switch to restaurant *Ringlers* enables the realization of an additional complementary IUR compared to the initially planned solution: "User 5 requests to visit *Pussers Bar* (action *15) bar*) simultaneously together with user 4".

As a consequence of b) with user 3 leaving the process, any IUR after action *6) café* with participation of user 3 could not be fulfilled any more, that means the initially expected positive utilities associated with complementary IUR are not realized but – on the other side – also the expected negative utilities related to conflicting IUR are avoided. For instance, this refers to the IUR "user 5 requests to visit *English Garden* (action *9) nature*) simultaneously with user 3".

Without re-selection, the occurrence of event c) (i.e., user 5's stay at *Ringlers* for *2) lunch* is increased by 11 minutes) would lead to a violation of user 5's global end-to-end constraint regarding NFP duration. Additionally, simultaneous complementary IUR planned further in the process would probably be not fulfilled, and simultaneous conflicting IUR expected to be avoided could potentially be realized. Indeed, in the re-optimized service composition user 5's visit of *3) museum BMW World* is abbreviated by 11 minutes to solve these issues. For example, as a result user 5's complementary IUR of visiting *English Garden* (action *9) nature*) simultaneously with user 3 can still be realized.

6 Conclusion, Limitations, and Further Research

Within this work, we presented a multi-user service re-selection approach enabling successful process recovery after occurrence of certain events at process runtime by determining the optimal service compositions for the remaining users and the remaining part of the process (contribution ❶).

For this, we proposed an optimization model taking into account multiple users with their preferences and requirements and dependencies resulting from mutual as well as simultaneous IUR. For the consideration of the latter, which requires temporal coordination of the users, we introduced a continuous time concept (contribution ❷) to overcome the flexibility and performance issues related to the existing discrete time concept – which is supported by the results of our performance evaluation in Section 5.1. We therefore contribute to the current body of knowledge in multi-user service (re-)selection. Furthermore, using the continuous time concept obviates the need for the definition of specific, discrete time intervals by the decision-maker when building the optimization model for a certain re-selection problem. Besides that, our findings also reveal important practical benefit. Depending on the problem size (i.e., number of service classes, service candidates and users), the number of possible service compositions can get extremely large. Additionally, with dependencies existing between different users and the required temporal coordination of the users' actions due to IUR this leads to a great complexity of the (initial) service selection problem. Thus, in case a process requires re-optimization due to an occurred event, the corresponding re-selection problem may still be of high complexity subject to the size of the remaining part of the process. The approach proposed in this work helps to deal with this complexity since it determines the optimal service composition for the rest of the process and each remaining user.

However, our approach is also subject to some limitations that need to be addressed in future research: As in this work the focus primarily lies upon the development of the model, we so far neglected the time-to-repair related to conducting re-selection at runtime (i.e., time needed from occurrence of an event until successful process continuation) (cf. Canfora et al. 2008; Sandionigi et al. 2013) – and by this also the duration overhead produced by the re-selection algorithm itself. From a practical point of view, the time-to-repair also needs to be taken into consideration in service re-selection as it adds up to the overall duration/ response time and also influences the realization of simultaneous IUR in the remaining part of the process. Furthermore, when aiming at an optimal solution as we do, we have to recognize that the service (re-)selection problem is NP-hard which generally corresponds to an exponential development in computation time (Abu-Khzam et al. 2015). In terms of service re-selection, this requires a cost-benefit tradeoff between improving the utility of a user's service composition and the duration overhead produced by the re-selection algorithm (Canfora et al. 2008). Thus, to further improve our approach we will in a next step focus on integrating the time-to-repair in our model and, additionally, on reducing the computation time needed for solving the multi-user service re-selection problem. A promising starting point for that could be the development of a heuristic technique (cf., e.g., Alrifai et al. 2012; Canfora et al. 2008; Qiqing et al. 2009) by means of an algorithm that efficiently scales with the problem size while achieving high decision quality, that means close-to-optimal solutions.

In conclusion, this work can serve as a first step for a comprehensive service re-selection approach regarding multi-user processes where dependencies among users exist, for example, resulting from IUR.

References

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., and Sikora, F. 2015. "On the Complexity of QoS-Aware Service Selection Problem," in *Service-oriented computing*, A. Barros (ed.), Heidelberg: Springer, pp. 345–352.

Ai, L., and Tang, M. 2008. "QoS-Based Web Service Composition Accommodating Inter-service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm," in *IEEE Fourth International Conference on eScience*, Indianapolis, IN, USA, pp. 119–126.

Alrifai, M., and Risse, T. 2009. "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proceedings of the 18th International Conference on World Wide Web*, J. Quemada, G. León, Y. Maarek and W. Nejdl (eds.), Madrid, Spain, pp. 881–890.

Alrifai, M., Risse, T., and Nejdl, W. 2012. "A hybrid approach for efficient Web service composition with end-to-end QoS constraints," *ACM Transactions on the Web* (6:2), pp. 1–31.

Ardagna, D., and Mirandola, R. 2010. "Per-flow optimal service selection for Web services based processes," *Journal of Systems and Software* (83:8), pp. 1512–1523.

Ardagna, D., and Pernici, B. 2007. "Adaptive Service Composition in Flexible Processes," *IEEE Transactions on Software Engineering* (33:6), pp. 369–384.

Barry, D. K. 2012. *Web Services, Service-Oriented Architectures, and Cloud Computing*: *The Savvy Manager's Guide*, Burlington: Elsevier Science.

Benouaret, K., Benslimane, D., and Hadjali, A. 2012. "Selecting Skyline Web Services for Multiple Users Preferences," in *IEEE 19th International Conference on Web Services (ICWS)*, Honolulu, HI, USA, pp. 635–636.

Berbner, R., Spahn, M., Repp, N., Heckmann, O., and Steinmetz, R. 2007. "Dynamic Replanning of Web Service Workflows," in *Inaugural IEEE-IES Digital EcoSystems and Technologies Conference*, Cairns, Australia, pp. 211–216.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2008. "A framework for QoS-aware binding and re-binding of composite web services," *Journal of Systems and Software* (81:10), pp. 1754–1769.

Dannewitz, C., Pentikousis, K., Rembarz, R., Renault, É., Strandberg, O., and Ubillos, J. 2008. "Scenarios and research issues for a network of information," in *4th International Mobile Multimedia Communications Conference*.

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

Heinrich, B., Klier, M., Lewerenz, L., and Mayer, M. 2015a. "Enhancing Decision Support in Multi User Service Selection," in *36th International Conference on Information Systems (ICIS),* Fort Worth, USA.

Heinrich, B., Klier, M., Lewerenz, L., and Zimmermann, S. 2015b. "Quality-of-Service-Aware Service Selection: A Novel Approach Considering Potential Service Failures and Nondeterministic Service Values," *Service Science* (7:1), pp. 48–69.

Hinkelmann, K., Maise, M., and Thonssen, B. 2013. "Connecting enterprise architecture and information objects using an enterprise ontology," in *Enterprise Systems Conference (ES),* Cape Town, South Africa, pp. 1–11.

Jin, H., Zou, H., Yang, F., Lin, R., and Shuai, T. 2012. "A Novel Method for Optimizing Multi-User Service Selection," *Journal of Convergence Information Technology* (7:5), pp. 296–310.

Kang, G., Liu, J., Tang, M., Liu, X., and Fletcher, K. K. 2011. "Web Service Selection for Resolving Conflicting Service Requests," in *IEEE International Conference on Web Services,* Washington, DC, pp. 387–394.

Klusch, M., and Kapahnke, P. 2012. "The iSeM matchmaker: A flexible approach for adaptive hybrid semantic service selection," *Web Semantics: Science, Services and Agents on the World Wide Web* (15), pp. 1–14.

Lewis, J., and Fowler, M. 2014. *Microservices*. http://martinfowler.com/articles/microservices.html. Accessed 14.01.16.

Li, J., Ma, D., Li, L., and Zhu, H. 2008. "AADSS: Agent-Based Adaptive Dynamic Semantic Web Service Selection," in *4th International Conference on Next Generation Web Services Practices,* Seoul, Korea. 20.10.08 - 22.10.08, IEEE, pp. 83–89.

Li, J., Ma, D., Mei, X., Sun, H., and Zheng, Z. 2011. "Adaptive QoS-Aware Service Process Reconfiguration," in *IEEE International Conference on Services Computing (SCC),* Washington, DC, USA, pp. 282–289.

Lin, K.-J., Zhang, J., Zhai, Y., and 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), pp. 157–168.

Nemhauser, G. L., and Wolsey, L. A. 1988. *Integer and combinatorial optimization*, New York: Wiley.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. 2007. "Service-Oriented Computing: State of the Art and Research Challenges," *Computer* (40:11), pp. 38–45.

Prat, N., Comyn-Wattiau, I., and Akoka, J. 2015. "A Taxonomy of Evaluation Methods for Information Systems Artifacts," *Journal of Management Information Systems* (32:3), pp. 229–267.

Qiqing, F., Xiaoming, P., Qinghua, L., and Yahui, H. 2009. "A Global QoS Optimizing Web Services Selection Algorithm Based on MOACO for Dynamic Web Service Composition," in *International Forum on Information Technology and Applications (IFITA),* Chengdu, China, pp. 37–42.

Rodriguez-Mier, P., Mucientes, M., and Lama, M. 2012. "A Dynamic QoS-Aware Semantic Web Service Composition Algorithm," in *Service-oriented computing*, C. Liu, H. Ludwig, F. Toumani and Q. Yu (eds.), Berlin: Springer, pp. 623–630.

Sandionigi, C., Ardagna, D., Cugola, G., and Ghezzi, C. 2013. "Optimizing Service Selection and Allocation in Situational Computing Applications," *IEEE Transactions on Services Computing* (6:3), pp. 414–428.

Shen, Y., Wang, M., Tang, X., Luo, Y., and Guo, M. 2012a. "Context-Aware HCI Service Selection," *Mobile Information Systems* (8:3), pp. 231–254.

Shen, Y., Yang, X., Wang, Y., and Ye, Z. 2012b. "Optimizing QoS-Aware Services Composition for Concurrent Processes in Dynamic Resource-Constrained Environments," in *IEEE 19th International Conference on Web Services (ICWS),* Honolulu, HI, USA, pp. 250–258.

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. 2014. "Web services composition: A decade's overview," *Information Sciences* (280), pp. 218–238.

Stein, S., Payne, T. R., and Jennings, N. R. 2009. "Flexible provisioning of web service workflows," *ACM Transactions on Internet Technology* (9:1), pp. 1–45.

Strunk, A. 2010. "QoS-Aware Service Composition: A Survey," in *IEEE 8th European Conference on Web Services (ECOWS), 2010*, A. Brogi (ed.), Ayia Napa, Cyprus. 1/12/2010 - 3/12/2010, Piscataway, NJ: IEEE, pp. 67–74.

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

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., and Li, P. 2010. "Web Service Selection for Multiple Agents with Incomplete Preferences," in *IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology,* Toronto, AB, Canada, pp. 565–572.

Weinhardt, C., Blau, B., Conte, T., Filipova-Neumann, L., Meinl, T., and Michalk, W. 2011. *Business Aspects of Web Services*, Berlin, Heidelberg: Springer.

Williams, H. P. 2013. *Model building in mathematical programming,* Chichester: Wiley.

Yu, T., and Lin, K.-J. 2005. "Adaptive algorithms for finding replacement services in autonomic distributed business processes," in *Proceedings of Autonomous Decentralized Systems,* Chengdu, Jiuzhaigou, China. April 4-8, 2005, pp. 427–434.

Yu, T., Zhang, Y., and Lin, K.-J. 2007. "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web* (1:1).

Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., and Chang, H. 2004. "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering* (30:5), pp. 311–327.

Zheng, Z., and Lyu, M. R. 2010. "An adaptive QoS-aware fault tolerance strategy for web services," *Empirical Software Engineering* (15:4), pp. 323–345.

Zheng, Z., Zhang, Y., and Lyu, M. R. 2014. "Investigating QoS of Real-World Web Services," *IEEE Transactions on Services Computing* (7:1), pp. 32–39.

4 Paper 3: Service Selection in Mobile Environments: Considering Multiple Users and Context-Awareness

Status	Published	Full Citation
Accepted	09/2018	Heinrich, B., and Mayer, M. 2018. “Service selection in mobile environments: considering multiple users and context-awareness,” *Journal of Decision Systems* (27:2), pp. 92–122.
Annotation: The paper is written in British English		

Post publication changes:

- The format of the keywords was changed for consistency reasons
- In the whole paper, the format of the references was changed for consistency reasons
- In the whole paper, the format of the headings was changed for consistency reasons
- In the whole paper, the comma setting for “e.g.” and “i.e.” was adjusted for consistency reasons
- “single user” changed to “single-user” in Section 6.1
- “$\overline{\overline{U}}_e^a$” changed to “$\overline{\overline{U}}_e^\propto$” in Appendix B term (1)

Abstract

In mobile environments, users often need to coordinate their actions with other users with regard to user-individual context information like current location when selecting suitable services for a process. Thereby, some users may prefer to conduct particular services together with certain other users. Such multi-user context-aware service selections could result in complex decision problems – making decision support for the participating users highly valuable or even necessary. To do so, we propose an optimisation-based service selection approach for multi-user context-aware processes. We also show how our approach provides decision support by evaluating its efficacy based on a real-world scenario.

Keywords: Service selection, Multi-user processes, Context information, Mobile environment

1 Introduction

The tremendous advances in mobile technologies and the rise of mobile business over the last decade have led to a rapid growth of the service market (Statista 2017). Selecting services for processes in mobile environments like a tourism city day trip often results in a decision problem of high complexity as it is often necessary to coordinate the actions of multiple users as well as to consider context information. In this regard, context information can refer to the current location, daytime, and so on, or generally speaking ‘any information that can be used to characterize the situation of an entity[15]' (Dey 2001, p. 5). Such multi-user context-aware processes in mobile environments can be found, for example, in roadside, healthcare or disaster relief assistance, the areas of everyday efficiency and planning (price comparison, routing, schedule management on mobile devices), or in the tourism domain (cf. Gavalas et al. 2014; Neville et al. 2016; Ventola 2014; Zhang et al. 2009).

Considering, for instance, healthcare assistance in hospitals, healthcare professionals need to be assigned to patients in a suitable way to adequately support their therapy, where some patients need multiple treatments (i.e. services) in a defined order (i.e. process) (cf. Marynissen and Demeulemeester 2016). Here, healthcare professionals currently start to use mobile devices in combination with hospital information systems to retrieve information about patients such as medical data and previous diseases but also about treatment rooms and operating theatres in terms of context information like location and time schedule (cf. Boruff and Storie 2014; Ventola 2014). This information can then be used for assigning healthcare professionals with certain skills to patients and near-located, available treatment rooms / operating theatres to minimise the overall duration (including waiting time) for the patients, for instance. Consequently, healthcare professionals need to conduct certain actions to treat patients in the best way. For some of these actions, it is more beneficial when they are conducted together by several healthcare professionals with different skills (e.g. surgery) – requiring the coordination of the healthcare professionals. This can be characterised as a multi-user context-aware service selection problem focusing on the support of patients' medical therapy where the respective selection (i.e. assignment) is highly complex (cf. Marynissen and Demeulemeester 2016).

Another application field for multi-user context-aware processes in a mobile environment is the tourism domain, for instance, a city day trip conducted together by a group of users. Here, the users can retrieve information about real-world entities like sights, restaurants or museums by using mobile information applications (e.g. Yelp, TripAdvisor) – where each entity with its properties (e.g. price, duration, location, business hours) can be understood as a service object (cf. Lewerenz 2015; Yu and Reiff-Marganiec 2009). Such a city day trip usually encompasses many different actions like visiting a museum, having lunch and visiting a sight. Each of these actions could then be realised by different real-world entities represented by service objects, for example, 'Pinakothek of Modern Art' or 'Bavarian National Museum' (referring to the city of Munich, Germany). Selecting suitable service objects for such a process (i.e. trip) requires to deal with the preferences (e.g. price more important than duration) and requirements (e.g. overall budget) of each individual user as well as with the context information of both the users (location, daytime, etc.) and the real-world entities (location, business hours, etc.). Moreover, with several users conducting a city day trip together finding the optimal composition of service objects for each user additionally requires a coordination of the users' actions in their processes. Thereby, when dealing with multiple users in service selection, we need to cope with (inter-)user preferences, which we denote as Inter-User-Requests (IUR). An example for an IUR here would be a user favouring to visit the 'Bavarian National Museum' together with two other

[15] 'An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves' (Dey (2001), p. 5)

particular users participating in the trip. Thus, in addition to context information, we also consider such IUR in this work that means user-defined requests referring to other users.

Against this background, users trying to determine their optimal composition of services resp. service objects to conduct a multi-user context-aware process are usually confronted with an information overload problem (cf. Shen et al. 2012; Zhang et al. 2009) since there often exist many alternative service objects for realising each action of such a process (referring to the example above, TripAdvisor lists over 3,000 different restaurants for having lunch in Munich[16]). More precisely, when taking into account multiple users and context information, a service selection problem of high complexity results since it requires to consider dependencies that exist within a user's service composition as well as among different users' service compositions. These dependencies are illustrated in more detail in the next section. As a consequence, a suitable approach is needed to support the users in terms of selecting the optimal service composition for each user. To the best of our knowledge, none of the existing optimisation-based service selection approaches aims at integrating multiple users and context-awareness (cf. section 3.1 Related Literature). This leads us to the following research question for our paper:

> *How to develop an optimisation-based service selection approach which considers dependencies resulting from both multiple users and context information?*

In the following section, we present a motivating scenario for our research which is followed by the background in terms of a discussion of related literature, the resulting research gap, our contribution, and the introduction of our model setup. In the fourth section, we analyse and model both multiple users and context information. Based on that, we propose our approach in terms of an optimisation model (cf. Section 5), which we then evaluate regarding efficacy and performance. Finally, we conclude our paper with a discussion on implications (Section 7), important limitations and an outlook on further research (Section 8).

2 Motivating Scenario

Our scenario refers to a tourism day trip to the City of Munich, Germany, by three individual users where the users plan to conduct several different actions such as visiting a museum, having lunch or visiting a café (cf. Figure 1 for an example). Obviously, there exist numerous alternatives for conducting each action (e.g. Restaurant 'Vinaiolo', Restaurant 'L'Ancora', etc.). Subject to the individual price, duration and location of these alternatives, some of them are more valuable for a user than others based on her/his own individual target weights (e.g. price may be more important than duration) and requirements (e.g. overall budget). Furthermore, in such a scenario, it is likely that some users also have requests that refer to other users (i.e. IUR), for example, 'user 3 requests to take a coffee together with user 2 regardless which café' or 'user 1 requests not to go all together to the "German Theatre Munich"'. Taking the first IUR, user 3 associates a positive value for being at the same café at the same time as

---

[16] https://www.tripadvisor.com/Tourism-g187309-Munich_Upper_Bavaria_Bavaria-Vacations.html, accessed July 2018

user 2. Moreover, as some museums or sights offer group discounts, it could be more beneficial for the three users to visit the same museum or sight.

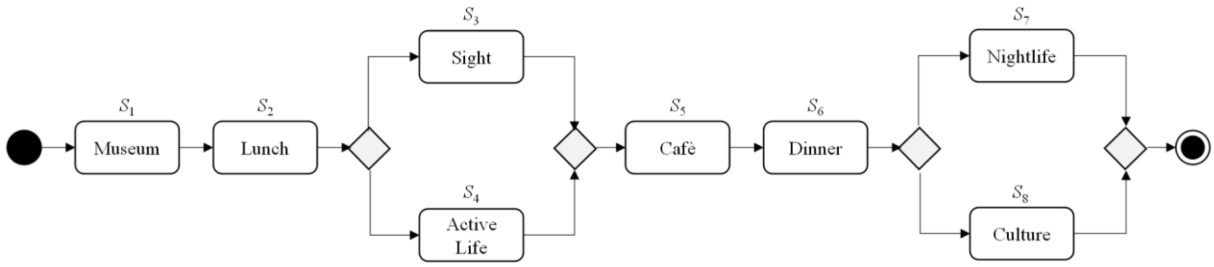

Figure 1. Process model for city day trip

Obviously, due to the high number of available real-world entities, individual target weights and requirements as well as IUR, decision support is valuable to determine the best entities for the complete day trip regarding all users. Therefore, we represent each entity (e.g. Restaurant 'Vinaiolo') and its information such as price, duration, business hours and location as service object with non-functional properties (NFP). Based on this, service selection can be used to determine the optimal set of service objects (i.e. service composition) for each user and the entire process.

Regarding the above-mentioned IUR 'user 3 requests to take a coffee together with user 2 regardless which café', the realisation of the positive value associated by user 3 supposes that the selected service compositions of both users encompass the same café (represented by the same service object). However, if the service objects are selected independently for both users, this would obviously only happen by chance. Therefore, realising IUR requires to take such *preference-based dependencies* between different users' service compositions into account when selecting suitable service objects (cf. Heinrich et al. 2015). Additionally, both users must arrive at the café at the same time, which is dependent on the individual starting time of the day trip of both users (i.e. initial context of each user) and the duration of the previously conducted actions which is most likely different for each of them. In this respect, it may also be beneficial for one user to wait a certain amount of time to be able to visit the same café at the same time as the other user and thus realising the IUR. To consider such *temporal-based dependencies*, a temporal coordination of the users' actions including possible waiting times is necessary, too. The same applies to *context-based dependencies* that result from context information such as group discounts or the distance to cover between, for example, the café visited by both users and the preceding actions each user has conducted.

To sum up, in order to provide feasible and suitable decision support in such a multi-user context-aware scenario, all these different types of *preference-based, context-based* and *temporal-based dependencies* must be taken into account when selecting the optimal service compositions for all users.

3 Background

Next, we review existing works and based on that discuss both our research gap and contribution. This will be followed by the introduction of our model setup.

3.1 Related Literature

We structure the existing literature dealing with multiple users and context-awareness in optimisation-based service selection according to the types of *preference-based, context-based* and *temporal-based dependencies* introduced above.

First, we analyse *preference-based dependencies* resulting from multiple users. In this respect, existing multi-user service selection approaches deal with restrictions that prescribe or limit the usage of services (or service objects) by two or more users. Those so-called hard restrictions must be satisfied in a feasible service composition. For example, Benouaret et al. (2012), Wanchun et al. (2011) and Wang et al. (2010) examine a situation, where the mutual usage of a certain service by several users is mandatory, while He et al. (2012), Kang et al. (2011) and Wang et al. (2014) address capacity limits of services. However, preference-based dependencies and thus users preferring (but not enforcing) to use certain services (or service objects) together with other users have not been addressed in literature so far.

Second, when considering *context-based dependencies*, there are many works that deal with context information and context-awareness in terms of selecting (single) services for a single user (e.g. Ai and Tang 2008; Deng et al. 2016; Sandionigi et al. 2013; Vanrompay et al. 2009; Yu and Reiff-Marganiec 2009; Zhou et al. 2008). Few of them also consider context-based dependencies that could exist within a certain part or the entire service composition of a user (e.g. Deng et al. 2016; Shen et al. 2012; Xu and Jennings 2010; Yu and Reiff-Marganiec 2009; Zhou et al. 2008). However, those approaches focus solely on a single user and thus on context-based dependencies within a single user’s service composition. But as we consider multi-user processes, we must account for the fact that there could also exist context information referring to multiple users.

Third, when addressing both (time-dependent) preferences/IUR and (time-dependent) context information, we additionally need to deal with *temporal-based dependencies*. Optimisation-based service selection approaches coping with such temporal-based dependencies can be found in (Guidara et al. 2014; Heinrich and Lewerenz 2015; Xu and Jennings 2010). Although they define a time concept, none of them addresses a temporal coordination of the users’ actions including possible waiting times. In this regard, the consideration of waiting times is necessary for comprehensive decision support as, for instance, this allows one or many users to wait instead of moving to a less favoured service (or service object).

3.2 Identified Research Gap and Contribution

In summary, important contributions have been made with respect to multiple users and context-awareness in service selection. However, an optimisation-based service selection approach that copes with *preference-based, context-based* and *temporal-based dependencies* is

– to the best of our knowledge – missing so far. Thus, we will address this gap in our work in terms of proposing a novel service selection approach.

Existing optimisation-based approaches, which solve the general service selection problem (i.e. without considering multiple users and context information), search for the optimal service composition for one single user under consideration of target weights and requirements regarding the NFP like price, availability, and so on (e.g. Alrifai et al. 2012; Ardagna and Pernici 2007; Yu et al. 2007; Zeng et al. 2004). At this, the service selection problem is usually formulated as knapsack optimisation problem (e.g. Alrifai et al. 2012; Alrifai and Risse 2009; Yu et al. 2007). However, when considering multiple users and context information, we have to deal with the question how to model and integrate the resulting preference-based, context-based and temporal-based dependencies in terms of an optimisation-based approach. Here the literature provides two fundamental alternatives: a stateless versus stateful representation of dependencies. In terms of a stateless representation, dependencies are integrated directly into an optimisation model. For instance, He et al. (2012), Jin et al. (2012) and Kang et al. (2011) consider multiple users and capacity limits by extending the optimisation model in terms of additional constraints. However, they only focus on hard restrictions. Regarding a stateful representation, first approaches (e.g. Lewerenz 2015) utilise the concept of world and belief states (cf. Ghallab et al. 2004) to organise and model context information. Thus, existing context-based dependencies are specified by state-service combinations that are determined before the optimisation takes place. However, they do not consider preference-based and temporal-based dependencies in their approaches.

We aim to provide both a stateless and stateful optimisation model, each incorporating dependencies resulting from multiple users and context information. This allows us to evaluate both alternatives and their advantages resp. disadvantages in detail. In conclusion, this leads us to the following three-fold contribution of our paper:

❶ Consideration of *preference-based* and *context-based dependencies* resulting from multiple users and context information

❷ Consideration of *temporal-based dependencies* resulting from time-dependent preferences/IUR and time-dependent context information which requires a time concept dealing especially with waiting times

❸ Development of *optimisation models* for a multi-user context-aware service selection based on a stateful resp. stateless representation of dependencies

3.3 Model Setup

In this section, we introduce our model setup, referring to those definitions and modelling elements in line with existing works that can serve as a common knowledge base. This allows for a better differentiation between existing knowledge and our contribution ❶-❸ in the Sections 4 and 5.

We consider a sequential process that consists of a number of actions or service classes S_i (with i = 1 to I). Each service class encompasses a set of functional equivalent services s_{ij} (with j =

1 to J_i) – which are referred to as service objects – that differ only in their NFP. Furthermore, a service composition is defined as a concrete implementation of a process in terms of a set of service objects with exactly one service object out of each service class of the process. Appendix A provides an overview of the used formal notation throughout this work.

When considering service selection without dealing with context information, a service object s_{ij} would be described only by the set M of non-context-aware (NCA) attributes like price or duration. Based on that, the vector $q_{ij} = \left[q_{ij}^1, \dots, q_{ij}^M\right]^T$ contains the quantified NFP values of a service object s_{ij} regarding all NCA attributes M. For the selection of service objects with several NFP, a utility function U is often used – where the purpose of U is to map the values of the different attributes onto a single utility value. In our work, we apply – in line with, for instance, Alrifai et al. (2012), Jin et al. (2012) and Guidara et al. (2014) – the utility function described in detail by Alrifai and Risse (2009). But without limitations, other utility functions could be used as well with our approach as the exact way the utility of a certain service object is calculated has no impact on the formulation of our optimisation models in Section 5. To determine the utility value of a service object, this utility function uses the simple additive weighting (SAW) technique consisting of normalisation and weighting of the NFP. For the normalisation step (i.e. to enable comparability between different NFP), the utility function utilises the aggregated minimum and maximum values of the attributes over all service classes S_i. Further, the attributes $\propto \in M$ can be divided into the subset of attributes M^- that need to be minimised and the subset of attributes M^+ that need to be maximised. The aggregated values $P_{min}^{\propto}$ and $P_{max}^{\propto}$ for each attribute $\propto$ in M^- and M^+ can be calculated as follows:

$$P_{min}^{\propto} = \sum_{i=1}^{I} \left(P_{i,min}^{\propto}\right) \text{ with } P_{i,min}^{\propto} = \min_{s_{ij} \in S_i} q_{ij}^{\propto} \qquad (1)^{17}$$

$$P_{max}^{\propto} = \sum_{i=1}^{I} \left(P_{i,max}^{\propto}\right) \text{ with } P_{i,max}^{\propto} = \max_{s_{ij} \in S_i} q_{ij}^{\propto}$$

These aggregated minima and maxima could then be used to normalise the NFP values. To achieve a single utility value U_{ij} (cf. Equation (2)) for a service object s_{ij}, the weighted sum over all attributes based on user-defined target weights $w^{\propto}$ regarding the attributes $\propto \in M$ is determined. Here, it must hold that $\sum_{\propto=1}^{M} w^{\propto} = 1$. Considering multi-user service selection and therefore multiple users $a \in A$ leads to possibly varying utility values $U_{a_{ij}}$ of a particular service object s_{ij} for different users a since each user is likely to have its own target weights $w_a^{\propto}$ (cf. Alrifai et al. 2012; Jin et al. 2012):

$$U_{a_{ij}} = \sum_{\propto \in M^-} \left(\frac{P_{i,max}^{\propto} - q_{ij}^{\propto}}{P_{max}^{\propto} - P_{min}^{\propto}}\right) * w_a^{\propto} + \sum_{\propto \in M^+} \left(\frac{q_{ij}^{\propto} - P_{i,min}^{\propto}}{P_{max}^{\propto} - P_{min}^{\propto}}\right) * w_a^{\propto} \qquad (2)$$

[17] The presented function refers only to the summation aggregation type (e.g. costs, duration). For other aggregation functions, please see, e.g. Alrifai et al. (2012).

Based on this, the overall utility value of a service composition can be calculated by summing up the individual utilities of all selected service objects. Besides the target weights $w_a^{\propto}$, user-defined requirements in terms of global end-to-end constraints $Q_a^{\propto}$ regarding the NFP must be considered as well (e.g. Jin et al. 2012; Yu et al. 2007).

Now, when additionally considering context information in service selection, we distinguish whether this context information is of static or dynamic nature. In contrast to the static nature (i.e. the context information is exogenously given regarding the service composition, like weather), we speak of the dynamic nature of context information when the set of selected service objects influences the actual manifestation of the context information (cf. Damián-Reyes et al. 2011; Vanrompay et al. 2009). Examples for such context information are daytime-dependent availability of service objects (i.e. business hours), price discount on a certain set of service objects, and the distance between different service providers or devices (Shen et al. 2012; Yu and Reiff-Marganiec 2009; Zheng et al. 2014; Zhou et al. 2008). Addressing this dynamic nature of context information leads to context-based dependencies between several or all service objects (Heinrich and Lewerenz 2015; Zhou et al. 2008).

In service selection, context information can be taken into account by means of context-aware (CA) attributes (cf. Ai and Tang 2008; Xu and Jennings 2010; Yu and Reiff-Marganiec 2009; Zhou et al. 2008) that together with the NCA attributes describe the NFP of a service object. Moreover, the subset of NCA attributes M and the subset of CA attributes O form together the set of attributes N (with $M \cup O = N$ and $M \cap O = \emptyset$) that are considered in a certain service selection problem. Furthermore, each user has her/his individual target weights $w_a^{\propto}$ and global end-to-end constraints $Q_a^{\propto}$ regarding all CA attributes O. But in contrast to NCA attributes, CA attributes are subject to the following three fundamental effects as a result of the existing *context-based dependencies* between different service objects (cf. Lewerenz 2015):

(1) The determination of context information is dependent on the service objects selected for a specific service composition. Thus, the *quantified values* of a CA attribute could be different for the same considered service object used in different service compositions.
(2) As a direct consequence of (1), the utility of a service object or a set of service objects is affected by context information, which means the corresponding *utility value* is different for each service composition (thus influencing the selection of the *optimal composition*).
(3) Furthermore, the selection of a service object could also have an effect on the *feasibility* of other service objects.

Consequently, all three fundamental effects need to be taken into account when modelling dependencies in the following.

4 Modelling Preference-based, Context-based and Temporal-based Dependencies

Based on the model setup, we will analyse and model *preference-based*, *context-based* and *temporal-based dependencies* as part of our contribution (cf. ❶-❷).

4.1 Modelling Preference-based and Context-based Dependencies

An IUR is understood as a user-defined request referring to other users. Thus, when specifying an IUR, both the set of participating users and the service object or service class related with that IUR need to be determined. Further, a particular positive versus negative value is associated with the realisation of that IUR. We distinguish four basic types of IUR, regarding the two dimensions *relation* and *time* (cf. Table 1).

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">Relation</th></tr>
<tr><th>Complementary</th><th>Conflicting</th></tr>
<tr><td rowspan="2">Time</td><td>Mutual
(time-independent)</td><td>Complementary mutual usage
• A user requests to use the same service object(s)/ service class(es) together with one or more other users.
• A positive value is associated with this IUR.</td><td>Conflicting mutual usage
• A user requests not to use the same service object(s)/service class(es) together with one or more other users.
• A negative value is associated with this IUR.</td></tr>
<tr><td>Simul-taneous
(time-dependent)</td><td>Complementary simultaneous usage
• A user requests to use and thus to start the same service object(s)/service class(es) together with one or more other users at the same time.
• A positive value is associated with this IUR.</td><td>Conflicting simultaneous usage
• A user requests not to use the same service object(s)/service class(es) together with one or more other users at any moment in time.
• A negative value is associated with this IUR.</td></tr>
</table>

Table 1. Categorisation of IUR subject to the dimensions 'relation' and 'time'

Initially, an IUR refers to a certain single service object or a certain service class. Since an IUR concerns more than one user, preference-based dependencies exist among different users' service compositions, which need to be taken into consideration when determining their utility. Further, simultaneous IUR additionally lead to dependencies of temporal nature, which are considered in Section 4.2 in detail.

When addressing context information in multi-user processes, we must account for the fact that CA attributes exist which refer to more than one user. A common example would be group discounts that will only be attained if a certain number of users will select the corresponding service object. Apart from that, CA attributes can also be time dependent like business hours. Accordingly, in Table 2 we distinguish four types of CA attributes, where each type represents a different kind of context-based dependency. Existing approaches merely address the single user-column of the table, which means they consider context-based dependencies and partially temporal-based dependencies for CA attributes referring to the service composition of a single user.

		CA Attributes with Relation to	
		Single User	**Multi User**
Time	**Time-independent**	CA Attributes resulting in • dependencies **within** one user's **service composition** • *e.g. distance, time-independent discount on service object A + B, favourite scores*[18]*, etc.*	CA Attributes resulting in • dependencies **among** different users' **service compositions** • *e.g. time-independent group discount, etc.*
	Time-dependent	CA Attributes resulting in • dependencies **within** one user's **service composition** • **temporal-based** dependencies • *e.g. availability/price of services objects dependent on daytime*	CA Attributes resulting in • dependencies **among** different users' **service compositions** • **temporal-based** dependencies • *e.g. time-dependent group discount, etc.*

Table 2. Categorisation of CA attributes and dependencies subject to the dimensions 'number of users' and 'time'

After systematising preference-based and context-based dependencies, we now model them formally. We first focus on preference-based dependencies resulting from IUR: In traditional single-user service selection, a user usually specifies her/his target weights and requirements regarding the NFP (cf. e.g. Alrifai et al. 2012; Yu et al. 2007; Zeng et al. 2004). When taking IUR into account, each user $a \in A$ additionally has the possibility to specify a set of different IUR E_a^{IUR}. In doing so, a user a defines for each IUR $e \in E_a^{IUR}$ the set of participating users A_e^{IUR}, for each participating user the associated service object/service class (which results in the set X_e^{IUR}), and whether that IUR is of the *mutual* (time-independent) or *simultaneous* (time-dependent) type. Furthermore, the user sets a particular request value q_e^{IUR} which is positive in the *complementary* case and negative in the *conflicting* case. This value corresponds to how important the user assesses the realisation of that IUR compared to other IUR she/he specified. To represent the importance of IUR, the user may also specify a target weight w_a^{IUR}. In that way, we consider IUR as regular attribute $IUR \in N$, more precisely as element of the subset of CA attributes O. As a consequence, for each IUR a utility value can be obtained through normalising and weighting the request value q_e^{IUR} by means of the same utility function applied on the NCA and CA attributes of the selection problem as described in Section 3.3. Here, we differentiate the utility values $\hat{U}_e^{IUR}$ for mutual (time-independent) IUR and $\bar{\bar{U}}_e^{IUR}$ for simultaneous (time-dependent) IUR where the utility values can be positive or negative subject to the inherent case (complementary or conflicting).

Second, context-based dependencies resulting from CA attributes could be modelled in a similar way. In detail, we break down the dependencies caused by a CA attribute $\propto \in O$ for each user $a \in A$ into a set of single dependencies $E_a^{\propto}$. Furthermore, each dependency $e \in E_a^{\propto}$ encompasses a set of service objects $X_e^{\propto}$ which belong together in terms of utility or feasibility

[18] Favourite scores represent user favourites with respect to a certain category (e.g. type of restaurant) of an attribute.

determination, for instance, the set of service objects which need to be selected to realise a certain group discount. In case the dependency e refers to utility determination regarding the set of service objects $X_e^{\propto}$, the corresponding utility associated with the CA attribute is obtained based on the quantified value $q_e^{\propto}$ of the related context information by applying the utility function. Here, we also differentiate between a utility value $\widehat{U}_e^{\propto}$ for time-independent CA attributes and a utility value $\overline{\overline{U}}_e^{\propto}$ for time-dependent CA attributes. To additionally consider the case of feasibility determination (e.g. business hours), we further consider the set $F_e^{\propto}$. This set is required to determine the feasibility of the service objects $X_e^{\propto}$, otherwise $F_e^{\propto} = \emptyset$. Moreover, the set $A_e^{\propto}$ is specified as the subset $A_e^{\propto} \subseteq A$ of users that are associated with that dependency e. In case the corresponding CA attribute $\propto$ is referring only to a single user, $|A_e^{\propto}| = 1$ holds for each $e \in E_a^{\propto}$ (e.g. business hours), otherwise $|A_e^{\propto}| > 1$ (e.g. group discounts).

Based on that, a single dependency $e \in E_a^{\propto}$ describing IUR as well as CA attributes is represented by the following 5-tuple (cf. Appendix A for the used notation):

$$e = \left(\widehat{U}_e^{\propto}, \overline{\overline{U}}_e^{\propto}, F_e^{\propto}, A_e^{\propto}, X_e^{\propto}\right) \tag{3}$$

In general, the utility value $\widehat{U}_e^{\propto}$ is distinct from 0 if the corresponding IUR or CA attribute is time independent, and the utility value $\overline{\overline{U}}_e^{\propto}$ is distinct from 0 if the corresponding IUR or CA attribute is time dependent. However, they are both equal 0 and $F_e^{\propto} \neq \emptyset$ if e only refers to feasibility determination. Note, $X_e^{\propto}$ contains one or more decision variables $x_{a_{ij}}$ for each user $a \in A_e^{\propto}$, where $x_{a_{ij}}$ is the binary decision variable corresponding to the service object s_{ij} for user a, and which is used in the optimisation models proposed later on. That is, $x_{a_{ij}}$ is 1 if the corresponding service object s_{ij} is selected for user a, and 0 if not. Further, by breaking down the dependencies of an IUR or CA attribute, it can be assured that the utility determined regarding a single dependency is definite, which means the associated positive or negative utility is realised if – and only if – all service objects in $X_e^{\propto}$ are part of the solution. The same applies for feasibility determination.

In conclusion, when taking CA attributes and IUR into account, the utility and feasibility determination of a service object or set of service objects requires the consideration of other service objects, too. However, we are able to model the resulting context-based and preference-based dependencies through sets of dependencies $E_a^{\propto}$ (with $\propto \in O$, where $IUR \in O$ and $F_e^{\propto} = \emptyset$ for all preference-based dependencies) where the values of $\widehat{U}_e^{\propto}$ and $\overline{\overline{U}}_e^{\propto}$ indicate whether the utility determination of the dependency is of temporal nature or not, and the set $F_e^{\propto}$ whether the feasibility determination is time dependent or not.

4.2 Modelling Temporal-based Dependencies

The consideration of simultaneous IUR and time-dependent CA attributes also leads to dependencies of temporal nature (cf. Table 1 and Table 2). More precisely, the utility or feasibility of a service object/set of service objects depends not only on the selection of other (preceding or succeeding) service objects but also on the exact point in time of their intended usages – and thus on the duration of all preceding service objects of the service composition.

In this context, the possibility *to wait* for the users instead of switching (or being forced to switch) to another, less favoured service object needs to be considered as well. When using waiting time as buffer (if necessary or if it creates higher utility), we need to take into account the service compositions of all users.

Thereby, a concept for modelling and integrating waiting times in an optimisation model is required. Regarding simultaneous IUR, there needs to be the possibility to wait for a user in order to realise a positive utility associated with a complementary simultaneous IUR or to avoid a negative utility associated with the realisation of a conflicting simultaneous IUR. In the case of time-dependent CA attributes, the delay achieved through waiting may enable an infeasible service object to become feasible (e.g. business hours) or may lead to a higher utility (e.g. time-dependent discounts), despite a decrease in utility which may be associated with the waiting time.

To enable this, we introduce the additional NCA attribute waiting time WT (with $WT \in N^{-}$) similar to duration. Moreover, to avoid an increasing complexity when modelling the optimisation problem, we propose special waiting service classes S_i^* right in front of each regular service class S_i as an alternative for a user to wait right between two succeeding regular service classes. Each waiting service class encompasses a set of waiting services where each waiting service $s_{ij}^* \in S_i^*$ is only described by the NCA attribute WT (i.e. all other NFP values are 0) to represent different manifestations of waiting time within one waiting service class. This allows us to model the time consumed by waiting as well as the resulting loss of utility caused by waiting. By placing a waiting service class right before each regular service class as illustrated in Figure 2, the service object selected in the regular service class can be delayed by the amount of WT related to the selected waiting service.

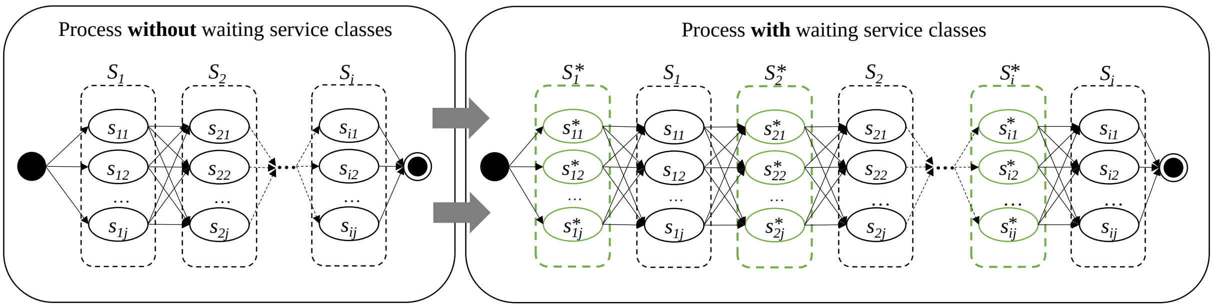

Figure 2. Illustration of a process without and with waiting service classes

As an example, let us consider a user a_1 requesting to use service object $s_{2\,2}$ simultaneously together with user a_2 (i.e. complementary simultaneous IUR), which implies that for realising the utility associated with this IUR both users must use service object $s_{2\,2}$ at the same point in time. Therefore, potential waiting times depend on the duration q_{ij}^{Dur} of the service objects both users have already accomplished so far (here: service objects selected in service class S_1). As a result, three possible alternatives can be distinguished:

a) Waiting is not necessary (e.g. if the aggregated duration of the selected service compositions till using service object $s_{2\,2}$ is the same for both users)

b) Waiting time is proposed for one of the two users (e.g. if the aggregated duration of the selected service compositions until using service object $s_{2\,2}$ is different for the users)
c) Waiting is dispensable (e.g. the IUR and the associated utility will not be realised)

To decide which alternative is the most beneficial, an optimisation model must evaluate if the additional utility realised by the IUR outweighs the loss of utility caused by waiting, which depends upon the amount of waiting time necessary. Considering the entire service composition, this can also lead to the selection of alternative preceding and succeeding service objects. To enable the determination of the right amount of waiting time q_{ij}^{WT}, we propose to model attributes representing 'time' (e.g. duration) as discrete, such that $q_{ij}^{WT}, q_{ij}^{Dur} \in \{k * c | k \in \mathbb{N}_0\}$, with $c \in \mathbb{R}^+$. Thus, each waiting service $s_{ij}^* \in S_i^*$ represents a different discrete manifestation of waiting time (e.g. discrete steps of 15 min). We argue that this seems appropriate for most service selection problems at planning time as the parameter c can be adjusted to every purpose or need.

5 Optimisation Models for a Stateless versus Stateful Representation

To incorporate preference-based, context-based and temporal-based dependencies in an optimisation-based approach, a stateful or a stateless representation can be applied (cf. ❸). In the latter case, dependencies can only be regarded directly within the scope of the optimisation model itself, whereas with a stateful representation the consideration of dependencies could also take place by explicitly modelling a state space in combination with the determination of utility and feasibility. Although both forms of representation are feasible, there are differences regarding criteria like model complexity and computational complexity (cf. Section 6.2 Performance Evaluation).

5.1 Stateless Representation

In the stateless representation, the multi-user context-aware service selection problem can be formulated as knapsack problem where the purpose of the corresponding optimisation model lies in determining the optimal service compositions for all users. Thereby, we propose to use the decision variables $x_{a_{ij}}$ for each user $a \in A$ and every (regular and waiting) service object s_{ij} of the underlying process. Each decision variable $x_{a_{ij}}$ is associated with a utility value $U_{a_{ij}}$ which could possibly be different for each user – subject to the user-defined target weights $w_a^{\propto}$ regarding the NFP. Here, $U_{a_{ij}}$ only represents the utility value for the NCA attributes concerning user a and service object s_{ij}. For utility determination of time-independent and time-dependent CA attributes and IUR, we apply the proposed modelling in terms of the utility values $\hat{U}_e^{\propto}$ and $\bar{\bar{U}}_e^{\propto}$ and the corresponding set of service objects $X_e^{\propto}$. In line with this, we divide our set O of CA attributes and IUR in elements $\hat{O}$ which require time-independent utility determination and those elements $\bar{\bar{O}}$ which require time-dependent utility determination. Thus, for the stateless case, we can formulate our optimisation model, which is non-linear, as follows:

$$\max_{x_{a_{ij}};s_e^\propto} \sum_{a\in A}\sum_{i=1}^{I}\sum_{s_{ij}\in S_i} U_{a_{ij}}x_{a_{ij}} + \sum_{a\in A}\sum_{\propto\in\hat{O}}\sum_{e\in E_a^\propto} \widehat{U}_e^\propto \prod_{x_{a_{ij}}\in X_e^\propto} x_{a_{ij}} + \sum_{a\in A}\sum_{\propto\in\bar{\bar{O}}}\sum_{e\in E_a^\propto} \bar{\bar{U}}_e^\propto s_e^\propto \prod_{x_{a_{ij}}\in X_e^\propto} x_{a_{ij}} \quad (4)$$

$$\text{s.t.} \sum_{i=1}^{I}\sum_{s_{ij}\in S_i} q_{ij}^\propto x_{a_{ij}} \leq Q_a^\propto \quad \forall\propto\in M;\ \forall a\in A \quad (5)$$

$$\sum_{e\in E_a^\propto} q_e^\propto \prod_{x_{a_{ij}}\in X_e^\propto} x_{a_{ij}} \leq Q_a^\propto \quad \forall\propto\in O;\ \forall a\in A \quad (6)$$

$$\sum_{s_{ij}\in S_i} x_{a_{ij}} = 1 \quad \forall i = 1\ to\ I; \forall\, a\in A;\ \text{with } x_{a_{ij}}\in\{0,1\};\ s_e^\propto\in\{0,1\} \quad (7)$$

The objective function (4) determines the accumulated maximum utility over all users $a \in A$, all service classes S_i and all service objects s_{ij} by taking into account the binary decision variables $x_{a_{ij}}$ and $s_e^\propto$ ($x_{a_{ij}} = 1$ indicates that service object s_{ij} is selected for user a, $x_{a_{ij}} = 0$ that is not). The first summand of the function $\sum_{a\in A}\sum_{i=1}^{I}\sum_{s_{ij}\in S_i} U_{a_{ij}}x_{a_{ij}}$ refers to utility determination regarding NCA attributes where no dependencies need to be considered. The second summand represents time-independent utility determination, for example, for mutual IUR. Here, the associated (positive or negative) utility $\widehat{U}_e^\propto$ is realised if the product $\prod_{x_{a_{ij}}\in X_e^\propto} x_{a_{ij}}$ is 1, which is only the case if all service objects given in $X_e^\propto$ are actually selected. In terms of time-dependent utility determination, additional constraints are required to enable the consideration of temporal-based dependencies. This is achieved by the third summand through relating the product of the decision variables $x_{a_{ij}}$ and the associated utility $\bar{\bar{U}}_e^\propto$ to an indicator variable $s_e^\propto$, that is 1 if the corresponding constraints hold and 0 if not. The formulation of the constraints depends upon the specific temporal relationship that needs to be satisfied to realise the utility.

In terms of feasibility determination, constraints (5) and (6) consider the global end-to-end constraints for NCA and CA attributes defined by the users. The consideration of feasibility determination referring to any dependencies between service objects is also achieved by adding constraints to the optimisation model. Similar to the time-dependent utility determination, their concrete formulation depends upon the set $F_e^\propto$. To hold the (standard) condition that for each user $a \in A$ and for every service class S_i exactly one service object must be selected, constraints (7) have also be part of our optimisation model.

Appendix B shows the stateless optimisation model and additional constraints required for time-dependent utility determination in terms of the integration of complementary and conflicting simultaneous IUR.

5.2 Stateful Representation

For our stateful approach, we base upon the concept of belief and world states (cf. Ghallab et al. 2004): Accordingly, a state space consists of one belief state BS_i for each action of the process where each belief state encompasses a set of belief state tuples bst_{ik} (with i referring to the corresponding service class S_i and k as the number of the tuple). Further, each world state $ws_{ik} \subseteq bst_{ik}$ holds exactly one state variable $v(bst_{ik})$ for each context information and its corresponding value. Finally, BS_1 represents the initial state of the process and BS_{I+1} the goal state, accordingly. The utility of a particular service object is then determined in respect of a certain world state, which means based on its quantified non-context and context information as illustrated in Figure 3. These generated state-service combinations (i.e. the state-service space) could then be used within an optimisation model to determine the best service composition for each user with regard to context information. In terms of feasibility determination referring to context-based dependencies, world states and service objects which are not feasible regarding their determined values will not be considered any further.

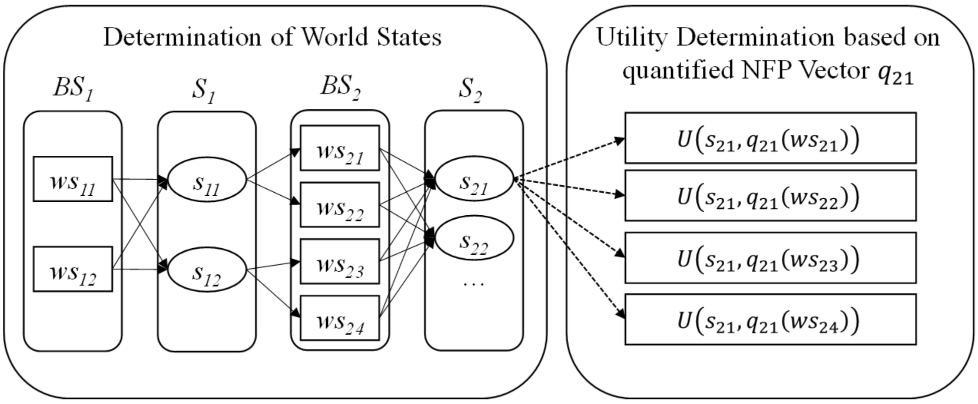

Figure 3. Illustration of utility determination with respect to world states determination (cf. Heinrich and Lewerenz 2015)

The main benefit of such a stateful representation is that the size of the state space for a user remains mostly constant regardless the number of different types of context information considered. But so far, in existing approaches only context-based dependencies in terms of single-user service selection are considered. This means, we need to extend those approaches by both multiple users (cf. ❶) and possible waiting times (cf. ❷). We propose therefore to determine in a first step the state space for each user $a \in A$ separately because each user may have her/his individual initial context (i.e. initial state BS_{a_1}), and determine then context-based and temporal-based dependencies that exist within the user's own service composition. As a result, each user $a \in A$ has its own state space consisting of belief states BS_{a_i}, belief state tuples $bst_{a_{ik}}$ and world state tuples $ws_{a_{ik}}$. Since waiting time and waiting service classes could be modelled as NCA attribute and regular service classes, they also result in belief states BS_{a_i}. To determine the values of the state variables $v\left(bst_{a_{ik}}\right) \in ws_{a_{ik}}$, an existing state-transition algorithm (e.g. Heinrich and Lewerenz 2015) needs to be extended: As the value of each state variable depends upon the corresponding service object and – subject to the type of CA attribute

– also on the preceding world state, the state transition for each variable $v\left(bst_{a_{ik}}\right) \in ws_{a_{ik}}$ could be defined as $v\left(bst_{a_{i'k}}\right) \leftarrow \Phi\left(q^{\propto}_{a_{ij}}, v\left(bst_{a_{ik}}\right)\right)$ [19].

However, dependencies resulting from IUR and CA attributes that exist among different users' service compositions require the determination of a joint state space for all participating users. But the implicit modelling of all possible service combinations regarding all users seems not a very promising approach in terms of computational complexity. Therefore, we propose a different way: For each dependency $e \in E^{\propto}_{a}$ with $|A^{\propto}_{e}| > 1$, we determine the set of associated world states in the created state spaces of the users $a \in A^{\propto}_{e}$. When considering time-dependent utility or feasibility determination (e.g. simultaneous IUR), there could exist more than one of such a set of world states, for instance, referring to different manifestations of daytime $v^{Time}\left(bst_{a_{ik}}\right)$. These sets of world states form the set $Z^{\propto}_{e}$, which is then linked to a new world state $ws^{\propto}_{e}$ addressing the dependency e.

In the optimisation model, the optimal solution over all users could then be calculated based on the determined state-service combinations of all users. In this regard, the objective function is formulated as follows:

$$\begin{aligned} \max_{x_{a_{ij}};\, y_{a_{ik}};\, y^{\propto}_{e}} & \sum_{a \in A} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} \sum_{\substack{ws_{a_{ik}} \\ \in BS_{a_i}}} U\left(s_{ij}, q_{a_{ij}}\left(ws_{a_{ik}}\right)\right) * x_{a_{ij}} * y_{a_{ik}} \\ & + \sum_{a \in A} \sum_{\propto \in O} \sum_{\left\{\substack{e \in E^{\propto}_{a} | \\ |A^{\propto}_{e}|>1 \wedge \\ (\hat{U}^{\propto}_{e} \neq 0 \vee \bar{\bar{U}}^{\propto}_{e} \neq 0)}\right\}} U\left(q^{\propto}_{e}\left(ws^{\propto}_{e}\right)\right) * y^{\propto}_{e} \end{aligned} \tag{8}$$

Similar to the stateless representation, the accumulated maximum utility is achieved by setting the corresponding binary decision variables $x_{a_{ij}}$, $y_{a_{ik}}$ and $y^{\propto}_{e}$. Here, $y_{a_{ik}}$ indicates whether the world state $ws_{a_{ik}}$ for user a is selected or not, and, likewise, $y^{\propto}_{e}$ indicates whether the world state $ws^{\propto}_{e}$ related to a dependency e is selected or not.

The first summand in the objective function (8) encompasses utility determination for all NCA and CA attributes referring to a single user, which means context-based and temporal-based dependencies existing within a user's service composition are considered. Generally, for each service class, only one service object s_{ij} and for each belief state only one world state $ws_{a_{ik}}$ is selectable (see complete model in Appendix C). Further, the second summand deals with utility determination for dependencies existing among different users' service compositions and hence for IUR and CA attributes referring to multiple users. More precisely, $U\left(q^{\propto}_{e}\left(ws^{\propto}_{e}\right)\right)$ corresponds to the utility values $\hat{U}^{\propto}_{e}$ and $\bar{\bar{U}}^{\propto}_{e}$ and is realised if $y^{\propto}_{e} = 1$, which means if the state $ws^{\propto}_{e}$ is

[19] The state transition function Φ for a state variable depends upon the type of the state variable and the corresponding context information.

selected. The required link of $y_e^\propto$ (and $ws_e^\propto$) to the associated service objects $x_{a_{ij}} \in X_e^\propto$ and the determined world state sets $Z_e^\propto$ is achieved through the following constraint:

$$y_e^\propto - \sum_{Z_{e_k}^\propto \in Z_e^\propto} \prod_{\begin{Bmatrix} a \in A_e^\propto | \\ x_{a_{ij}} \in X_e^\propto \end{Bmatrix}} x_{a_{ij}} \sum_{ws_{a_{ik}} \in Z_{e_k}^\propto} y_{a_{ik}} = 0 \tag{9}$$

$$\forall \propto \in O; \forall a \in A; \forall e \in E_a^\propto \text{ with } |A_e^\propto| > 1 \wedge \left(\widehat{U}_e^\propto \neq 0 \vee \overline{\overline{U}}_e^\propto \neq 0\right)$$

By this, dependencies resulting from mutual and simultaneous IUR as well as CA attributes referring to multiple users could be integrated straightforwardly in a stateful representation. The complete optimisation model also encompasses both constraints for considering the users' requirements regarding the NCA and CA attributes and constraints for feasibility determination dealing with dependencies among multiple users' service compositions. As a result, preference-based, context-based and temporal-based dependencies resulting from IUR and CA attributes could be considered upon the state spaces of the users in combination with the optimisation model.

6 Evaluation

In this section, we provide an evaluation of our approach. In detail, we want to show how our approach could provide decision support, which we will evaluate based on the scenario introduced in Section 2 in terms of the criterion *efficacy*. To analyse the computation time of the stateless and stateful model with respect to different multi-user context-aware service selection problems, we additionally evaluate our approach regarding the criterion *performance*. By this, the design of our evaluation follows the compositional styles *demonstration* and *simulation- and metric-based benchmarking of artefacts* (cf. Prat et al. 2015). We use integer programming (Nemhauser and Wolsey 1988) to find the optimal solution for both optimisation models. For this purpose, our presented non-linear optimisation models are transformed into linear ones, which are used throughout the evaluation.

To examine whether our stateless and stateful models provide the optimal service compositions and are consistent to each other, we implemented the linearised versions of the two models in Java and used the mathematical programming solver Gurobi Optimiser[20] for solving them. To ensure a correct implementation, we conducted intensive testing of the source code (i.e. manual analysis by other persons than the programmers, unit tests, JUnit regression tests, runs with extreme values). We then compared the optimal service compositions obtained from our stateless and stateful optimisation models with an exhaustive enumeration (for small problem sizes). In this regard, we analysed the results of over 15,000 randomly generated multi-user context-aware service selection problems (with a maximum problem size related to 16,777,216 possible service compositions). As the solutions were invariably the same for the enumeration, the stateless and the stateful model, we are convinced that our optimisation models are consistent and provide the correct solution.

[20] http://www.gurobi.com/, accessed July 2018

6.1 Efficacy

We analyse the efficacy of our approach in terms of the real-world scenario described in Section 2: A city day trip to Munich, Germany, by three users that encompasses eight different activities (visiting a museum, having lunch, etc.). Using *TripAdvisor*[21] and *Google Places*[22], we determine feasible service objects and their NFP (price, GPS location, business hours, duration) for each of the eight activities, where service objects with no fixed duration are modelled multiple times – each with a different possible manifestation of duration (e.g. a visit of a museum may last 60 min, 90 min, etc.). By this, we consider a process which can be realised by over 2.9 billion possible service compositions per user.

To demonstrate the efficacy of our approach, we compare the solution of i) an existing single-user context-aware service selection approach (i.e. the approach presented by Heinrich and Lewerenz (2015) for each user separately) to the solution of ii) our multi-user context-aware approach (regardless of whether using the stateless or stateful model here as they both provide the same solution). Thereby, we consider – by utilising the information gathered about the available service objects – the NCA attributes *duration* and *price* and the CA attributes *distance* (between two succeeding service objects subject to their GPS location) and *business hours*. Moreover, to get realistic initial contexts as well as target weights and requirements regarding these NCA and CA attributes in our scenario, we conducted a small laboratory experiment with three graduated students named *Pam*, *Marc* and *Dan* (Table 3). Additionally, we asked each of the students to define four *IUR* (one of each type) which are listed in Table 4. Further, we consider *group discounts* and the NCA attribute *waiting time*. The regarded discrete values of duration and waiting time range from 0 to 120 in steps of 15 min.

Parameter	Pam	Marc	Dan
NCA duration	target weight: 0.1 constraint: 650 min	target weight: 0.05 constraint: 650 min	target weight: 0.1 constraint: 600 min
NCA waiting time	target weight: 0.1 constraint: 30 min	target weight: 0.2 constraint: 20 min	target weight: 0.2 constraint: 80 min
NCA price	target weight: 0.5 constraint: 80 €	target weight: 0.05 constraint: 90 €	target weight: 0.2 constraint: 80 €
CA distance	target weight: 0.1 constraint: 15 km initial context: P+R Froettmaning	target weight: 0.3 constraint: 45 km initial context: Main station	target weight: 0.4 constraint: 10 km initial context: Karlsplatz Stachus
CA business hours	initial context: 11:45 am	initial context: 11:30 am	initial context: 11:30 am
IUR	target weight: 0.2	target weight: 0.4	target weight: 0.1

Table 3. Parameter settings retrieved by the laboratory experiment

[21] http://www.programmableweb.com/api/tripadvisor, accessed July 2018
[22] http://www.programmableweb.com/api/google-places, accessed July 2018

Defining User	Type of IUR	Referred Users	Action		Service Object		Utility
			ID	Name	ID	Name	
Pam	compl. simultaneous	2, 3	2	Lunch	3	Bavarese	+ 0.6
Pam	compl. mutual	2	3	Sight	0	Kaufinger- und Neuhauser Strasse	+ 0.04
Pam	confl. simultaneous	3	4	Active Life	4	Botanischer Garten Muenchen	- 0.2
Pam	confl. mutual	2, 3	8	Culture	5	Deutsches Theater Muenchen	- 0.14
Marc	compl. simultaneous	1, 3	7	Nightlife	3	CA-BA-LU	+ 0.08
Marc	compl. mutual	1, 3	4	Active Life	8	Froettmaninger Berg	+ 0.2
Marc	confl. simultaneous	1	3	Sight	4	Muenchner Freiheit	- 0.04
Marc	confl. mutual	1	3	Sight	7	Maximilianeum - Bayerischer Landtag	- 0.36
Dan	compl. simultaneous	2	5	Café	-	-	+ 0.08
Dan	compl. mutual	1	2	Lunch	7	Restaurant Al Paladino	+ 0.04
Dan	confl. simultaneous	1	7	Nightlife	9	Loretta	- 0.06
Dan	confl. mutual	1	7	Nightlife	0	Ryans Muddy Boot	- 0.02

Table 4. IUR specified for city day trip

	User	Optimal Service Composition	Duration (min)	Wai-ting Time (min)	Dis-tance (km)	Price (€)	Group Dis-count (€)	Rea-lised IUR
i) Existing Approaches	**Pam**	$s_{1\,18}$, $s_{2\,20}$, $s_{3\,8}$, $s_{5\,26}$, $s_{6\,16}$, $s_{7\,11}$	540	./.	12.801	60.00	./.	./.
	Marc	$s_{1\,11}$, $s_{2\,20}$, $s_{3\,8}$, $s_{5\,26}$, $s_{6\,16}$, $s_{7\,19}$	540	./.	3.820	65.00	./.	./.
	Dan	$s_{1\,1}$, $s_{2\,28}$, $s_{4\,27}$, $s_{5\,7}$, $s_{6\,4}$, $s_{7\,1}$	450	./.	5.451	60.00	./.	./.
ii) Multi-User Context-Aware Approach	**Pam**	$s_{1\,18}$, $s_{2\,20}$, $s_{3\,10}$, $s_{5\,26}$, $s_{6\,10}$, $s_{7\,11}$	555	0	12.834	60.00	0.00	1
	Marc	$s_{1\,10}$, $s_{2\,28}$, $s_{3\,10}$, $s_{5\,27}$, $s_{6\,14}$, $s_{7\,11}$	555	0	6.690	58.00	2.00	0
	Dan	$s_{1\,10}$, $s_{2\,8}$, $s_{4\,17}$, $s_{5\,27}$, $s_{6\,4}$, $s_{7\,1}$	450	45	5.503	58.00	2.00	1

Table 5. Solution of i) existing approaches versus ii) multi-user context-aware approach for a city day trip scenario

Given this setting, we compare the results of both approaches i) and ii), which means, the optimal service composition for each user and the corresponding NFP values (cf. Table 5): Considering service class *5) Café* and the users Marc and Dan in approach ii), we recognise that – in contrast to i) – for both users the same service object $s_{5\,27}$ (referring to a café named 'Puck') is selected. This can be directly ascribed to the realisation of the complementary simultaneous IUR ‘Dan requests to take a coffee together with Marc regardless which café’ (cf. ❶), but which also requires Dan to wait 45 min in total. However, for Dan the realisation of that IUR is still of higher value than waiting 45 min, which means, the positive utility $\overline{\overline{\overline{U}}}_e = 0.08$ Dan associated with that IUR is able to compensate the loss of utility resulting from waiting. Another realised complementary but mutual IUR is ‘Pam requests to visit the sight “Kaufinger and Neuhauser Street” with Marc’ (service object $s_{3\ 10}$). On the other side, the conflicting mutual IUR ‘Pam requests not to go all together to the “German Theatre Munich”’ (service class *8) Culture*) is not realised as none of the three users visits that theatre. Consequently, the utility of Pam’s overall service composition is not decreased by the associated negative utility $\widehat{U}_e = -0.06$. Furthermore, because of a group discount of 2.00 € each in approach ii) both Marc and Dan visit the museum ‘Pinakothek of Modern Art’ ($s_{1\ 10}$) and thus achieve a lower price (resulting in a higher utility) compared to i). To be able to go to the favoured dinner restaurant with respect to its business hours, in approach i) Dan needs to spend 15 min longer in one of the previous actions since the option to wait is not considered. In approach ii) instead, he waits 15 min as he prefers waiting over spending more time than favoured in one of the other actions (cf. ❷). This analysis illustrates the efficacy when considering ❶-❷ in a multi-user context-aware service selection which is also supported by the discussion of the results with the three graduated students participating in the scenario.

6.2 Performance

In this section, we analyse the stateless and stateful models with respect to their performance, which means, the computation time needed by them for solving multi-user context-aware service selection problems. With evaluating a NP-hard problem (Abu-Khzam et al. 2015) and an approach determining the optimal solution, we expect an over-proportional growth in computation time with increasing problem size (Nemhauser and Wolsey 1988). Computation time in the context of service selection usually depends on several parameters (Alrifai and Risse 2009). The influence of parameters referring to traditional single-user service selection, such as *number of service classes*, *number of service objects*, *number of considered NFP*, and so on, has already been studied thoroughly in literature. Thus, we focus on parameters related to our contribution ❶-❸: i) the *number of users*, ii) the *number of IUR*, iii) the *number of CA attributes*, and iv) the *number of waiting services* per waiting service class.

For our evaluation, we conduct a simulation experiment and an artificial dataset with randomly generated values. Our *initial problem size* encompasses four regular service classes á six service objects and – to consider waiting time – four waiting service classes á five waiting services per class, where waiting time is increased from 0 to 60 in steps of 15 time units. Further, the problem consists of three users, twelve IUR (i.e. four IUR per user, one of each type), three NCA attributes (duration, waiting time and price) and one CA attribute (distance type as representative for other CA attributes). Appendix D summarises the basic evaluation configuration. Founded on this basic configuration, we use four different scenarios corresponding to the four analysed parameters. In each scenario, one parameter is altered while all other parameters are kept constant as defined in the basic evaluation configuration (i.e. ceteris paribus):

i) The number of users is increased from 2 to 10 in steps of 1
ii) The number of IUR per User is increased from 2 to 10 in steps of 2 (1 complementary and 1 conflicting)
   (a) in terms of mutual IUR (in the absence of simultaneous IUR)
   (b) in terms of simultaneous IUR (in the absence of mutual IUR)
iii) The number of distance type CA attributes is increased from 1 to 10 in steps of 1
iv) The number of waiting services per class is increased from 3 to 10 in steps of 1

For all simulation runs, we use a machine with an Intel Xeon E5-2470 v2 processor with 2.40 GHz, 32 GB RAM, Win7 64bit, Java 1.8, and the mathematical solver Gurobi Optimiser 6.5. We conduct for each setting regarding the four scenarios i) to iv) 200 simulation runs and determine the average computation time (measured in milliseconds [ms]). To be able to compare the results of both optimisation models, the measured computation time encompasses not only the time Gurobi Optimiser needs for solving a model but also the time required for building a model, which includes the state space creation in terms of the stateful representation. In the following, the results are presented (cf. Figure 4-7):

When increasing the *i) number of users*, not only the number of variables and constraints regarding the additional users increase but also the number of dependencies resulting from IUR. As shown in Figure 4, this leads for both models to a continuous increase in computation time. To analyse *ii) the influence of time-independent (mutual) as well as time-dependent (simultaneous) IUR*, we consider them in separate simulation runs (cf. Figure 5). In the case of mutual IUR, the stateless model as well as the stateful model show an apparent slighter increase in computation time compared to simultaneous IUR. This is because mutual IUR only have a minor effect on the number of additional variables and constraints of the optimisation models, whereas for simultaneous IUR also temporal-based dependencies need to be considered, which results in a higher number of constraints. As the state space of the stateful model mostly remains constant in size regardless of the number of considered CA attributes, we do expect the computation time staying pretty much the same for the stateful model when increasing the *iii) number of CA attributes*. As Figure 6 illustrates, this is supported by our simulation experiment. In contrast, the stateless model shows a greater increase in computation time, resulting from the higher number of variables and constraints that must be considered with each additional CA attribute and the corresponding context-based dependencies. When increasing the *iv) number of waiting services* per class from 3 to 10, an increase in computation time is only apparent for the stateful model (cf. Figure 7). This results from the fact that each waiting service increases the state space by adding a new manifestation of daytime and therefore leads to a (significantly) larger state space. The stateless model however seems much more robust here. Indeed, an additional experiment reveals an average computation time of only 348 ms for 150 waiting services per waiting class.

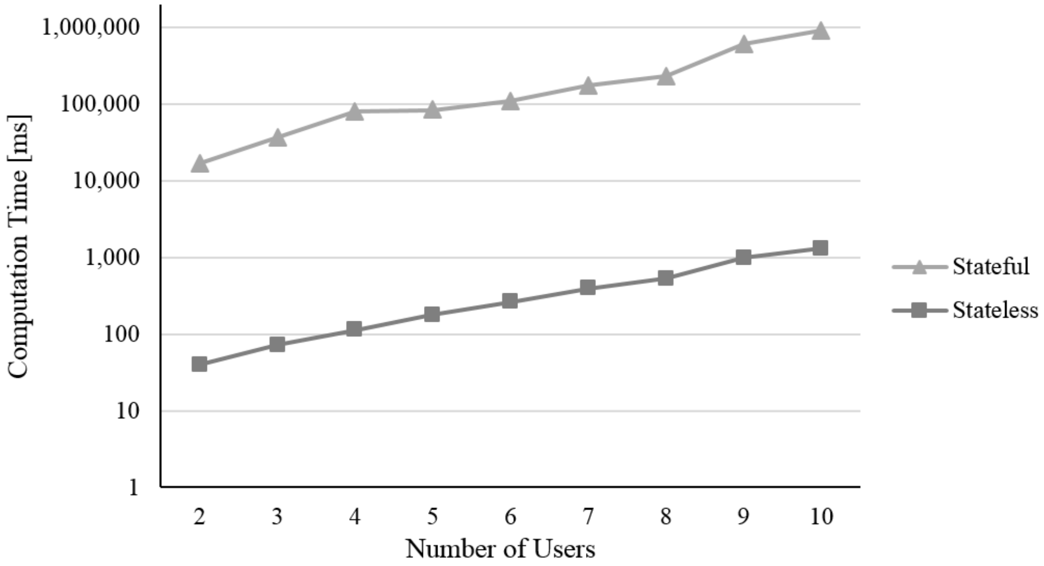

Figure 4. Scenario i)

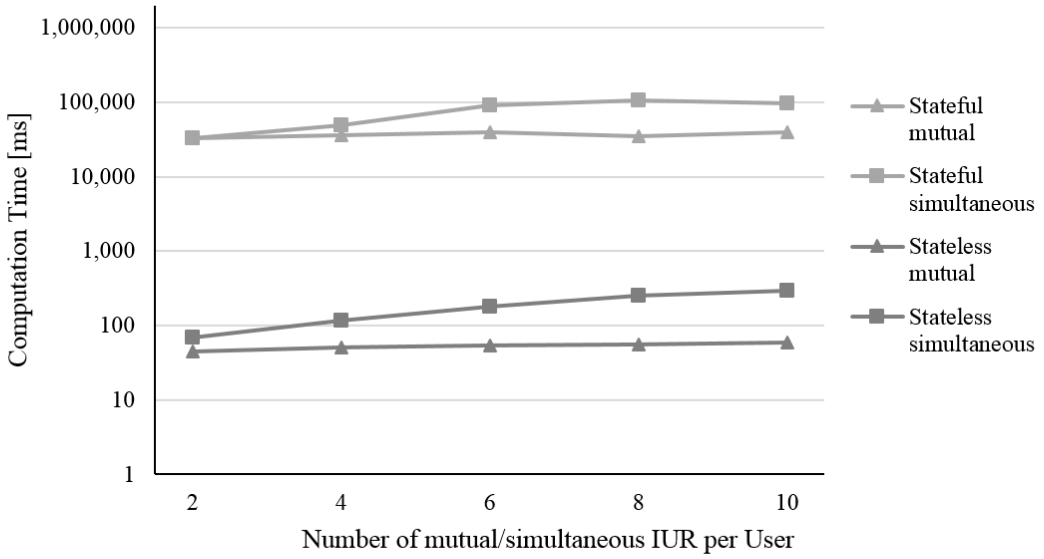

Figure 5. Scenario ii)

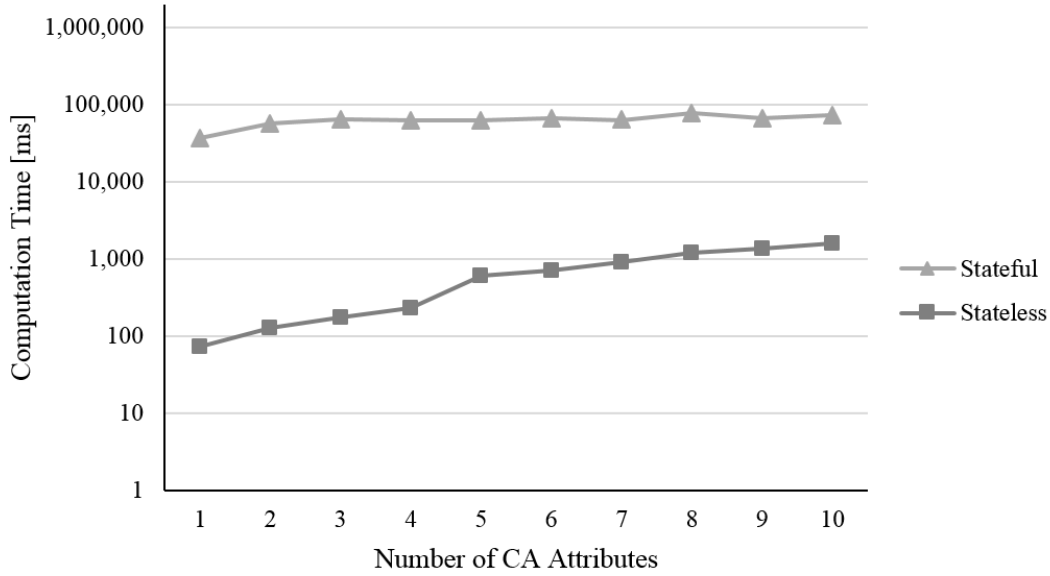

Figure 6. Scenario iii)

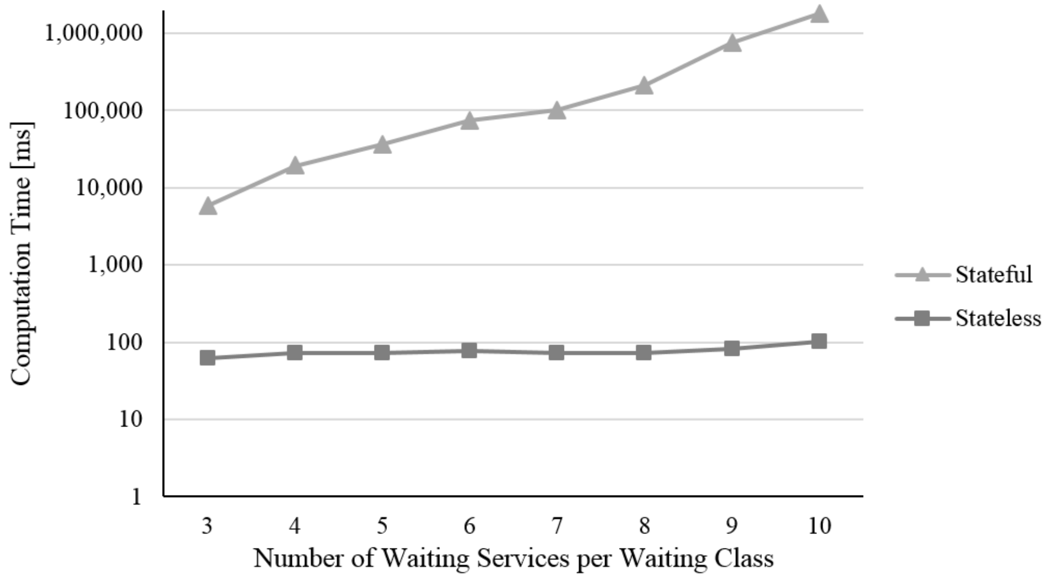

Figure 7. Scenario iv)

To sum up, considering our simulation experiment and scenarios, the performance of the stateless model is obviously much better than the stateful model. The reason is the high number of variables that need to be additionally considered through the creation of the state space. Furthermore, the stateful model appears to be more sensitive regarding the number of waiting services while the stateless model seems to be more sensitive regarding the number of CA attributes. In terms of the number of users and the number of IUR per user, both models show a rather similar change of computation time. As we do not aim to present a computation time optimised approach (e.g. a heuristic) but rather a first approach for a multi-user context-aware service selection at planning time, the computation times especially of the stateless model seem quite acceptable.

7 Discussion

This section discusses theoretical as well as practical implications of our work. Starting with theoretical implications, the multi-user context-aware service selection scenarios described in the paper can also be understood in general as service systems (cf. Alter 2012) – in terms of a context-aware interplay of stationary and mobile devices, services and users (Zaplata et al. 2009). In this regard, collaboration and contextualisation are part of service-dominant design which forms the basis for modern service systems (Alter 2012; Böhmann et al. 2014; Edvardsson et al. 2011). Collaboration (in terms of co-creation and co-consumption) means that the value of a considered service is created by multiple users (Grönroos 2011; Vargo and Lusch 2004). In adoption of the meta model presented by Alter (2012), additional value can be created by a context-aware selection of informational entities (service objects) as resources to perform actions of processes in mobile environments as illustrated in Figure 8.

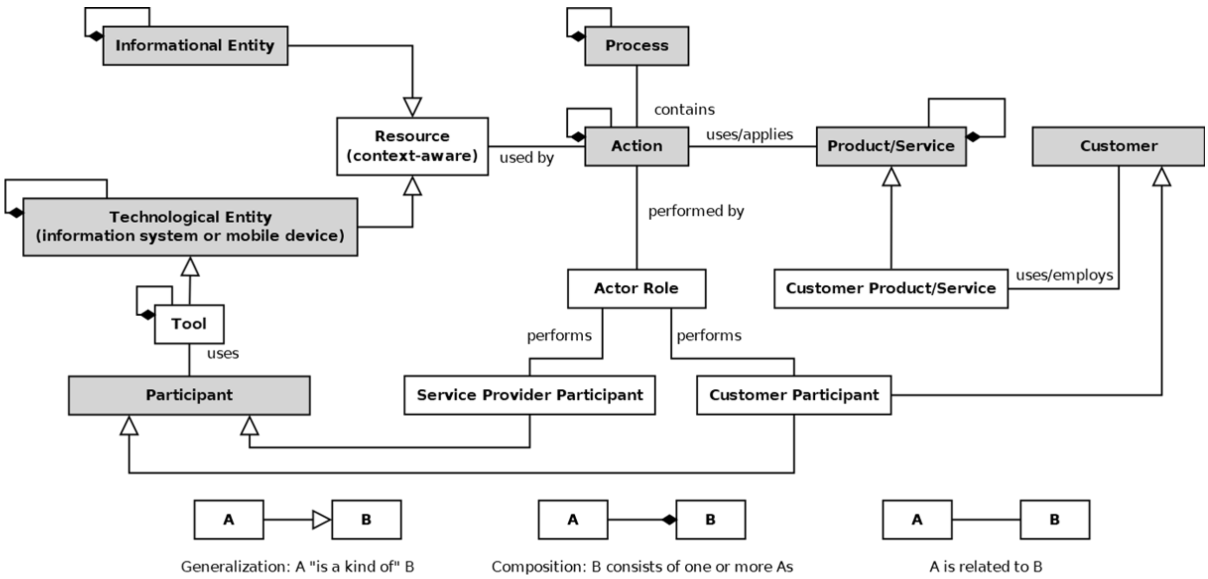

Figure 8. Excerpt of the meta model for a multi-user context-aware service system (based on Alter 2012)

In this meta model, each customer resp. user may conduct his own actions and processes for which informational (in terms of service objects) and technological entities (e.g. mobile devices) need to be selected resp. used as required resources. In mobile environments, this selection is typically context aware. In addition, taking co-consumption in form of IUR into account means that the mutual/simultaneous conduction of some actions by more than one user from otherwise possibly different processes generates additional (positive or negative) value for the users. According to this, analysing and modelling a multi-user context-aware service selection is an actual instantiation of the theoretical construct of a service system, which was proven by this research in order to carefully and specifically evaluate the general construct.

In terms of practical implications, practitioners should be aware that there could be significant advances regarding the optimal service compositions when using the presented approach (cf. Section 6.1). This is not only due to the consideration of dependencies resulting from multiple users and context information. Indeed, by selecting and presenting the optimal service objects for each user regarding an entire service composition, it also addresses the problem of information overload (Zhang et al. 2009) a decision maker may often be confronted with in such situations. By this, we are confident that practitioners could substantially benefit from our work when selecting services (or service objects) for a context-aware process with multiple participating users. For example, we currently plan to validate our approach in an interesting use case together with a big German automotive company: We want to realise our approach in an application for mobile meeting coordination supported in an automated manner. The companies' employees typically attend a lot of meetings every week while not all of these meetings are mandatory but valuable (in different levels) for the employees. However, due to the size of the company meeting planning can be very challenging as the potential participants and meeting rooms are distributed over several facilities. Therefore, the distance (and thus the time) the participants need to cover to get to the location also needs to be considered. Here, we are convinced that an app implementing our approach can support the companies' employees in determining the optimal time and location for a meeting.

8 Conclusion, Limitations and Further Research

Within this work, we presented a multi-user service selection approach, which is to the best of our knowledge the first optimisation-based approach that takes multiple users and context information into account. In this regard, both optimisation models cope with preference-based, context-based and temporal-based dependencies. Existing approaches either focus on context information in terms of single-user service selection or hard restrictions in terms of multi-user service selection (e.g. capacity limits) and neglect potential waiting times when dealing with dependencies of temporal nature.

To address the existing research gap, we first discussed four types of IUR and provided a way to model preference-based and context-based dependencies resulting from these IUR and context information. As considering IUR and context information could also lead to temporal-based dependencies, we further developed a concept for dealing with time especially waiting time by means of introducing waiting service classes and waiting services. Based on this, we presented a stateless as well as a stateful optimisation model to integrate these three types of dependencies. Additionally, by evaluating our approach, we were able to demonstrate its strengths and efficacy by means of a real-world scenario. In this regard, we could also show that in particular our stateless optimisation model could be solved in acceptable time for realistic problem sizes. We therefore contribute to the current body of knowledge in multi-user context-aware service selection.

Besides that, we also need to discuss some limitations of our work, which should be addressed in future research. First, we focused on service selection at planning time and, in this regard, we feel confident that modelling time as discrete seems sufficient in most cases. But there are certainly scenarios in which a consideration of time as quasi-continuous is required. This seems to be relevant, for instance, when selecting service objects at runtime of a process (e.g. re-planning during a city day trip). Although our approach could still consider such runtime scenarios by means of adjusting the factor c as needed (cf. Section 4.2), this would have a negative impact on the problem size and thus the computation time (cf. Scenario iv) of our performance evaluation in Section 6.2). Here, a promising idea may be the use of continuous instead of binary variables for time and waiting time in the stateless model. Second, although our performance evaluation – especially for our stateless model – mostly provided acceptable computation times from a planning point of view, we must account for the fact that the service selection problem is NP-hard which generally corresponds to an exponential development in computation time. Therefore, there are certainly situations where an approach providing an exact solution is not applicable. However, the aim of our work was not to present a computation time-optimised approach. Thus, further studies need to analyse whether and how time-optimised approaches resp. heuristic techniques (e.g. Alrifai et al. 2012; Canfora et al. 2008; Lewerenz 2015) could be developed for our approach to consider IUR and context information in terms of multi-user processes. In addition, we focused on sequential processes. But existing works provide techniques to consider further control flow patterns like parallel, pick or conditional constructs (cf. Ardagna and Pernici 2007; Yu et al. 2007), for instance, by using execution routes (cf. e.g. Alrifai et al. 2012; Ardagna and Pernici 2007; Zeng et al. 2004). By this, our approach can easily be extended in future research to cope with such control flow patterns.

In conclusion, the provided multi-user service selection approach can serve as a promising first step for the aforementioned and further research in this interesting field.

References

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., and Sikora, F. 2015. “On the Complexity of QoS-Aware Service Selection Problem,” in *Service-oriented computing*, A. Barros (ed.), Heidelberg: Springer, pp. 345–352.

Ai, L., and Tang, M. 2008. “QoS-Based Web Service Composition Accommodating Inter-service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm,” in *IEEE Fourth International Conference on eScience,* Indianapolis, IN, USA, pp. 119–126.

Alrifai, M., and Risse, T. 2009. “Combining global optimization with local selection for efficient QoS-aware service composition,” in *Proceedings of the 18th International Conference on World Wide Web,* J. Quemada, G. León, Y. Maarek and W. Nejdl (eds.), Madrid, Spain, pp. 881–890.

Alrifai, M., Risse, T., and Nejdl, W. 2012. “A hybrid approach for efficient Web service composition with end-to-end QoS constraints,” *ACM Transactions on the Web* (6:2), pp. 1–31.

Alter, S. 2012. “Metamodel for Service Analysis and Design Based on an Operational View of Service and Service Systems,” *Service Science* (4:3), pp. 218–235.

Ardagna, D., and Pernici, B. 2007. “Adaptive Service Composition in Flexible Processes,” *IEEE Transactions on Software Engineering* (33:6), pp. 369–384.

Benouaret, K., Benslimane, D., and Hadjali, A. 2012. "Selecting Skyline Web Services for Multiple Users Preferences," in *IEEE 19th International Conference on Web Services (ICWS),* Honolulu, HI, USA, pp. 635–636.

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

Boruff, J. T., and Storie, D. 2014. "Mobile devices in medicine: A survey of how medical students, residents, and faculty use smartphones and other mobile devices to find information," *Journal of the Medical Library Association : JMLA* (102:1), pp. 22–30.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2008. "A framework for QoS-aware binding and re-binding of composite web services," *Journal of Systems and Software* (81:10), pp. 1754–1769.

Damián-Reyes, P., Favela, J., and Contreras-Castillo, J. 2011. "Uncertainty Management in Context-Aware Applications: Increasing Usability and User Trust," *Wireless Personal Communications* (56:1), pp. 37–53.

Deng, S., Huang, L., Hu, D., Zhao, J. L., and Wu, Z. 2016. "Mobility-Enabled Service Selection for Composite Services," *IEEE Transactions on Services Computing* (9:3), pp. 394–407.

Dey, A. K. 2001. "Understanding and Using Context," *Personal and Ubiquitous Computing* (5:1), pp. 4–7.

Edvardsson, B., Ng, G., Zhi Min, C., Firth, R., and Yi, D. 2011. "Does service-dominant design result in a better service system?" *Journal of Service Management* (22:4), pp. 540–556.

Gavalas, D., Konstantopoulos, C., Mastakas, K., and Pantziou, G. 2014. "Mobile recommender systems in tourism," *Journal of Network and Computer Applications* (39), pp. 319–333.

Ghallab, M., Nau, D. S., and Traverso, P. 2004. *Automated planning*: *Theory and practice,* Amsterdam: Elsevier/Morgan Kaufmann.

Grönroos, C. 2011. "Value co-creation in service logic: A critical analysis," *Marketing Theory* (11:3), pp. 279–301.

Guidara, I., Guermouche, N., Chaari, T., Tazi, S., and Jmaiel, M. 2014. "Pruning Based Service Selection Approach Under QoS and Temporal Constraints," in *IEEE International Conference on Web Services (ICWS),* D. de Roure (ed.), Anchorage, AK, USA, Piscataway, NJ: IEEE, pp. 9–16.

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

Heinrich, B., Klier, M., Lewerenz, L., and Mayer, M. 2015. "Enhancing Decision Support in Multi User Service Selection," in *36th International Conference on Information Systems (ICIS),* Fort Worth, USA.

Heinrich, B., and 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* (24:4), pp. 406–432.

Jin, H., Zou, H., Yang, F., Lin, R., and Shuai, T. 2012. "A Novel Method for Optimizing Multi-User Service Selection," *Journal of Convergence Information Technology* (7:5), pp. 296–310.

Kang, G., Liu, J., Tang, M., Liu, X., and Fletcher, K. K. 2011. “Web Service Selection for Resolving Conflicting Service Requests,” in *IEEE International Conference on Web Services,* Washington, DC, pp. 387–394.

Lewerenz, L. 2015. “A Heuristic Technique for an Efficient Decision Support in Context-aware Service Selection,” in *36th International Conference on Information Systems (ICIS),* Fort Worth, USA.

Marynissen, J., and Demeulemeester, E. 2016. “Literature review on integrated hospital scheduling problems,” .

Nemhauser, G. L., and Wolsey, L. A. 1988. *Integer and combinatorial optimization*, New York: Wiley.

Neville, K., O’Riordan, S., Pope, A., Rauner, M., Rochford, M., Madden, M., Sweeney, J., Nussbaumer, A., McCarthy, N., and O‘Brien, C. 2016. “Towards the development of a decision support system for multi-agency decision-making during cross-border emergencies,” *Journal of Decision Systems* (25:sup1), pp. 381–396.

Prat, N., Comyn-Wattiau, I., and Akoka, J. 2015. “A Taxonomy of Evaluation Methods for Information Systems Artifacts,” *Journal of Management Information Systems* (32:3), pp. 229–267.

Sandionigi, C., Ardagna, D., Cugola, G., and Ghezzi, C. 2013. “Optimizing Service Selection and Allocation in Situational Computing Applications,” *IEEE Transactions on Services Computing* (6:3), pp. 414–428.

Shen, Y., Wang, M., Tang, X., Luo, Y., and Guo, M. 2012. “Context-Aware HCI Service Selection,” *Mobile Information Systems* (8:3), pp. 231–254.

Statista. 2017. *Mobile professional services market size worldwide 2014-2018*. https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/. Accessed 20.03.17.

Vanrompay, Y., Pinheiro, M. K., and Berbers, Y. 2009. “Context-Aware Service Selection with Uncertain Context Information,” *Electronic Communications of the EASST* (19).

Vargo, S. L., and Lusch, R. F. 2004. “Evolving to a New Dominant Logic for Marketing,” *Journal of Marketing* (68:1), pp. 1–17.

Ventola, C. L. 2014. “Mobile Devices and Apps for Health Care Professionals: Uses and Benefits,” *Pharmacy and Therapeutics* (39:5), pp. 356–364.

Wanchun, D., Chao, L., Xuyun, Z., and Chen, J. 2011. “A QoS-Aware Service Evaluation Method for Co-selecting a Shared Service,” in *IEEE International Conference on Web Services (ICWS),* Washington, DC, USA, pp. 145–152.

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., and Li, P. 2010. “Web Service Selection for Multiple Agents with Incomplete Preferences,” in *IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology,* Toronto, AB, Canada, pp. 565–572.

Wang, S., Hsu, C.-H., Liang, Z., Sun, Q., and Yang, F. 2014. “Multi-user web service selection based on multi-QoS prediction,” *Information Systems Frontiers* (16:1), pp. 143–152.

Xu, L., and Jennings, B. 2010. "A Cost-Minimizing Service Composition Selection Algorithm Supporting Time-Sensitive Discounts," in *IEEE International Conference on Services Computing (SCC),* Miami, FL, USA, pp. 402–408.

Yu, H. Q., and Reiff-Marganiec, S. 2009. "A Backwards Composition Context Based Service Selection Approach for Service Composition," in *IEEE International Conference on Services Computing,* Bangalore, India, pp. 419–426.

Yu, T., Zhang, Y., and Lin, K.-J. 2007. "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web* (1:1).

Zaplata, S., Kunze, C. P., and Lamersdorf, W. 2009. "Context-based Cooperation in Mobile Business Environments," *Business & Information Systems Engineering* (1:4), pp. 301–314.

Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., and Chang, H. 2004. "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering* (30:5), pp. 311–327.

Zhang, D., Adipat, B., and 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).

Zheng, Z., Zhang, Y., and Lyu, M. R. 2014. "Investigating QoS of Real-World Web Services," *IEEE Transactions on Services Computing* (7:1), pp. 32–39.

Zhou, T., Zheng, X., Song, W. W., Du, X., and Chen, D. 2008. "Policy-Based Web Service Selection in Context Sensitive Environment," in *IEEE Congress on Services Part 1,* Honolulu, HI, USA, pp. 255–260.

Appendix

A. Notation

Notation	Description	Notation	Description
A	set of users participating in the process	$q_e^{\propto}$	quantified NFP value associated with dependency e and attribute $\propto$
$A_e^{\propto}$	set of users associated with dependency e and attribute $\propto$	$q_{a_{ij}}(ws_{a_{ik}})$	quantified NFP value associated with world state $ws_{a_{ik}}$
BS_{a_i}	belief state of user a and service class i	$q_e^{\propto}(ws_e^{\propto})$	quantified NFP value associated with world state $ws_e^{\propto}$ and attribute $\propto$
$bst_{a_{ik}}$	belief state tuple k of user a and service class i	$Q_a^{\propto}$	global end-to-end constraint of user a regarding attribute $\propto$
Dur	NFP duration	s_{ij}	service object j of service class i
$E_a^{\propto}$	set of dependencies of user a regarding attribute $\propto$	s_{ij}^*	waiting service
$F_e^{\propto}$	set of feasibility policies associated with dependency e and attribute $\propto$	S_i	service class/ action i
I	number of service classes of the process	S_i^*	waiting service class
IC_a^{Time}	initial context of user a in terms of daytime at process start	$s_e^{\propto}$	binary decision variable associated with dependency e and attribute $\propto$
IUR	NFP Inter-User-Request	$U_{a_{ij}}$	utility value for user a and service object s_{ij}
J_i	number of service objects in service class i	$\hat{U}_e^{\propto}$	utility value associated with time-independent dependency e and attribute $\propto$
M	the subset of NCA attributes, e.g. duration Dur, waiting time WT	$\bar{\bar{U}}_e^{\propto}$	utility value associated with time-dependent dependency e and attribute $\propto$
N	the set of NCA and CA attributes (NFP)	$v(bst_{a_{ik}})$	state variable for belief state tuple $bst_{a_{ik}}$
N^+	subset of attributes that need to be maximised	$w_a^{\propto}$	target weight of user a regarding attribute $\propto$
N^-	subset of attributes that need to be minimised	$ws_{a_{ik}}$	world state k of user a and service class i
O	the subset of CA attributes, e.g. IUR, *group discounts*	$ws_e^{\propto}$	world state for dependency e and attribute $\propto$
$P_{max}^{\propto}$	maximum quantified NFP value for attribute $\propto$ aggregated over all service classes	WT	NFP waiting time
$P_{i,max}^{\propto}$	maximum quantified NFP value for attribute $\propto$ of service class i	$x_{a_{ij}}$	binary decision variable for service object s_{ij} and user a
$P_{min}^{\propto}$	minimum quantified NFP value for attribute $\propto$ aggregated over all service classes	$X_e^{\propto}$	set of service objects associated with dependency e and attribute $\propto$
$P_{i,min}^{\propto}$	minimum quantified NFP value for attribute $\propto$ of service class i	$y_{a_{ik}}$	binary decision variable for $ws_{a_{ik}}$
$q_{ij}^{\propto}$	quantified NFP value of service object s_{ij} regarding attribute $\propto$	$y_e^{\propto}$	binary decision variable for $ws_e^{\propto}$

$q_{a_{ij}}^{\propto}$	quantified NFP value of user a for service object s_{ij} regarding attribute $\propto$	$Z_e^{\propto}$	set of world states associated with dependency e and attribute $\propto$

B. Stateless Optimisation Model

$$\max_{x_{a_{ij}};s_e^{\propto}} \sum_{a\in A}\sum_{i=1}^{I}\sum_{s_{ij}\in S_i} U_{a_{ij}}x_{a_{ij}} + \sum_{a\in A}\sum_{\propto\in\hat{O}}\sum_{e\in E_a^{\propto}} \hat{U}_e^{\propto}\prod_{x_{a_{ij}}\in X_e^{\propto}} x_{a_{ij}} + \sum_{a\in A}\sum_{\propto\in\bar{\bar{O}}}\sum_{e\in E_a^{\propto}} \bar{\bar{U}}_e^{\propto}s_e^{\propto}\prod_{x_{a_{ij}}\in X_e^{\propto}} x_{a_{ij}} \tag{1}$$

$$s.t. \quad \sum_{s_{ij}\in S_i} x_{a_{ij}} = 1 \quad \forall i = 1,\dots,I;\ \forall\, a\in A \tag{2}$$

$$\sum_{i=1}^{I}\sum_{s_{ij}\in S_i} q_{ij}^{\propto}x_{a_{ij}} \le Q_a^{\propto} \quad \forall\propto\in M;\ \forall a\in A \tag{3}$$

$$\sum_{e\in E_a^{\propto}} q_e^{\propto}\prod_{x_{a_{ij}}\in X_e^{\propto}} x_{a_{ij}} \le Q_a^{\propto} \quad \forall\propto\in O;\ \forall a\in A \tag{4}$$

$$\left[\max_{\left\{\substack{a\in A_e^{IUR}|\\ x_{a_{i'j'}}\in X_e^{IUR}}\right\}}\left(\sum_{i=1}^{i'-1}\sum_{s_{ij}\in S_i}\sum_{\propto\in\{Dur,WT\}} q_{ij}^{\propto}x_{a_{ij}} + IC_a^{Time}\right) - \min_{\left\{\substack{a\in A_e^{IUR}|\\ x_{a_{i'j'}}\in X_e^{IUR}}\right\}}\left(\sum_{i=1}^{i'-1}\sum_{s_{ij}\in S_i}\sum_{\propto\in\{Dur,WT\}} q_{ij}^{\propto}x_{a_{ij}} + IC_a^{Time}\right)\right] * s_e^{IUR} = 0 \ \forall e\in E_a^{IUR}, \bar{\bar{U}}_e^{IUR} > 0 \tag{5}$$

$$\left[\max_{\left\{\substack{a\in A_e^{IUR}|\\ x_{a_{i'j'}}\in X_e^{IUR}}\right\}}\left(\sum_{i=1}^{i'-1}\sum_{s_{ij}\in S_i}\sum_{\propto\in\left\{\substack{Dur,\\ WT}\right\}} q_{ij}^{\propto}x_{a_{ij}} + IC_a^{Time}\right) - \min_{\left\{\substack{a\in A_e^{IUR}|\\ x_{a_{i'j'}}\in X_e^{IUR}}\right\}}\left(\sum_{i=1}^{i'-1}\sum_{s_{ij}\in S_i}\sum_{\propto\in\left\{\substack{Dur,\\ WT}\right\}} q_{ij}^{\propto}x_{a_{ij}} + q_{i'j'}^{Dur}x_{a_{i'j'}} + IC_a^{Time}\right)\right] * (1 - s_e^{IUR}) \ge 0 \tag{6}$$

$\forall e\in E_a^{IUR}, \bar{\bar{U}}_e^{IUR} < 0$

$$\text{with } x_{a_{ij}} \in \{0,1\};\ s_e^{\propto}\in\{0,1\} \tag{7}$$

C. Stateful Optimisation Model

$$\max_{x_{a_{ij}};\, y_{a_{ik}};\, y_e^{\propto}} \sum_{a\in A}\sum_{i=1}^{I}\sum_{s_{ij}\in S_i}\sum_{\substack{ws_{a_{ik}}\\ \in BS_{a_i}}} U\left(s_{ij}, q_{a_{ij}}\left(ws_{a_{ik}}\right)\right) * x_{a_{ij}} * y_{a_{ik}} + \sum_{a\in A}\sum_{\propto\in O}\sum_{\left\{\substack{e\in E_a^{\propto}|\\ |A_e^{\propto}|>1\wedge\\ (\hat{U}_e^{\propto}\neq 0\vee \bar{\bar{U}}_e^{\propto}\neq 0)}\right\}} U\left(q_e^{\propto}(ws_e^{\propto})\right) * y_e^{\propto} \tag{1}$$

$$s.t. \sum_{s_{ij}\in S_i} x_{a_{ij}} = 1 \quad \forall i = 1\ to\ I;\ \forall\ a \in A \tag{2}$$

$$\sum_{ws_{a_{ik}}\in BS_{a_i}} y_{a_{ik}} = 1 \quad \forall i = 1\ to\ I;\ \forall\ a \in A \tag{3}$$

$$\sum_{ws_{a_{ik}}\in BS_{a_i}} \left(y_{a_{ik}} * \sum_{\left(s_{(i-1)j},\, ws_{a_{(i-1)k'}}\right)\in\Theta_{ik}} \left(x_{a_{(i-1)j}} * y_{a_{(i-1)k'}}\right)\right) = 1 \quad \forall a \in A;\ i \in [2; I] \tag{4}$$

$$y_e^{\propto} - \sum_{Z_{e_k}^{\propto}\in Z_e^{\propto}} \prod_{\left\{\substack{a\in A_e^{\propto}|\\ x_{a_{ij}}\in X_e^{\propto}}\right\}} x_{a_{ij}} \sum_{ws_{a_{ik}}\in Z_{e_k}^{\propto}} y_{a_{ik}} = 0 \tag{5}$$

$\forall \propto \in O; \forall a \in A; \forall e \in E_a^{\propto}$ with $|A_e^{\propto}| > 1 \wedge \left(\hat{U}_e^{\propto} \neq 0 \vee \bar{\bar{U}}_e^{\propto} \neq 0\right)$

$$\sum_{Z_{e_k}\in Z_e} \prod_{\left\{\substack{a\in A_e^{\propto}|\\ x_{a_{ij}}\in X_e^{\propto}}\right\}} x_{a_{ij}} \sum_{ws_{a_{ik}}\in Z_{e_k}} y_{a_{ik}} = 0 \tag{6}$$

$\forall \propto \in O; \forall a \in A; \forall e \in E_a^{\propto}$ with $|A_e^{\propto}| > 1 \wedge F_e^{\propto} \neq \emptyset$

$$\sum_{i=1}^{I}\sum_{s_{ij}\in S_i} q_{a_{ij}}^{\propto} * x_{a_{ij}} \leq Q_a^{\propto} \quad \forall \propto \in M \text{ and } \forall a \in A \tag{7}$$

$$\sum_{i=1}^{I}\sum_{s_{ij}\in S_i}\sum_{ws_{a_{ik}}\in BS_{a_i}} q_{a_{ij}}^{\propto}\left(ws_{a_{ik}}\right) * x_{a_{ij}} * y_{a_{ik}} + \sum_{\left\{\substack{e\in E_a^{\propto}|\\ |A_e^{\propto}|>1\wedge\\ (\hat{U}_e^{\propto}\neq 0\vee \bar{\bar{U}}_e^{\propto}\neq 0)}\right\}} q_e^{\propto}(ws_e^{\propto}) * y_e^{\propto} \leq Q_a^{\propto} \tag{8}$$

$\forall \propto \in O;\ \ \forall a \in A$

$$\text{with } x_{a_{ij}} \in \{0,1\};\ y_{a_{ik}} \in \{0,1\};\ y_e^{\propto} \in \{0,1\} \tag{9}$$

D. Performance Evaluation: Basic Configuration

Parameter	Regular Service Class	Waiting Service Class
No. of service classes	4	4
No. of service objects	6	5
NCA duration	randomly selected in the interval [15;60] min in steps of 15 min	
NCA waiting time		0 - 60 min in steps of 15 min
NCA price	randomly selected within Gaussian distribution X~N(10, 25) €	
CA distance	randomly selected in the interval [47.95;48.35] for latitude and [11.25;11.90] for longitude	
No. of users	3	
No. of IUR	4 IUR per user (one of each type), with randomly generated utility in the interval [0;0.3] for complementary IUR and [-0.3;0] for conflicting IUR	
Users' target weights regarding NCA & CA attributes	same target weight for duration, waiting time, price, distance and IUR for each user	
Users' constraints regarding NCA & CA attributes	max. possible aggregated NFP value for duration, waiting time, price and distance for each user	
Users' initial context (i.e. GPS position)	randomly selected in the interval [48.06;48.25] for latitude and [11.36;11.72] for longitude for each user	

5 Paper 4: Multi User Context-Aware Service Selection for Mobile Environments – A Heuristic Technique

Status	Published	Full Citation
Accepted	07/2018	Bortlik, M., Heinrich, B., and Mayer, M. 2018. “Multi User Context-Aware Service Selection for Mobile Environments: A Heuristic Technique,” *Business & Information Systems Engineering* (60:5), pp. 415–430.

Link to online version: https://link.springer.com/article/10.1007/s12599-018-0552-2

6 Conclusion

This section provides a summary of the major findings of this thesis, its limitations, and potential starting points for future research.

6.1 Major Findings

The purpose of this thesis was to develop novel QoS-aware service selection concepts and optimization models for processes with multiple users and context information in mobile environments. In subsequent paragraphs, the five major findings are presented. Figure 4 illustrates the primary focal points of these findings with respect to the research phases discussed in Section 1.3.

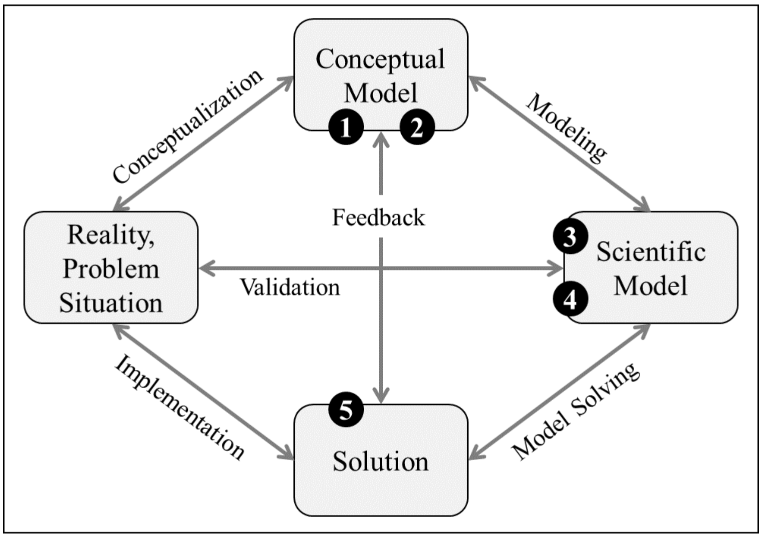

Figure 4. Mapping of Major Findings to Research Phases

❶ Breaking down IUR and context information into their resulting dependencies allows for their formal definition and modeling.

In Papers 1 and 3, the various possible types of IUR and context information were identified and categorized regarding the dimensions of *time* and *relation* for IUR and the dimension of *time* and *number of users* for context information. Based on this, Paper 3 proposed modeling IUR and context information by breaking them down into all their resulting dependencies, as well as provided a generic, unified, formal definition for these dependencies. The definition of any such dependency encompasses the set of affected users (one or more), the set of affected service objects (of one or more users), the associated utility value for determining the overall utility of a service composition (time-independent or time-dependent), and the associated set of feasibility constraints for feasibility determination. Accordingly, this formal definition enables modeling dependencies of all types, regardless of whether they are time-independent or time-dependent and affect only one or multiple users, as well as the utility and/or feasibility of a service composition. For this, IUR and different types of context information were all defined as distinct NFP attributes (in addition to non-context-aware attributes such as price and duration).

Thus, the generic nature of this formal definition supports the practical application by facilitating the integration of different types of IUR and context information in an optimization model (see also Finding 3), to solve the service selection problem at hand.

❷ Temporal coordination of the users can be achieved in terms of a discrete and a continuous time concept, including the consideration of waiting time

The temporal coordination of users' actions is necessary to enable the consideration of temporal-based dependencies resulting from time-dependent IUR and time-dependent context information, because the utility and feasibility of certain service objects also depend on the exact point in time of their execution. In service selection, time is usually represented by the NFP "duration" or "response time" (depending on the scenario). However, for a temporal coordination, this is not sufficient at all because, for instance, potential waiting times must also be taken into account.

Therefore, the concept for considering time in an optimization model developed in Paper 1 proposed introducing "waiting time" as an additional NFP as well as special waiting service classes right before each regular service class of the process. Furthermore, each waiting service class then encompasses a defined number of waiting services – each with a different value for the NFP "waiting time". One fundamental advantage of this concept is that it only utilizes common service selection modeling elements (i.e., services, service classes, and NFP), and thus, does not increase the general complexity when it is integrated into an optimization model. Moreover, the NFP representing "time" ("duration"/"response time" and "waiting time") were modeled in discrete steps. As a lab experiment conducted with graduated students and a performance evaluation showed, this seems to be sufficient for service selection at planning time in many scenarios (cf. Paper 1).

However, it might not be sufficient from a performance and flexibility perspective for service re-selection at execution time. To address this, a continuous time concept was developed in Paper 2 based on the underlying idea of the discrete time concept: Instead of using waiting service classes and several waiting services, the special character of the utility function described by Alrifai and Risse (2009) allows connecting a single waiting time variable with each service class and user, wherein the optimal waiting time (i.e., the optimal value for each waiting time variable) is then determined when solving the optimization model. The conducted performance evaluation supports that the continuous time concept outperforms the discrete time concept above a certain granularity level for the discrete time steps, whereas for scenarios with only a few discrete time steps, the discrete time concept has superior performance. These findings are the results of a simulation experiment. Consequently, practitioners should choose the continuous time concept for service selection problems where a fine granular time concept is required, whereas in other cases, the discrete time concept is sufficient.

❸ Multiple users and context information can be considered within a knapsack optimization model

As discussed in Section 1.1, the traditional single user, non-context-aware service selection problem is often modeled as a knapsack optimization problem. Papers 1, 2, and 3 of this thesis provided a method of integrating multiple users, context information, and the resulting dependencies in a knapsack optimization model:

First, the consideration of multiple users can be realized by introducing a decision variable for each user and service object. In the objective function of the model, these decision variables are then connected with the specific utility values the users associate with the service objects. By additionally integrating the users' global end-to-end constraints, all users can be considered concurrently within a single optimization model. Second, when time-dependent IUR or time-dependent context information must be considered, the required temporal coordination can be achieved by integrating the proposed (discrete or continuous) time concept into the model. Third, based on the formal definition provided for dependencies resulting from IUR and context information, utility and feasibility determination can be completely realized by both the objective function of the model and additional constraints. This model is then able to determine the optimal solution for all users by maximizing the accumulated utility while satisfying all constraints.

The last paragraph refers to the developed *stateless* representation of the optimization model. In addition, Paper 3 also proposed a *stateful* representation to consider multiple users and context information. Herein, the resulting dependencies are considered within a state space and the optimal solution can be determined by solving the corresponding optimization model based on the state space.

- ❹ Re-optimization of multi user processes at execution time can be achieved by taking a global perspective and adjusting a basic optimization model subject to the disruptive events considered

To achieve optimal service re-selection at execution time for a multi user process after the occurrence of a disruptive event, it is necessary to take a global perspective. This means considering the whole process in the optimization model, including the already executed actions of the process (which are then integrated as fixed in the optimization model). This ensures that all global end-to-end constraints of the users and potential dependencies resulting from IUR are feasibly considered, because dependencies could exist between the already executed part of the process and the remaining part. Paper 2 presented such a multi user service re-selection approach that re-optimizes the remaining part of the process for all users. It consists of a basic optimization model, which then requires adjusting and extending subject to the specific characteristics of the event that caused the service re-selection (e.g., a failed service object). This approach covers all three main goals of service re-selection – recovery, feasibility, and optimality (cf. Berbner et al. 2007) – and therefore, can support decision-makers in case of disruptive events occurring at execution time.

- ❺ Applying a heuristic technique based on decomposition of constraints combined with local service selection enables fast decision support for multi user context-aware processes

In Paper 4, a heuristic technique was developed for multi user context-aware service selection considering scenarios where the simultaneous use of the same service object by multiple users is mandatory. This heuristic technique consists of two stages: Multi user-oriented decomposition of all users' global end-to-end constraints into local constraints and ensuing local service selection. The approach is able to consider the dependencies resulting from both multiple users (primarily through decomposition) and context information (primarily through backtracking in the local service selection). As the conducted evaluation of the developed heuristic technique shows, it can overcome the performance issues of exact approaches related to the NP-hardness of the QoS-aware service selection problem.

6.2 Limitations and Further Research

As the discussion of findings and insights in the previous section reveals, the QoS-aware service selection approaches presented in this thesis significantly contribute to the current body of knowledge in that domain. Moreover, this work can be the basis for future research, but also faces some limitations. Both are discussed in subsequent paragraphs.

In terms of limitations, the aim of this thesis was to first develop service selection concepts and models considering multiple users and context information in mobile environments. Therefore, the focus in the presented papers was primarily on sequential processes. However, processes may also encompass other control flow patterns such as pick (cf. Wan et al. 2008; Yu et al. 2007), conditional (cf. Alrifai et al. 2012; Yu et al. 2007), loop (cf. Alrifai et al. 2012; Zeng et al. 2004), or parallel constructs (cf. Alrifai et al. 2012; Ardagna and Pernici 2007). This is especially true when taking multiple users into account because users might need to choose between several actions, conduct one action several times, or conduct more than one action simultaneously. To enable consideration of these non-sequential control flow patterns in multi user context-aware service selection, the idea of execution routes could be adapted (cf., e.g., Alrifai et al. 2012; Ardagna and Pernici 2007; Yu et al. 2007; Zeng et al. 2004). Basically, this means determining the (close-to-)optimal service composition for each possible process path subject to the patterns existing in the process and aggregating or orchestrating them, respectively, into the optimal solution(s). One major challenge here is handling dependencies resulting from multiple users and context information that could exist between different process paths.

Furthermore, the approaches developed in the four papers were primarily illustrated and evaluated regarding real-world examples in tourism, because this domain offers a large amount of real-world data and also has shown its usefulness and attractiveness to mobile users (cf., e.g., Gerpott and Berg 2011; Vos et al. 2008). Other areas where multi user context-aware service selection can potentially provide valuable decision support can be found, for instance, in joint schedule management of several users, coordination of fire workers in emergency situations or field workers in disaster relief management (cf. Fajardo and Oppus 2009; Kartiwi and Gunawan 2013; Monares et al. 2011), roadside assistance and health care (cf. DeRenzi et al. 2011; Marynissen and Demeulemeester 2018; Ventola 2014). Moreover, because of the increasing interconnectedness of the users and the ongoing digitalization of private and business life, it is likely that new applications will continue to emerge.

Apart from these limitations, the work presented in this thesis can also be the starting point for further research: For instance, whereas service selection at planning time was addressed in this study for processes with multiple users and context information, service re-selection – which is required to handle disruptive events occurring at execution time – was considered only for multi user processes (cf. Section 1.4 and Figure 3). Taking also context information into account may significantly enhance the provided decision support in mobile environments. As an example, physical sensors and wireless communication enable real-time updates of context information, such as the current location and time of users, which can then be used in a multi user context-aware service re-selection approach. For this purpose, a heuristic technique would be appropriate because re-selection at execution time requires fast decision support, and multi user context-aware service (re-)selection problems are usually of high complexity (cf. Papers 3 and 4). Consequently, this heuristic technique would be required to handle the dependencies resulting from multiple users (with IUR) and context information, as well as disruptive events occurring at execution time. Future research may analyze whether the decomposition-based heuristic technique proposed in Paper 4 could be extended to adequately deal with this or whether a novel approach must be developed. If a novel approach must be developed, a heuristic technique based on a genetic algorithm may be a promising starting point: Genetic algorithms are inspired by biological evolution and have already been utilized for service re-selection by other researchers (e.g., Canfora et al. 2008). They usually start with a first generation of possible solutions and aim to iteratively improve the solutions in succeeding generations through mutation and crossover (cf. Zhang et al. 2016). Referring to multi user context-aware service re-selection, the consideration of multiple users could possibly be supported by several genetic algorithmic representations in parallel (cf. Luque and Alba 2011), whereas dependencies existing within and among different users' service compositions may be taken into account during the mutation and crossover phase (cf. Ai and Tang 2008; Yuan et al. 2013; Zhang et al. 2013b).

To accomplish QoS-aware service selection, a defined process (model) is first required. Therefore, an entity (person or software) must first plan and specify the process. In terms of multiple participating users and context information, this planning process may be quite challenging: For instance, referring to the city trip example introduced in Section 1.1, users probably have different preferences for the type of actions to conduct on their day trip and in what order. Furthermore, the process must be aligned with the process exogenous context information, such as the weather conditions (e.g., heavy rain impedes visiting an outdoor sight). Here, automated planning approaches (cf., e.g., Heinrich et al. 2012; Henneberger et al. 2008; Hoffmann et al. 2009) could support the process planner. More precisely, these approaches aim to deliver a defined process model that considers users' preferences regarding the process (e.g., Heinrich et al. 2018) as well as process exogenous context information (e.g., Heinrich and Schön 2015). Consequently, to achieve consistent automatic support for the planning and succeeding service selection of a multi user context-aware process, future studies could analyze how existing automatic planning approaches can be combined with the service selection approaches presented in this work. A promising point of departure could be the stateful service selection model presented in Paper 3, because several existing process planning algorithms (e.g., Heinrich et al. 2018; Heinrich and Schön 2015; Henneberger et al. 2008; Hoffmann et al. 2009) also utilize a stateful representation. Furthermore, a chance might exist to already incorporate potential multi user dependencies (i.e., IUR, mandatory conduct of the same action by several users) when planning the process model (cf. Heinrich et al. 2018).

Finally, the optimization-based service selection approaches developed in this work assume a central decision maker who determines the service compositions for all users. When considering multi user processes, the participating users could forgo a central decision maker and instead attempt to determine a suitable solution for the decision problem in a decentralized manner. Users would then act as autonomous agents, thereby participating in a so-called multi agent system (cf. Wooldridge 2009). A decentralized approach would also imply that the users are able to communicate with each other (cf. Stone and Veloso 2000) to reach a common solution through auction, negotiation, or argumentation (e.g., cf. Wooldridge 2009). Indeed, agent-based service selection approaches already exist in the literature. Usually, such approaches only address single user processes while considering service providers (e.g., Comuzzi and Pernici 2005; Lee et al. 2012; Siala and Ghedira 2011; Wang et al. 2008) or QoS brokers (e.g., Rajendran et al. 2010; Seo and Song 2006) as agents. An exception is Shen et al. (2012b), who proposed auction-based negotiation among users (= agents) as part of their service re-selection approach. However, they did not consider any user preferences in terms of IUR. Therefore, future research could examine how to realize decentralized decision-making in terms of an agent-based service selection approach that enables many-to-many interactions of users, to also consider IUR and context information.

In sum, the concepts and approaches presented in this thesis can help decision-makers to provide suitable decision support for multi user context-aware service selection problems in mobile environments, as well as serve as a profound basis for further research.

7 References (Section 1 and 6)

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., and Sikora, F. 2015. "On the Complexity of QoS-Aware Service Selection Problem," in *Service-oriented computing*, A. Barros (ed.), Heidelberg: Springer, pp. 345–352.

Ai, L., and Tang, M. 2008. "QoS-Based Web Service Composition Accommodating Inter-service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm," in *IEEE Fourth International Conference on eScience*, Indianapolis, IN, USA, pp. 119–126.

Alrifai, M., and Risse, T. 2009. "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proceedings of the 18th International Conference on World Wide Web*, J. Quemada, G. León, Y. Maarek and W. Nejdl (eds.), Madrid, Spain, pp. 881–890.

Alrifai, M., Risse, T., and Nejdl, W. 2012. "A hybrid approach for efficient Web service composition with end-to-end QoS constraints," *ACM Transactions on the Web* (6:2), pp. 1–31.

Ardagna, D., Baresi, L., Comai, S., Comuzzi, M., and Pernici, B. 2011. "A Service-Based Framework for Flexible Business Processes," *IEEE Software* (28:2), pp. 61–67.

Ardagna, D., and Mirandola, R. 2010. "Per-flow optimal service selection for Web services based processes," *Journal of Systems and Software* (83:8), pp. 1512–1523.

Ardagna, D., and Pernici, B. 2006. "Global and Local QoS Guarantee in Web Service Selection," in *Business process management workshops*, C. Bussler (ed.), Berlin: Springer, pp. 32–46.

Ardagna, D., and Pernici, B. 2007. "Adaptive Service Composition in Flexible Processes," *IEEE Transactions on Software Engineering* (33:6), pp. 369–384.

Barry, D. K. 2012. *Web Services, Service-Oriented Architectures, and Cloud Computing*: *The Savvy Manager's Guide*, Burlington: Elsevier Science.

Benouaret, K., Benslimane, D., and Hadjali, A. 2012. "Selecting Skyline Web Services for Multiple Users Preferences," in *IEEE 19th International Conference on Web Services (ICWS)*, Honolulu, HI, USA, pp. 635–636.

Berbner, R., Spahn, M., Repp, N., Heckmann, O., and Steinmetz, R. 2007. "Dynamic Replanning of Web Service Workflows," in *Inaugural IEEE-IES Digital EcoSystems and Technologies Conference*, Cairns, Australia, pp. 211–216.

Bertrand, W. M. J., and Fransoo, J. C. 2002. "Operations management research methodologies using quantitative modeling," *International Journal of Operations & Production Management* (22:2), pp. 241–264.

Boruff, J. T., and Storie, D. 2014. "Mobile devices in medicine: A survey of how medical students, residents, and faculty use smartphones and other mobile devices to find information," *Journal of the Medical Library Association : JMLA* (102:1), pp. 22–30.

Bretthauer, K. M., and Shetty, B. 2002. "The nonlinear knapsack problem – algorithms and applications," *European Journal of Operational Research* (138:3), pp. 459–472.

Camargo, V. C.B., Mattiolli, L., and Toledo, F. M.B. 2012. "A knapsack problem as a tool to solve the production planning problem in small foundries," *Computers & Operations Research* (39:1), pp. 86–92.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2005. "An approach for QoS-aware service composition based on genetic algorithms," *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 1069–1075.

Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. 2008. "A framework for QoS-aware binding and re-binding of composite web services," *Journal of Systems and Software* (81:10), pp. 1754–1769.

Cao, H., Feng, X., Sun, Y., Zhang, Z., and Wu, Q. 2007. "A Service Selection Model with Multiple QoS Constraints on the MMKP," in *IFIP International Conference on Network and Parallel Computing workshops*, K. Li (ed.), Dalian, Liaoning, China, Los Alamitos, Calif.: IEEE Computer Society, pp. 584–589.

Comuzzi, M., and Pernici, B. 2005. "Negotiation Support for Web Service Selection," in *Technologies for E-services*, M.-C. Shan, U. Dayal and M. Hsu (eds.), Berlin: Springer, pp. 29–38.

Damián-Reyes, P., Favela, J., and Contreras-Castillo, J. 2011. "Uncertainty Management in Context-Aware Applications: Increasing Usability and User Trust," *Wireless Personal Communications* (56:1), pp. 37–53.

Dannewitz, C., Pentikousis, K., Rembarz, R., Renault, É., Strandberg, O., and Ubillos, J. 2008. "Scenarios and research issues for a network of information," in *4th International Mobile Multimedia Communications Conference*.

Deng, S., Huang, L., Hu, D., Zhao, J. L., and Wu, Z. 2016. "Mobility-Enabled Service Selection for Composite Services," *IEEE Transactions on Services Computing* (9:3), pp. 394–407.

DeRenzi, B., Borriello, G., Jackson, J., Kumar, V. S., Parikh, T. S., Virk, P., and Lesh, N. 2011. "Mobile phone tools for field-based health care workers in low-income countries," *The Mount Sinai journal of medicine, New York* (78:3), pp. 406–418.

Dey, A. K. 2001. "Understanding and Using Context," *Personal and Ubiquitous Computing* (5:1), pp. 4–7.

Fajardo, J. T. B., and Oppus, C. M. 2009. "A Mobile Disaster Management System Using the Android Technology," *International Journal of Communications* (3:3), pp. 78–86.

Gartner. 2017. *Gartner Says Worldwide Device Shipments Will Decline 0.3 Percent in 2017*. https://www.gartner.com/newsroom/id/3754963. Accessed 05.12.17.

Gerpott, T. J., and Berg, S. 2011. "Determinants of the Willingness to Use Mobile Location-Based Services," *Business & Information Systems Engineering* (3:5), p. 279.

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

Heider, F. 1958. *The psychology of interpersonal relations*, New York: Wiley.

Heinrich, B., Klier, M., Lewerenz, L., and Zimmermann, S. 2015. "Quality-of-Service-Aware Service Selection: A Novel Approach Considering Potential Service Failures and Nondeterministic Service Values," *Service Science* (7:1), pp. 48–69.

Heinrich, B., Klier, M., and Zimmermann, S. 2012. "Automated Planning of Process Models: Towards a Semantic-Based Approach," in *Semantic technologies for business and information systems engineering*: *Concepts and applications,* F. Teuteberg, O. Thomas and S. Smolnik (eds.), Hershey, Pa: IGI Global (701 E. Chocolate Avenue Hershey Pennsylvania 17033 USA), pp. 169–194.

Heinrich, B., and 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* (24:4), pp. 406–432.

Heinrich, B., Schiller, A., and Schön, D. 2018. *The Coordination of Multiple Users within Process Models: An Automated Planning Approach:* Work in Progress.

Heinrich, B., and Schön, D. 2015. "Automated Planning of context-aware Process Models," in *European Conference on Information Systems (ECIS),* Münster, Germany. May 26-29.

Henneberger, M., Heinrich, B., Lautenbacher, F., and Bauer, B. 2008. "Semantic-Based Planning of Process Models," *Proceedings of the 2008 Multikonferenz Wirtschaftsinformatik,* pp. 1677–1689.

Hinkelmann, K., Maise, M., and Thonssen, B. 2013. "Connecting enterprise architecture and information objects using an enterprise ontology," in *Enterprise Systems Conference (ES),* Cape Town, South Africa, pp. 1–11.

Hoffmann, J., Weber, I., and Kraft, F. 2009. "Planning@SAP: An Application in Business Process Management," in *2nd International Scheduling and Planning Applications woRKshop,* Thessaloniki, Greece. September 2009.

Hu, B., Wang, Z., and Dong, Q. 2012. "A Modeling and Reasoning Approach Using Description Logic for Context-Aware Pervasive Computing," in *Emerging Research in Artificial Intelligence and Computational Intelligence,* J. Lei, F. L. Wang, H. Deng and D. Miao (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 155–165.

Jaeger, M. C., Mühl, G., and Golze, S. 2005. "QoS-Aware Composition of Web Services: An Evaluation of Selection Algorithms," in *On the move to meaningful Internet systems 2005: CoopIS, DOA, and ODBASE,* R. Meersman and Z. Tari (eds.), Berlin: Springer, pp. 646–661.

Kamal Bashah, N. S., Kryvinska, N., and van Thanh, D. 2012. "Novel Service Discovery Techniques for Open Mobile Environments and Their Business Applications," in *Exploring services science,* M. Snene (ed.), Berlin, Heidelberg: Springer, pp. 186–200.

Kang, G., Liu, J., Tang, M., Liu, X., and Fletcher, K. K. 2011. "Web Service Selection for Resolving Conflicting Service Requests," in *IEEE International Conference on Web Services,* Washington, DC, pp. 387–394.

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

Kirsch-Pinheiro, M., Vanrompay, Y., and Berbers, Y. 2008. "Context-aware service selection using graph matching," *Proceedings of the 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop,* pp. 1–14.

Lai, Y.-C., and Barkan, C. P. L. 2011. "Comprehensive Decision Support Framework for Strategic Railway Capacity Planning," *Journal of Transportation Engineering* (137:10), pp. 738–749.

Lee, J., Lee, S.-J., Chen, H.-M., and Wu, C.-L. 2012. "Composing web services enacted by autonomous agents through agent-centric contract net protocol," *Information and Software Technology* (54:9), pp. 951–967.

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

Lin, C.-F., Sheu, R.-K., Chang, Y.-S., and Yuan, S.-M. 2011. "A relaxable service selection algorithm for QoS-based web service composition," *Information and Software Technology* (53:12), pp. 1370–1381.

Lin, D., Shi, C., and Ishida, T. 2012. "Dynamic Service Selection Based on Context-Aware QoS," in *Proceedings of the 2012 IEEE Ninth International Conference on Services Computing*, Honolulu, HI, USA. 24-29 June, IEEE Computer Society, pp. 641–648.

Luque, G., and Alba, E. 2011. *Parallel genetic algorithms*: *Theory and real world applications*, Berlin, New York: Springer.

Martello, S., and Toth, P. 1987. "Algorithms for knapsack problems," in *Surveys in combinatorial optimization*, S. Martello, G. Laporte, M. Minoux and C. Ribeiro (eds.), Amsterdam, New York, New York: North-Holland, pp. 213–258.

Mărușter, L., Weijters, T., Vries, G. de, van den Bosch, A., and Daelemans, W. 2002. "Logistic-based patient grouping for multi-disciplinary treatment," *Artificial intelligence in medicine* (26:1-2), pp. 87–107.

Marynissen, J., and Demeulemeester, E. 2016. "Literature review on integrated hospital scheduling problems," .

Marynissen, J., and Demeulemeester, E. 2018. "Literature review on multi-appointment scheduling problems in hospitals," *European Journal of Operational Research* .

Meredith, J. R., Raturi, A., Amoako-Gyampah, K., and Kaplan, B. 1989. "Alternative research paradigms in operations," *Journal of Operations Management* (8:4), pp. 297–326.

Messmer, E. 2008. *Tech Talk: Where'd it Come From, Anyway?* https://www.pcworld.com/article/147698/tech.html. Accessed 05.12.17.

Mitroff, I. I., Betz, F., Pondy, L. R., and Sagasti, F. 1974. "On Managing Science in the Systems Age: Two Schemas for the Study of Science as a Whole Systems Phenomenon," *Interfaces* (4:3), pp. 46–58.

Monares, Á., Ochoa, S. F., Pino, J. A., Herskovic, V., Rodriguez-Covili, J., and Neyem, A. 2011. "Mobile computing in urban emergency situations: Improving the support to firefighters in the field," *Expert Systems with Applications* (38:2), pp. 1255–1267.

Moser, M., Jokanovic, D. P., and Shiratori, N. 1997. "An Algorithm for the Multidimensional Multiple-Choice Knapsack Problem," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences* (E80-A:3), pp. 582–589.

Newman, D. 2015. *Mobile vs. Mobility: Understanding the Business Difference*. https://fowmedia.com/mobile-mobility-understanding-business-difference/. Accessed 05.12.17.

Pinedo, M. 2005. *Planning and scheduling in manufacturing and services*, New York: Springer.

Raij, A., Ghosh, A., Kumar, S., and Srivastava, M. 2011. "Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment," in *Conference proceedings and extended abstracts / the 29th Annual CHI Conference on Human Factors in Computing Systems*, D. Tan, G. Fitzpatrick, C. Gutwin, B. Begole and W. A. Kellogg (eds.), Vancouver, BC, Canada. 7/5/2011 - 12/5/2011, New York, NY: ACM, pp. 11–20.

Rajendran, T., Balasubramanie, P., and Cherian, R. 2010. "An Efficient WS-QoS Broker Based Architecture for Web Services Selection," *International Journal of Computer Applications* (1:9), pp. 110–115.

Schutte, N. S., Malouff, J. M., Bobik, C., Coston, T. D., Greeson, C., Jedlicka, C., Rhodes, E., and Wendorf, G. 2001. "Emotional intelligence and interpersonal relations," *The Journal of social psychology* (141:4), pp. 523–536.

Seo, Y.-J., and Song, Y.-J. 2006. "A QoS Oriented Broker System for Autonomic Web Services Selection," in *Frontiers of high performance computing and networking - ISPA 2006 workshops*, G. Min (ed.), Berlin: Springer, pp. 519–531.

Shen, Y., Wang, M., Tang, X., Luo, Y., and Guo, M. 2012a. "Context-Aware HCI Service Selection," *Mobile Information Systems* (8:3), pp. 231–254.

Shen, Y., Yang, X., Wang, Y., and Ye, Z. 2012b. "Optimizing QoS-Aware Services Composition for Concurrent Processes in Dynamic Resource-Constrained Environments," in *IEEE 19th International Conference on Web Services (ICWS),* Honolulu, HI, USA, pp. 250–258.

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. 2014. "Web services composition: A decade's overview," *Information Sciences* (280), pp. 218–238.

Siala, F., and Ghedira, K. 2011. "A Multi-Agent selection of Web Service providers driven by composite QoS," in *IEEE Symposium on Computers and Communications (ISCC), 2011,* Corfu, Greece, Piscataway, NJ: IEEE, pp. 55–60.

Statista. 2017a. *Mobile phone user penetration as percentage of the population worldwide from 2013 to 2019*. https://www.statista.com/statistics/470018/mobile-phone-user-penetration-worldwide/. Accessed 05.12.17.

Statista. 2017b. *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 16.02.17.

Stein, S., Payne, T. R., and Jennings, N. R. 2009. "Flexible provisioning of web service workflows," *ACM Transactions on Internet Technology* (9:1), pp. 1–45.

Stone, P., and Veloso, M. 2000. "Multiagent Systems: A Survey from a Machine Learning Perspective," *Autonomous Robots* (8:3), pp. 345–383.

Strunk, A. 2010. "QoS-Aware Service Composition: A Survey," in *IEEE 8th European Conference on Web Services (ECOWS), 2010,* A. Brogi (ed.), Ayia Napa, Cyprus. 1/12/2010 - 3/12/2010, Piscataway, NJ: IEEE, pp. 67–74.

Temple, S. 2014. *Inside the 3G Mobile Revolution*. http://www.gsmhistory.com/inside_the_3g_revolution/. Accessed 05.12.17.

Vanrompay, Y., Pinheiro, M. K., and Berbers, Y. 2009. "Context-Aware Service Selection with Uncertain Context Information," *Electronic Communications of the EASST* (19).

Ventola, C. L. 2014. "Mobile Devices and Apps for Health Care Professionals: Uses and Benefits," *Pharmacy and Therapeutics* (39:5), pp. 356–364.

Vos, H. de, Haaker, T., and Teerling, M. 2008. "Consumer Value of Context Aware and Location Based Mobile Services," *BLED 2008 Proceedings*, pp. 50–62.

Vries, G., Vissers, J., and Vries, G. de. 1999. "Logistic control system for medical multi-disciplinary patient flows," in *Monitoring, Evaluating, Planning Health Services*: World Scientific, pp. 141–151.

Wan, C., Ullrich, C., Chen, L., Huang, R., Luo, J., and Shi, Z. 2008. "On Solving QoS-Aware Service Selection Problem with Service Composition," in *Seventh International Conference on Grid and Cooperative Computing*, J. Fan (ed.), Shenzhen, Guangdong, China, Los Alamitos, California: IEEE Computer Society, pp. 467–474.

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

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., and Li, P. 2010. "Web Service Selection for Multiple Agents with Incomplete Preferences," in *IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology,* Toronto, AB, Canada, pp. 565–572.

Wang, M., Liu, J., Wang, H., Cheung, W., and Xie, X. 2008. "On-demand e-supply chain integration: A multi-agent constraint-based approach," *Expert Systems with Applications* (34:4), pp. 2683–2692.

Wang, S., Sun, L., Sun, Q., Li, X., and Yang, F. 2015. "Efficient Service Selection in Mobile Information Systems," *Mobile Information Systems* (2015), pp. 1–10.

Weinhardt, C., Blau, B., Conte, T., Filipova-Neumann, L., Meinl, T., and Michalk, W. 2011. *Business Aspects of Web Services,* Berlin, Heidelberg: Springer.

Wooldridge, M. J. 2009. *An introduction to multiagent systems*, Chichester: Wiley.

Xu, L., and Jennings, B. 2010. "A Cost-Minimizing Service Composition Selection Algorithm Supporting Time-Sensitive Discounts," in *IEEE International Conference on Services Computing (SCC),* Miami, FL, USA, pp. 402–408.

Yu, H. Q., and Reiff-Marganiec, S. 2009a. "A Backwards Composition Context Based Service Selection Approach for Service Composition," in *IEEE International Conference on Services Computing,* Bangalore, India, pp. 419–426.

Yu, H. Q., and Reiff-Marganiec, S. 2009b. "Automated Context-Aware Service Selection for Collaborative Systems," in *Advanced information systems engineering,* P. van Eck, J. Gordijn and R. Wieringa (eds.), Berlin: Springer, pp. 261–274.

Yu, T., and Lin, K.-J. 2005. "Adaptive algorithms for finding replacement services in autonomic distributed business processes," in *Proceedings of Autonomous Decentralized Systems,* Chengdu, Jiuzhaigou, China. April 4-8, 2005, pp. 427–434.

Yu, T., Zhang, Y., and Lin, K.-J. 2007. "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web* (1:1).

Yuan, Y., Zhang, X., Sun, W., Cao, Z., and Wang, H. 2013. "Optimal Web Service Composition Based on Context-Awareness and Genetic Algorithm," in *International Conference on Information Science and Cloud Computing Companion,* Guangzhou, China, pp. 660–667.

Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., and Chang, H. 2004. "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering* (30:5), pp. 311–327.

Zhang, C., Zhang, L., and Zhang, G. 2016. "QoS-Aware Mobile Service Selection Algorithm," *Mobile Information Systems* (2016:12), pp. 1–6.

Zhang, D., Adipat, B., and 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).

Zhang, M., Liu, C., Yu, J., Zhu, Z., and Zhang, B. 2013a. "A correlation context-aware approach for composite service selection," *Concurrency and Computation: Practice and Experience* (25:13), pp. 1909–1927.

Zhang, Z., Zheng, S., Li, W., Tan, Y., Wu, Z., and Tan, W. 2013b. "Leveraging Genetic Algorithm to Compose Web Services in a Context-Aware Environment," in *IEEE International Conference on Systems, Man and Cybernetics*, Manchester, pp. 829–834.

Zheng, Z., and Lyu, M. R. 2010. "An adaptive QoS-aware fault tolerance strategy for web services," *Empirical Software Engineering* (15:4), pp. 323–345.

Zheng, Z., Zhang, Y., and Lyu, M. R. 2014. "Investigating QoS of Real-World Web Services," *IEEE Transactions on Services Computing* (7:1), pp. 32–39.

Zhu, W., Yin, B., Gong, S., and Cai, K.-Y. 2017. "An Approach to Web Services Selection for Multiple Users," *IEEE Access* (5), pp. 15093–15104.