Enhancing Decision Support in Multi User Service Selection

Completed Research Paper

Bernd Heinrich
University of Regensburg
Universitätsstr. 31
93053 Regensburg, Germany
Bernd.Heinrich@ur.de

Mathias Klier
University of Ulm
Helmholtzstr. 22
89081 Ulm, Germany
Mathias.Klier@uni-ulm.de

Lars Lewerenz
University of Regensburg
Universitätsstr. 31
93053 Regensburg, Germany
Lars.Lewerenz@ur.de

Michael Mayer
University of Regensburg
Universitätsstr. 31
93053 Regensburg, Germany
Michael1.Mayer@ur.de

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

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 Srinivasa 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.

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.

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 and Risse 2009; Alrifai et al. 2012; 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, i.e., the capacity is set to 1.

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.

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. 2014) 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:

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

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

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[1]:

$$\begin{aligned} &\max_{x_{ij}} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} U_{ij} * x_{ij} \\ &s.t. \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} q_{ij}^n * x_{ij} \le Q_c^n \ \ \forall n = 1 \ to\ N \\ &\sum_{s_{ij} \in S_i} x_{ij} = 1\ \forall i = 1\ to\ I; x_{ij} \in \{0,1\} \end{aligned} \tag{4}$$

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

[1] 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.

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.[2]

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*[3] and *TripAdvisor*[4] 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[5]. 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.

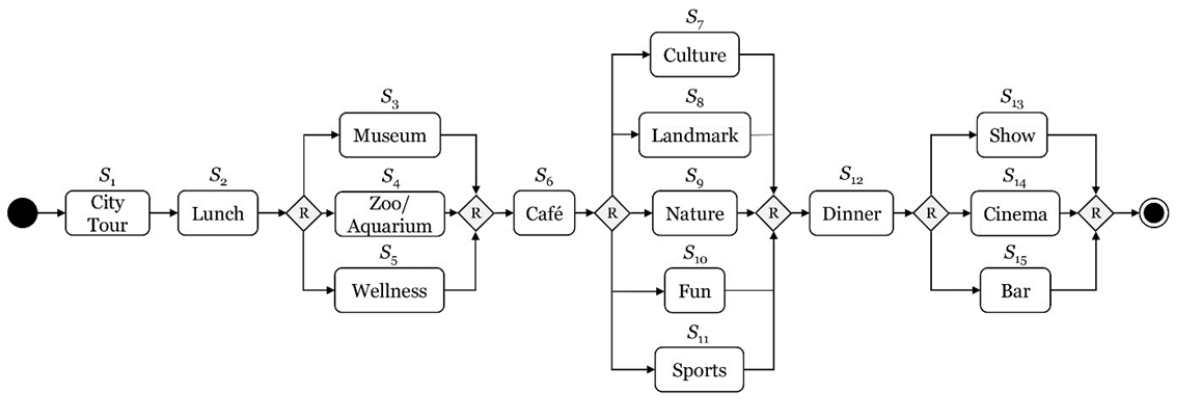

Figure 1. Real-world Process

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 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

---

[2] 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.
[3] http://www.programmableweb.com/api/google-places, accessed September 2015
[4] http://www.programmableweb.com/api/tripadvisor, accessed September 2015
[5] The entire data can be made available on request.

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).

Table 1. Optimal Service Composition per User

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

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.

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.

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:

Table 2. Categories of IUR

	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.

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:

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

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.

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.

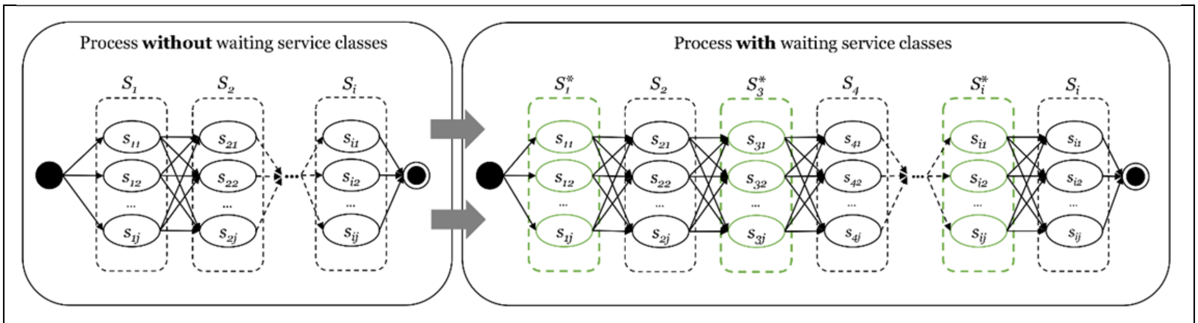

Figure 2. Considering Waiting Service Classes

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 $\overline{\overline{U}}_k$ 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).

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: modelling 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:

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

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

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

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

$$\left[\max_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum_{i=1}^{i'-1} \sum_{s_{ij} \in S_i} \sum_{n \in \{D, WT\}} q_{ij}^n x_{aij} \right) - \min_{\substack{a \in A_k | \\ x_{ai'j'} \in X_k}} \left(\sum_{i=1}^{i'-1} \sum_{s_{ij} \in S_i} \sum_{n \in \{D, WT\}} q_{ij}^n x_{aij} + q_{i'j'}^d x_{ai'j'} \right) \right] * (1 - s_k) \geq 0 \tag{9}$$

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

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

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

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).

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.

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[6], an integer programming solver, was used. To verify these

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

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[7]) 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.

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.[8]

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.

[7] The exact solving of greater problem sizes is not feasible with Excel calculations.
[8] A detailed list of input and output data is left out due to length restrictions. The data can be made available on request.

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

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

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.

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 and Risse 2009; Alrifai et al. 2012; 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, i.e., 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 modelling 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 interdependencies 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, i.e., 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.

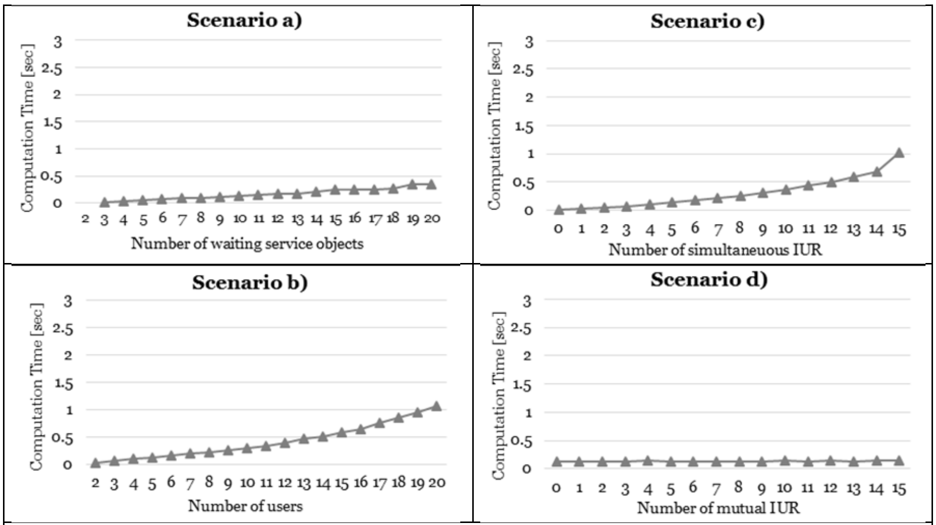

Figure 3. Performance Analysis

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).

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 SODSS 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 modelling 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), Article 7.

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," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*, Washington, DC, USA, 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 *Proceedings of the 2008, International Mobile Multimedia Communications Conference*, European Alliance for Innovation (ed.), Oulu, Finnland, pp. 1-7.

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," *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine* (78:3), pp. 406-418.

Dong, C.-S. J., and Srinivasa, A. 2013. “Agent-enabled Service-oriented Decision Support System,” *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.), Vol. 17, 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. 77-86.

Fletcher, G. J. O., and Fitness, J. (eds.) 1995. *Knowledge structures and interaction in close relationships: A social psychological approach*, Hillsdale, NJ: 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,” in *IEEE Congress on Services Part 1*, Honolulu, HI, USA, pp. 384-388.

Gavalas, D., Konstantopouls, 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* (5), pp. 279-287.

Google 2013. *Our Mobile Planet: United States of America*. http://services.google.com/fh/files/misc/omp-2013-us-en.pdf, Accessed 21 March 2015.

Han, X., Liu, Y., Xu, B., and Zhang, G. 2011. “A Survey on QoS-Aware Dynamic Web Service Selection,” in *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*, Hoboken, NJ, US: John Wiley & Sons Inc.

Heinrich, B., Klier, M., Lewerenz, L., and Zimmermann, S. 2015a. “Quality of Service-aware Service Selection: A Novel Approach Considering Potential Service Failures and Non-Deterministic Service Values,” INFORMS *Service Science*, (7:1), pp 48-69.

Heinrich, B., Klier, M., 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* (in press). doi:10.1080/12460125.2015.1080498

Heinrich, B., and Schön, D. 2015. “Automated Planning of Context-Aware Process Models,” in *Proceedings of the 2015, European Conference on Information Systems*, AIS (ed.), Munster, Germany.

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

Hinkelmann, K., Maise, M., and Thönssen, B. 2013. “Connecting Enterprise Architecture and Information Objects Using an Enterprise Ontology,” in *Proceedings of the 2013, International Conference on Enterprise Systems*, A. Gerber and P. v. Deventer (eds.), 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.), Vol. 10, 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 Bipar-tite 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 (ICWS)*, Washington, DC, USA, 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, 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*, J. Liangzhong (ed.), Berlin, Heidelberg: Springer, 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. v. 1992. *Coordinating plans of autonomous agents*, Berlin New York: Springer Verlag.

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*, T. Wang (ed.), Berlin, Heidelberg: 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 Hofstede, A. T. 2002. "What's in a Service? Towards Accurate Description of Non-Functional Service Properties," *Distributed and Parallel Databases* (12), 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, Department of Computer Science, University of Copenhagen.

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

Salovey, P., Mayer, J. D., and Rosenhan, D. 1991. "Mood and helping: Mood as a motivator of helping and helping as a regulator of mood," *Prosocial behavior*, M. S. Clark (ed.), Vol. 12, Newbury Park, CA: Sage, 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. 2014. "An Approach for Selecting Best Available Services Through a New Method of Decomposing QoS Constraints," *Service Oriented Computing and Applications*.

Tao, F., Zhao, D., Yefa, H., and Zhou, Z. 2010. "Correlation-aware Resource Service Composition and Optimal-Selection in Manufactoring 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 Enviornmental Crisis Management," *Future Generation Computer Systems* (28:3), pp. 593-604.

Vos, H. d., Haaker, T., and Teerling, M. 2008. "Consumer Value of Context Aware and Location Based Services," in *BLED 2008 Proceedings*, AIS (ed.), Slovenia, 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 *Proceedings of the 2008, IEEE International Conference on Grid and Cooperative Computing*, IEEE Computer Society (ed.), Shenzhen, China, 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., 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), Article 6.

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.