Decision support for the usage of mobile information services: A context-aware service selection approach that considers the effects of context interdependencies

Bernd Heinrich[a;*] and Lars Lewerenz[a; 1]

a: *Department of Management Information Systems, University of Regensburg, Universitätsstraße. 31, 93053 Regensburg, Germany*

*: Corresponding author: Bernd.Heinrich@wiwi.uni-regensburg.de (B. Heinrich)
Phone: +49 941 643 6100, Fax: +49 941 943 6120
1: Lars.Lewerenz@wiwi.uni-regensburg.de (L. Lewerenz)
Phone: +49 941 643 6107, Fax: +49 941 943 6120

Acknowledgement:

The Version of Record of this manuscript has been published and is available in Journal of Decision Systems, 11 Sep 2015,

http://www.tandfonline.com/doi/abs/10.1080/12460125.2015.1080498

Decision support for the usage of mobile information services: A context-aware service selection approach that considers the effects of context interdependencies

In mobile business, context information is utilised to select services mostly tailored to a user's current situation and preferences. In existing context-aware service selection approaches, a service utility is determined by comparing its non-functional properties with current context information but without considering its integration in a service composition. This may cause suboptimal selection results, as context information and thus the determined utility of a certain service are usually dependent on its preceding and succeeding services. The latter we denote as *context interdependencies*. In this paper, we investigate how the effects of context interdependencies can be modelled for the context-aware service selection at planning time (i.e. before starting to accomplish a service composition). To develop this approach, we use the concept of states to model context information for the selection. In our evaluation, we find that our approach leads to superior results compared to current context-aware service selection approaches.

Keywords: mobile business, context-aware, service selection, context interdependencies, mobile information service

1 Introduction

The progressive technical development of mobile devices combined with the still-growing spread of fast wireless communication networks enables a mobile way of doing business, so-called mobile business or mobile commerce. According to Heath et al. (2014), the global mobile commerce market was worth US $133 billion dollars in 2013 and is expected to be worth US $627 billion dollars in the year 2018. In this fast growing market, mobile services that can be accessed by a user anytime and anywhere via a mobile device (Okazaki & Mendenz, 2013) play a vital role. Besides their ubiquity, a key advantage of mobile services is that they can be offered to a broad variety of user preferences and situations when being enriched with context information. According to Dey (2001, p.2), context information can be defined as:

> Context information 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 application themselves.

Commonly used context information is the location (by means of a global positioning system [GPS] position), user preferences or favourites (e.g. regarding restaurants), time of day or weather (see Chen & Kotz, 2000; Dhar & Varshney, 2011; Gerpott & Berg, 2011; Zhang, Boonlit, & Yaser, 2009). This information is used for various types of mobile services such as transaction services (e.g. mobile banking, shopping, auctions), communication services (e.g. email, short message service [SMS], instant messaging), navigation services (e.g. google Maps, TomTom) or information services (e.g. Yelp, Foursquare, TripAdvisor, Facebook places or Google; see Daurer, Molitor, Spann, & Manchanda, 2013; Dhar & Varshney, 2011; Gerpott & Berg, 2011; Schumann & Stock, 2014; Vrček, Bubaš, & Bosilj, 2008). Enriched with context information, mobile services open up new ways of creating revenues for businesses (see Luo, Gu, Fang, &

Xu, 2013), whereas for users they allow for the use of support, communication or information that is mostly tailored to the current preferences and situations. For instance, mobile services can be used to provide support in emergency situations (see Dhar & Varshney, 2011), for navigation (e.g. Google Maps) or to find the cheapest gas station on the way home (see Leberknight, 2010). But especially mobile services that provide the user with information, for instance, about cultural heritages (see Aart, Wielinga, & Hage, 2010) or with recommendations for tourist activities (e.g. sights, restaurants, bars, cafés, etc.; see Dhar & Varshney, 2011; Setten, Pokraev, & Koolwaaij, 2004; Zhang et al., 2009) have shown an increase in willingness to use (see Gerpott & Berg, 2011) and in perceived value (see Vos, Haaker, & Teerling, 2008), respectively.

In this paper, we focus on mobile information services. The latter can be understood as a mobile application (e.g. Yelp, Foursquare or TripAdvisor) or a website (e.g. www.yellowpages.com) that is accessed by the user via a mobile device, with the purpose of satisfying his or her information needs regarding a certain topic of interest or activity (e.g. finding a restaurant for dinner). In this matter, the provided information (e.g. about restaurant *a*, restaurant *b*, restaurant *c*, etc.) can be understood as an information respectively service object representing a real life entity (see Dannewitz et al., 2008; Hinkelmann, Maise, & Thönssen, 2013; for reasons of simplicity, we use the terms *service objects* and *service composition* for composed service objects in the following). Service objects can be further denoted by a couple of non-functional properties (see O'Sullivan, Edmond, & Hofstede, 2002) of the represented real life entity (e.g. address or GPS position, business hours, composition costs, etc.) (see Figure 1).

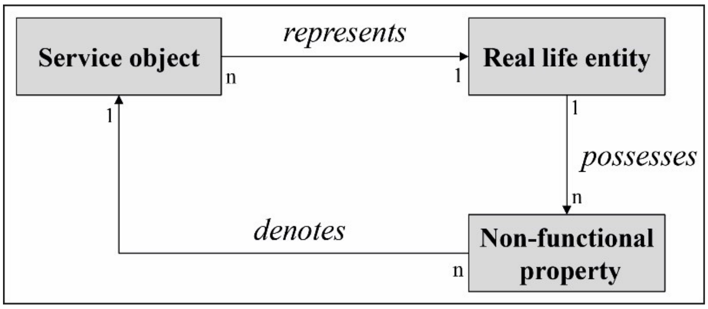


Figure 1 Service object

Based upon the latter, the user is offered the service objects that fit best to his or her current context information (e.g. location or filter). The decision-making of which service object to use for an activity is then performed by the user him-/herself (e.g. which specific restaurant is selected). However, with a growing number of available service objects[1] as well as more context information that is considered, the user might need more and more assistance in selecting a service object for an activity at hand, due to being confronted with an information overload problem (see Zhang et al., 2009). This is even more relevant if not just a single activity but rather multiple activities within a process need to be considered, with several service objects being composed together. In the tourist domain, for instance, a user usually wants to perform several activities one after another (e.g. visiting a museum, seeing an attraction, having dinner, etc.) rather than just a single one. For such a process, finding the optimal service composition while considering a number of non-functional properties and context information may be very difficult or even impossible for the user. Hence, to provide well-founded decision support for the user, the problem of providing a suitable service composition can be understood as a problem of service selection considering context information as well as

[1] TripAdvisor offers, for instance, over 530,000 service objects just for the category “sight” (see Tripadvisor (2015) accessed in 7/2015). Moreover, the number of provided service objects by yelp increased in an exponential fashion in the recent years (see Yelp (2015) accessed in 7/2015).

quantitative non-functional properties. We aim to develop such a service selection approach in the paper. The non-functional properties of a service object can be represented by a set of attributes. More precisely, there are non-context-aware and context-aware attributes which are defined in the following (see also Ai & Tang, 2008b; Gao, Yang, Tang, & Zhang, 2006; Lin & Ishida, 2012; Ramacher & Mönch, 2012; Xu & Jennings, 2010; Yu & Reiff-Marganiec, 2009b).

Definition: A *context-aware attribute* can be characterised by the fact that its *quantified* values (e.g. the *distance* between two subsequently visited real-life entities represented by their service objects) are dependent on context information (e.g. the specific GPS position) arising from a particular service composition in which the service object is integrated. Hence, the quantified values of context-aware attributes are non-fixed for any possible service selection. In case the above characteristic does not hold, we speak of *non-context-aware-attributes*. Figure 2[2] illustrates this definition.

In order to assess the service objects and, consequently, to select the optimal service composition for the accomplishment of a whole process, the different quantified values of both the non-context-aware and the context-aware attributes of a service object are mapped onto a single utility value (see Badidi & Larbi, 2010; Yu & Reiff-Marganiec, 2009b). Moreover, based upon the aggregated quantified values, we can evaluate whether a service composition is feasible with respect to user given requirements by means of end-to-end constraints (e.g. an upper limit regarding the overall distance).

[2] In Figure 2, price, duration and availability are characterized exemplified as *non-context-aware-attributes* even if there may be cases in which they are *context-aware*.

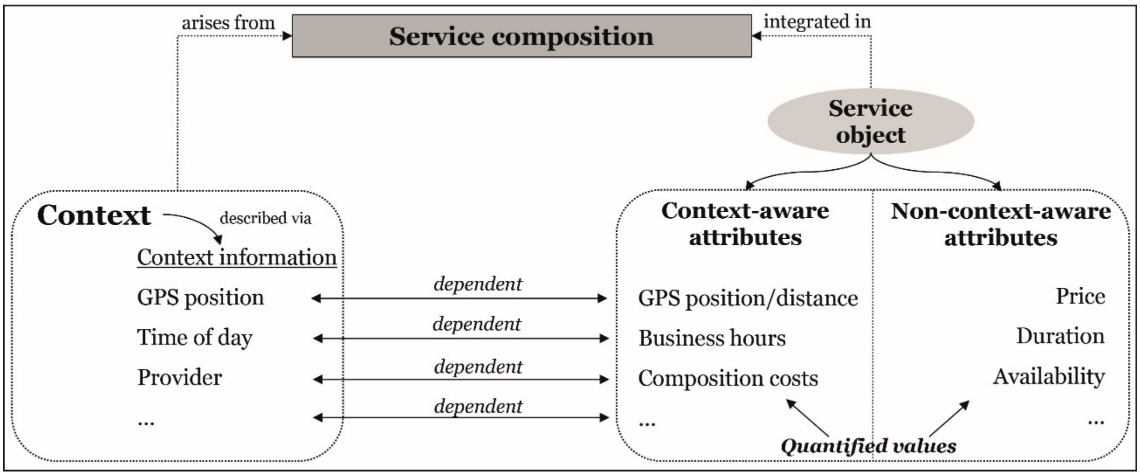


Figure 2 Context-aware and non-context-aware attributes of a service object

When developing a context-aware service selection approach, it is necessary to take the dynamic characteristic of context information into account (see Damián-Reyes, Favela, & Contreras-Castillo, 2011; Kirsch-Pinhero, Yves, & Yolande, 2008; Vanrompay, Kirsch-Pinhero, & Yolande, 2009), which is the cause for the following three effects:

➊ The quantified values of the context-aware attributes of a service object depend on the specific context information, depending itself on prior accomplished service objects. The context information ‘time of day’, for instance, strongly depends on the duration of the service objects previously accomplished. By implication, this means that *different context information* can be determined subject to the considered service composition.

➋ As the quantified values of context-aware attributes are dependent on the context information (see ➊), the *utility of the same service object may be a different one* for each possible service composition. This fact has to be considered when selecting the optimal service objects.

➌ The accomplishment of a service object can lead to context information in which succeeding *service objects are not feasible* with respect to the constraints of the user (e.g. the distance between two subsequently visited real-life entities is too great).

In the following, we denote ➊, ➋ and ➌ as the *effects of context interdependencies*. As these effects have a direct influence on the utility and feasibility of a service composition, they need to be considered when selecting the optimal service objects. Hence, in this paper we aim at developing a novel context-aware service selection approach in which the effects of context interdependencies are considered at planning time (i.e. before starting to accomplish a selected service composition). This comes with several advantages. First, we find that considering the effects of context interdependencies leads to sustainable, better results with respect to the selected service composition and compared to existing approaches (see evaluation section). Second, knowing the optimal service composition already at planning time allows for a preparation of the service composition actually accomplished later on (e.g. for a daytrip in an unknown city: how much money is needed, which shoes and clothes should be taken along for the overall distance and the expected weather, etc.). Existing (non-context aware) re-planning approaches focusing on planning time (e.g. Heinrich, Klier, Lewerenz, and Zimmermann (2015)) are proposed for similar reasons. Finally, there is the possibility of adjusting or refining the results - for example, due to changing constraints. In combination with the ubiquity of mobile services, adjustments can be made (almost) up until the proposed service composition is accomplished (e.g. upon arrival). Further, physical and virtual sensors of a mobile device can further support such adjustments, as they provide important current context information (e.g. for a daytrip - where [GPS position] and when [time of day] to start) for the selection and adjustment of the proposed service composition. In that way, we aim to support decision making for the usage of mobile information services.

Our paper is structured as follows: the next section discusses the background of the context-aware service selection problem. Then, we introduce our model setup showing - in line with the existing literature - how the problem of context-aware service selection can be defined and modelled. Based upon the model setup, we present our approach and thus the contribution of the paper. This approach is capable of considering the effects ❶, ❷ and ❸ of context interdependencies. Afterwards, we evaluate the strength of our approach in comparison with existing context-aware selection approaches. Further, its practical applicability and benefits are demonstrated with the help of a real world example in the tourist domain. We conclude the paper with a discussion of important limitations and an outlook on future research.

2 Background

Our research directly contributes to the literature on (1) context-aware service selection and is related to the literature on (2) service selection based on context-aware quality of service (QoS) and on (3) QoS-aware service selection considering service interdependencies and conflicts.

The literature on (1) context-aware service selection aims to select optimal service objects/service composition based upon context information. To achieve this goal, existing context-aware service selection approaches determine first and mainly at *runtime* the context information for the service objects available *for the next activity* within the process in order to *select the optimal service object* for this activity. We start with a discussion on the organisation and modelling of context information. Concerning this matter, Zheng, Shi, Wang, and Xu (2013) model context information by means of a vector, where each element of the vector holds specific context information such as weather, address or time. Yu and Reiff-Marganiec (2009a) use web ontology language [OWL] to model context information and resource description framework [RDF] to store context data. Each ontology element thereby stores specific context information. Moreover, they differentiate between the user profile context (e.g. personal data), resource context (e.g. which devices are available), activity context (e.g. describing everything a person does) and physical location context (e.g. GPS position). A further approach to modelling and organising context information is the usage of a graph (see Fujii & Suda, 2009; Kirsch-Pinhero et al., 2008; Vanrompay et al., 2009). In the graph, each node represents a property (e.g. context information) and the edges represent the relationship between the nodes. In a second step, the approaches (see Fujii & Suda, 2009; Haddad, Manouvrier, Reiff-Marganiec, & Rukoz, 2010; Kirsch-Pinhero et al., 2008; Vanrompay et al., 2009; Yu & Reiff-Marganiec, 2009a; Zheng et al., 2013) quantify the values of the context-aware attributes (e.g. the GPS position) of each single service object available for the next activity, and subsequently determine its utility. The latter is used to assess the service objects and consequently to select the optimal one for the next activity. Usually, this activity is (immediately) executed right after the selection (see Fujii & Suda, 2009; Kirsch-Pinhero et al., 2008; Vanrompay et al., 2009; Zheng et al., 2013). Opposed to that is the backwards composition context-based service selection (BCCbSS) approach by Yu and Reiff-Marganiec (2009b) and Haddad et al. (2010). Before executing the next activity, the BCCbSS approach additionally selects the optimal service object for the activity after next taking the to-be context information into account. Then, the selection in the next activity is reconsidered. The goal is to determine alternative context information potentially resulting in a higher utility of the selected service objects for the next two activities. If this is the case, a re-selection is performed. Otherwise the initial selection is maintained. Afterwards, the next activity is executed and the approach proceeds further.

In contrast to the runtime approaches above, Yuan, Zhang, Sun, Cao, and Wang (2013) *determine context information* and *select an optimal service object* for every activity of the process (i.e. service composition) *at planning time*. Here, context information is used to provide information about, for instance, price discounts in case certain service objects are composed together. However, the quantification of context-aware attributes is not taken into account. For a single activity of the process and the corresponding service objects, the authors use a matrix to represent different, already quantified values of a single attribute (e.g. price) that can emerge due to all possible service object selections in the preceding activity. Based upon this information, a genetic algorithm is used to determine the service composition resulting in the highest utility.

To sum up, as the effects of context interdependencies have a significant influence on the utility and feasibility of service objects, it is obvious that these effects should be considered when determining the optimal service composition for a process. The analysed literature on (1), however, shows that the current selection approaches leave the effects of context interdependencies (see effect ❶-❸) widely unconsidered. The only exceptions are the approaches by Yu and Reiff-Marganiec (2009b), Haddad et al. (2010) and Yuan et al. (2013), considering context interdependencies between two successive activities but not throughout the entire process. Thus, we contribute to the literature on (1) by considering jointly the effects ❶-❸ of context interdependencies.

A discussion on the influence of context information on service selection takes place in other streams of literature with objectives differing from that in the literature on (1) but that are, nonetheless, related to our research. These streams comprise the literature on (2) service selection based on context-aware QoS, and on (3) QoS-aware service selection considering service interdependencies and conflicts. We will discuss those streams in the following as they provide some ideas that can be transferred to our research.

We start by discussing the literature on (2). Today, many service providers offer different prices for their services based on the amount of the consumed service invocations (see Legner, 2009) or the time of invocation (see Xu & Jennings, 2010). Moreover, the response time of a service can differ in dependence on the distance between the service provider and the service consumer (see Lin & Ishida, 2012). Hence, context information (e.g. number of invocations, time or location) existing for a service invocation influences directly the QoS values of a service (price, response time, etc.) and subsequently its utility, as the two short examples illustrate. The literature on (2) considers such influences of context information. Xu and Jennings (2010) propose a selection approach where time-sensitive intra- and interprovider discounts are considered. They use an expected utility to select services that minimise the costs with respect to a time interval of interest. The latter is broken down into a number of subintervals, with each subinterval possessing a probability of service invocation. Ramacher and Mönch (2012) propose an approach to minimise the expected costs for a given time period while considering quantity and bundle discounts, subscription-based charging, and admission fees. They model their optimisation problem as a mixed integer programming one and use the constraints to consider the described influences of context information (e.g. bundle discounts). In addition to the costs, Lin and Ishida (2012) also consider response time and quality in their optimisation approach. The basic idea of their approach is to predict the QoS values based upon a context-dependency graph (e.g. distance affects response time).

The literature on (3) aims to consider service interdependencies and conflicts occurring for the QoS-aware service selection. A service provider, for instance, may offer a price reduction if two - instead of only one - of his services are composed together. Contrary to that, there are situations where two services from different providers are not allowed to be composed together or cannot be composed due to their interfaces or input and output data. Hence, context information that exists for the selection of services can lead to service interdependencies and conflicts, as these two examples suggest. Gao et al. (2005) proposed a selection approach where interface matching of services is considered. Ardagna and Pernici (2007) proposed an approach that is based upon stateful services where service interdependencies (e.g. one service can execute one or more tasks) are considered by means of the constraints. A quite similar approach was proposed by Ai and Tang (2008a).

In summary, we discussed the literature on (2) and (3) as it offers some valuable ideas that can be transferred to our research. In detail, the approach to consider service interdependencies and conflicts with the help of constraints in an optimisation model is a first promising step. Furthermore, the literature on (2) illustrates the importance of considering the influence that context information can have on the QoS values of a service and thus on the utility of a service composition. It is very clear that these effects need to be considered in a context-aware selection approach to gain valid results.

To conclude this section, to the best of our knowledge, there does not exist any approach that jointly addresses effects ❶-❸ of context interdependencies within a context-aware service selection approach. Thus, in the following, we present an approach that can cope with these challenges.

3 Model setup

We introduce our model setup in line with existing works, referring to those definitions and modelling elements that can serve as a common knowledge base.

In our model, we consider a process that consists of a number of activities *i* (with *i* = 1 to *I*) that contribute to achieving the intended goal of the process. A service class S_i includes all service objects s_{ij} (with *i* = 1 to I, j = 1 to J_i) that provide the equivalent functionality to implement the activity *i,* but differ in their non-functional properties. Further, we define a service composition sc_l (with *l* = *1* to *L*) that is a tuple of service objects s_{ij} which are able to implement all activities *i* of the considered process.

3.1 Non-context-aware and context-aware attributes

We focus on context-aware and non-context-aware attributes both describing the non-functional properties (NFP) of a service object (see definition above). For this purpose, we define *N* as the initial set of attributes (with *n* = 1 to *N*), the subset of non-context-aware attributes as *M* (with *m* = 1 to *M*) and the subset of context-aware attributes as *O* (with *o* = 1 to *O*) such that $M \cup O = N$ and $M \cap O = \emptyset$ holds. We thereby introduce $nca_{ij} = \left[nca_{ij}^1, \dots, nca_{ij}^M\right]^T$ as the NFP vector for a single service object s_{ij} that includes *M* values, each for a single non-context-aware attribute *m*. Further, we define $ca_{ij} = \left[ca_{ij}^1, \dots, ca_{ij}^O\right]^T$ as the NFP vector for a single service object s_{ij} that includes *O* values, each for a single context-aware attribute *o*. For instance, the GPS position as a context-aware attribute can be used to quantify the distance between two successive service objects. For this purpose different quantification functions are needed, which are described in the next section.

3.2 Type based quantification functions

In line with existing literature (e.g. Shen, Wang, Tang, Luo, & Guo, 2012; Yu & Reiff-Marganiec, 2009a), we describe in the following different quantification functions that can be used to determine the quantified values of the context-aware attributes *O* in dependence of context information (see Figure 2). We group the context-aware attributes into three different *types* and provide a quantification function τ^{γ} as well as the needed context information for each type γ.

The first quantification function is for the *Boolean type.* This simple type is used for context-aware attributes where the value needs to satisfy a certain criterion *G* that is context information. Hence, the function is defined as:

$$\tau^{B} = \begin{cases} 1 & if\ ca_{ij}^{o}\ satisfies\ criteria\ G \\ 0 & otherwise \end{cases} \tag{1}$$

Let us focus, for instance, on the context-aware attribute 'business hours'. If the criterion *G* is 'Monday at 10:00 am', then the quantified value of the context-aware attribute business hours ca_{ij}^{o} will be determined to '1' for all service objects where ca_{ij}^{o} satisfy the criterion *G* and to '1' if not.

The second quantification function is for *the discount type* which is an extension of the *Boolean type*. This type of function is used for context-aware attributes where the quantified value is determined with respect to the criterion *G*. Hence, the quantification function is defined as follows:

$$\tau^{CC} = ca_{ij}^{o} * dc\ with \begin{cases} 0 < dc < 1\ if\ \tau^{B} = 1 \\ dc = 1 \quad\ if\ \tau^{B} = 0 \end{cases} \tag{2}$$

In term (2), *dc* is a discount factor. Let us focus, for instance, on the context-aware attribute 'composition costs'. The quantified value of a service object is lower if it is composed with a service object from the same service provider[3] (i.e. criterion *G*). In this case, the discount *dc* is taken into account, which means that the costs for composing

[3] For instance, certain cities offer a pass allowing for the free use of public transportation and discounts for a number of museums and sights (cf. Vienna Tourist Board (2015) accessed in 07/2015).

these service objects are lower compared to a situation in which the two considered service objects are from different providers.

The third quantification function is for the *distance typ*e. This type of quantification function is used for context-aware attributes where the quantified value is determined with respect to the distance to a certain location *LO* determined by context information (e.g. another service object). Thus, the quantification function is defined as follows:

$$\tau^D = \begin{cases} R * c & if\ c \geq 1 \\ R * c * \arcsin(1) & if\ c < 1 \end{cases} \tag{3}$$

$$With\ c = 2 * arcsin\sqrt{sin2\left(\frac{|L2 - L1|}{2}\right) + \cos(L1) * \cos(L2) * sin2\left(\frac{|G2 - G1|}{2}\right)}$$

$$and\ R = 6371km$$

L1 and G1 are the latitude and longitude of the service object s_{ij} (that is the GPS position ca_{ij}^{O}), and L2 and G2 are the latitude and longitude of the service object $s_{i\prime j}$, respectively.

We want to emphasise that this is not a complete list of all quantification functions that exist (see Shen et al., 2012; Yu & Reiff-Marganiec, 2009a). Dependent on the considered use case, other quantification functions can be proposed. In this paper, however, the above quantification functions have proven to be sufficient. With the use of the quantification functions τ^{γ}, context information (e.g. *G* and *LO*) and the vector $ca_{ij} = \left[ca_{ij}^{1}, \dots, ca_{ij}^{O}\right]^{T}$, the quantified values of the context-aware attributes of a service object s_{ij} can be determined. As a result, a service object s_{ij} possesses two NFP vectors ca_{ij} and nca_{ij} where each vector holds a quantified value for an attribute *n*. To determine the utility of a service object in a simple manner, we integrate them within one vector. Thus, we introduce $p_{ij} = \left[p_{ij}^1, \dots, p_{ij}^N\right]^T$ as the *quantified* NFP vector for service object s_{ij} that includes *N* quantified values, each for a single attribute *n*.

3.3 Utility function and constraints

For the selection of service objects for which multiple attributes have to be considered, we use a utility function *U*. The purpose of *U* is to map the different values of the attributes onto a single utility value in order to assess the different service objects and subsequently select the optimal service composition. For this purpose, we differentiate the attributes into three subsets. The first subset contains all attributes that need to be maximised. We denote those attributes with N^+ in the following. The second subset contains all attributes that need to be minimised. We denote those attributes with N^- in the following. Finally, the third subset contains all attributes where a certain target value *tv* needs to be reached (e.g. the lower the difference to *tv* the better). Those attributes are denoted with N^T in the following. For the determination of the utility of a service object, we apply – in line with the existing literature (see Ai & Tang, 2008a, 2008b; Lin & Ishida, 2012) – the simple additive weighting (SAW) technique (Hwang & Yoon, 1981) that consists of two steps: scaling and weighting. In the scaling step, the values of the attributes are normalised in the interval [0;1] in order to achieve comparability between different attributes. To normalise the attributes, we use the (possible) aggregated maximum and minimum values over all service classes S_i (e.g. Alrifai, Risse, & Nejdl, 2012). These values can easily be determined by using the maximum and minimum value over all service objects of a service class S_i. Hence, the aggregated maximum and minimum values for the subsets of the attributes $N^+$and N^- can be defined as:

$$Pmin'(n) = \sum_{i=1}^{I}\left(Pmin'^{(i,n)}\right) \; with \; Pmin'(i,n) = \min_{s_{ij} \in S_i} p_{ij}^{n} \tag{4}$$

$$Pmax'(n) = \sum_{i=1}^{I}\left(Pmax'(i,n)\right) \; with \; Pmax'(i,n) = \max_{s_{ij} \in S_i} p_{ij}^{n} \tag{5}$$

And for the attributes in subset N^{T}, the aggregated maximum and minimum values are defined as (with *tv* as target value):

$$Pmin^{*}(n) = \sum_{i=1}^{I}\left(Pmin^{*}(i,n)\right) \; with \; Pmin^{*}(i,n) = \min_{s_{ij} \in S_i}\left(\left|p_{ij}^{n^T} - tv\right|\right) \tag{6}$$

$$Pmax^{*}(n) = \sum_{i=1}^{I}\left(Pmax^{*}(i,n)\right) \; with \; Pmax^{*}(i,n) = \max_{s_{ij} \in S_i}\left(\left|p_{ij}^{n^T} - tv\right|\right) \tag{7}$$

In the second step of the SAW technique, the normalised values of the attributes are weighted using the preferences of the user. Thus, the utility of a service object $U(s_{ij})$ can be defined as follows:

$$U(s_{ij}) = \Upsilon + \Psi + \Omega$$

$$\Upsilon = \sum_{n^-=1}^{N^-}\left(\frac{Pmax'(i,n^-)-p_{ij}^{n^-}}{Pmax'(n^-)-Pmin'(n^-)}\right) * w^{\mathrm{n}^-}$$

$$\Psi = \sum_{n^+=1}^{N^+}\left(\frac{p_{ij}^{n^+}-Pmin'(i,n^+)}{Pmax'(n^+)-Pmin'(n^+)}\right) * w^{n^+}$$

$$\Omega = \sum_{n^T=1}^{N^T}\left(\frac{Pmax^{*}(i,n^T)-\left(\left|p_{ij}^{n^T}-tv\right|\right)}{Pmax^{*}(n^T)-Pmin^{*}(n^T)}\right) * w^{n^T} \tag{8}$$

Considering $U(s_{ij})$, $p_{ij}^{n^+}, p_{ij}^{n^-}$ and $p_{ij}^{n^T}$ are the quantified values for each single attribute *n* of the NFP vector of a service object s_{ij}. The user can set up preferences (i.e. $w^{n^+}, w^{n^-}, w^{n^T}$) for each attribute *n*, where 0 < $w^{n^+}, w^{n^-}, w^{n^T}$ < 1 and $\sum_{n^-=1}^{N^-} w^{n^-} + \sum_{n^+=1}^{N^+} w^{n^+} + \sum_{n^T=1}^{N^T} w^{n^T} = 1$ hold. Based on this, the utility of a service composition can be computed by aggregating the utility of the selected service objects.

To represent the user's requirements regarding the different aggregated values of a service composition (e.g. the price of the entire process should be less than US $10), we introduce the end-to-end constraints vector $Q_c = [Q_c^1, \dots, Q_c^N]^T$ that includes N constraints values, each for a single attribute n.

3.4 Problem formulation

The problem of selecting the optimal service composition can be considered an optimisation problem where (a) the overall utility of a service composition needs to be maximised while (b) satisfying the constraints Q_c.

4 Novel approach

The basic idea of our approach is to consider the effects of context interdependencies by quantifying and evaluating the values of the context-aware attributes of a service object. This needs to be done for all $s_{ij} \in S_i$ while considering the composition in which it is integrated as well as the context information that arises from a particular composition. To achieve this, we (I) use the concept of states (world and belief states; see Ghallab, Nau, & Traverso, 2004; Heinrich, Bewernik, Henneberger, Krammer, & Lautenbacher, 2008; Heinrich, Bolsinger, & Bewernik, 2009; Heinrich & Schön, 2015) to model and organise context information. We (II) propose an algorithm that determines the world states of a particular composition and thus the context information subject to this composition. Based upon this foundation, we then quantify the values of context-aware attributes for a service object and subsequently determine its utility in dependence of the considered context information. Further, we (III) define a global optimisation model to select the optimal context-aware service composition. Summing up, the contribution of our paper is specified by (I) to (III). Moreover, the proposed optimisation model can be solved by using existing techniques; thus, defining such techniques is not a goal of this paper. For the *purpose of illustration* of our approach and without loss of any generality, we will focus on the following non-context-aware and context-aware attributes:

(1) Non-context-aware attributes nca_{ij}: duration and user favourites[4]

(2) Context-aware attributes ca_{ij}: GPS position, business hours and composition costs.

4.1 Modelling and organising context information

In contrast to the existing approaches, we decided to use the concept of states to model and organise context information. The reason for this is the dynamic characteristic of context information (see Damián-Reyes et al., 2011; Vanrompay et al., 2009; Zhang et al., 2009). The latter leads to the effect that context information varies in dependence of the a priori accomplished service objects (see effect ❶). Hence, to model and organise context information in a well-founded manner, an approach is required that is capable of dealing with these challenges. The concept of states is highly promising to address these challenges as it allows for an efficient consideration of the effects that a service object (or environmental changes) can have on context information. Accordingly, we introduce a belief state BS_i as a set of belief state tuples b_{ik} (with *k* as the number of tuples and *i* as the number of the corresponding service class) with $b_{ik} \coloneqq \left(v(b_{ik}), r(b_{ik})\right)$ where $v(b_{ik})$ is a state variable and restriction $r(b_{ik})$ is a subset of its predefined domain $dom(b_{ik})$. We work with belief states as they allow for a consideration of every possible piece of context information that can arise due to a service composition. In that way, we can assure that the effect ❶ of context interdependencies is entirely taken into

[4] The user favourites are represented by scores with respect to a certain category (e.g. type of restaurant) of a non-context-aware attribute.

account. We further define a world state tuple as $ws_{ik} \subseteq b_{ik}$ where $\forall\ v(b_{ik}) \in b_{ik}\ \ |r(b_{ik})| = 1$ (see Ghallab et al., 2004; Heinrich et al., 2009; Heinrich & Schön, 2015). Thus, for every context information (e.g. G or LO), there is exactly one state variable $v(b_{ik})$ in ws_{ik} with a value that represents this context information. Further, we define the belief state BS_1 as the *initial state* that holds context information at the beginning of the process, and the belief state BS_{I+1} as the *goal state* that holds context information for the end of the process, respectively.

4.2 Determining context information for a service composition

With the concept of states on hand, we now can proceed to determine context information regarding a service composition at planning time. This is necessary, as context information arises from the service composition that is considered. For example, the context information location (given by the GPS position) depends on the a priori accomplished service object of the considered service composition. Consequently, context information is not fixed and given. We can start with the initial state to determine context information for a service composition. As the initial state is the same for any service composition of the considered process, context information can be determined by the use of different sensors, e.g. physical sensors, virtual sensors and logical sensors (see Baldauf, Schahram, & Florian, 2007). Furthermore, the user himself/herself or historical data (e.g. a process is accomplished *n*-times at the same day, time and location) could be used. Starting from this, we can proceed to determine context information for the next service class S_i and thus the next belief state (state-transition).

In this connection, it is important to consider that some context information is only dependent on the last considered service object of the service composition. For example, the context information location is only dependent on the GPS position of the last considered service object. Hence, we will denote that kind of context information and its corresponding state variable in the following as *non-consecutive*. In contrast to that, there is context information that depends not only on the last considered service object s_{ij} but also on the last considered world state ws_{ik}. For example, the context information ‘time of day’ is dependent on the duration of the service object as well as on the time of day of the world state that has been considered before. Thus, we denote that kind of context information and its corresponding state variable in the following as *consecutive*. Considering this differentiation, the algorithm below determines the belief states and thus the context information for every service class S_i. A pseudo code of the algorithm is shown in Table 1.

```
Algorithm 1. CreateWorldStates
Input:   Initial state $BS_1$
Output:  A belief state $BS_i$ for every service class $S_i$
For $i$ = 1 to $I$
    For each $s_{ij} \in S_i$
        For each $ws_{ik} \in BS_i$
            For each $v(b_{ik}) \in ws_{ik}$
              If $v(b_{ik})$ is non-consecutive
                Then $v(b_{i'k}) \leftarrow ca_{ij}^{n} \vee nca_{ij}^{n}$
                Else $v(b_{i'k}) \leftarrow \otimes\left(ca_{ij}^{n}, v(b_{ik})\right) \vee v(b_{i'k}) \leftarrow \otimes\left(nca_{ij}^{n}, v(b_{ik})\right)$ [5]
              End If
            End For
            If $ws_{i'k}$ already exists in $BS_{i'}$
              Then delete $ws_{i'k}$
              Else add $ws_{i'k}\ with\ v(b_{i'k})$ to $BS_{i'}$
            End If
        End For
    End For
End For
```

Table 1 State transition algorithm

[5] Different state transition functions (denoted by the symbol ⊗) are applied in dependence on the type of state variable and context information, respectively.

Beginning with the initial state BS_1 and the first service class S_1, the algorithm determines every next belief state $BS_{i'}$ with the corresponding world states $ws_{i'k}$ for the successor class $S_{i'}$ and subsequently for all S_i. This is achieved by imitating the selection of a service object s_{ij}. The algorithm differentiates between non-consecutive and consecutive state variables. For non-consecutive state variables it determines the value of $v(b_{i'k})$ by using only the value ca_{ij}^n or nca_{ij}^n of the considered service object s_{ij} (see line 6 in Table 1). In contrast to that, the value of a consecutive state variable is determined by taking into account the value of ca_{ij}^n or nca_{ij}^n in combination with the value of the state variable $v(b_{ik})$ in ws_{ik} (cf. line 6). Finally, the algorithm evaluates whether the determined world state $ws_{i'k}$, with its values of the state variables $v(b_{i'k})$, already exists in the belief state $BS_{i'}$. If this is the case, the determined world state $ws_{i'k}$ remains unconsidered; otherwise, it is stored in $BS_{i'}$.

The result of the algorithm is a belief state BS_i for every service class S_i that holds all feasible values for every state variable $v(b_{ik}) \in ws_{ik}$. In that way, we are able to address effect ❶.

For a better understanding, the algorithm is illustrated with the help of two small examples (see Figure 3 and Figure 4). We start by concentrating only on *non-consecutive* state variables (see Figure 3) which are represented by the context information location. The initial state is denoted by the belief state BS_1 which comprises two different feasible world states ws_{11} and ws_{12} (representing two different locations). In other words, the process can be started from one of the two different locations. As only non-consecutive state variables are considered, the values of the state variables in $ws_{21}, ws_{22} \in BS_2$ are determined by using the GPS positions of s_{11} and s_{12}, respectively. With the help of the world states in BS_2, the quantified values of the context-aware attributes (see $p_{21}(ws_{21})$ and $p_{21}(ws_{22})$) for s_{21} can then be determined.

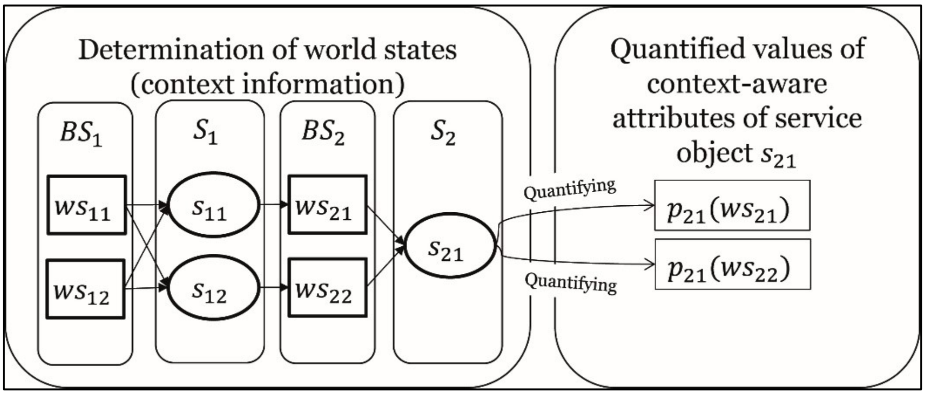


Figure 3 Determining the value of non-consecutive state variables

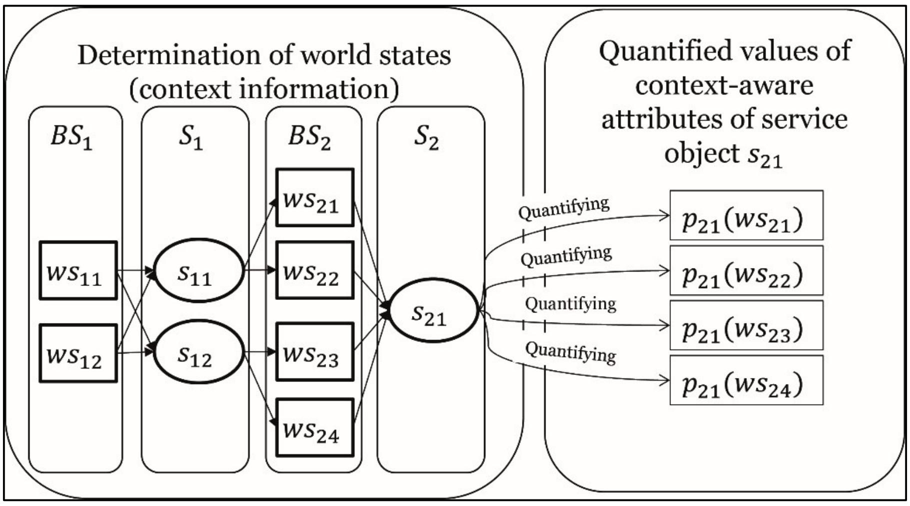


Figure 4 Determining the value of consecutive state variables

We proceed by concentrating only on *consecutive* state variables (see Figure 4), which are represented by the context information time of day. Similar to Figure 3, belief state BS_1 is the initial state. Again, BS_1 comprises the two different world states ws_{11} and ws_{12}, with the corresponding context information time of day being different in $v(b_{11})$ and in $v(b_{12})$. In contrast to the context information location, the time of day is depending on the last considered service object s_{ij} as well as on the world state ws_{ik}. This means that for the determination of belief state BS_2, both the time of day in $v(b_{11})$ and $v(b_{12})$ and the duration of the service objects s_{11} and s_{12} have to be considered. Taking this into account, four new world states can be determined for belief state BS_2. For example, by aggregating the value of $v(b_{11})$ with the duration of the service object

s_{11} the world state ws_{21} is created. Based upon the set of world states in BS_2 the quantified values of the context-aware attributes (see $p_{21}(ws_{21}), p_{21}(ws_{22}), p_{21}(ws_{23}), p_{21}(ws_{24})$) for s_{21} can then be determined.

Using the concept of belief and world states, it is necessary to discuss the problem of state explosion. The consideration of *consecutive* state variables in the above example (see Figure 4) causes the number of world states K_1 (e.g. two) in belief state BS_1 compared to the number of world states K_2 (e.g. four) in state belief BS_2 to increase. In detail, the number of world states increases with each additional service class by the factor $K * J_I$ (J_I is the total number of service objects available in service class *I*). As a result, we are confronted with a problem of a probably rapid growth of world states as the process size increases. In many cases, due to the described problem, it is hardly possible to determine all world states for the process in an efficient and scalable way. A main reason for the problem of state explosion can be found in the non-restricted codomain of those attributes (see line 7 in Table 1) that are used to determine the value of consecutive state variables. Resulting from this, a (huge) number of world states are determined which, however, may not differ much in their values of the state variables. At planning time (in contrast to runtime), working on this level of detail (e.g. context information in world state ws_{ik} is 10:00 am, and 10:01 am in world state $ws_{i'k}$) may be neither practical nor necessary. Hence, to address the problem of state explosion, we work with discrete codomains for those attributes that are used to determine the value of consecutive state variables. In that way, we deal with a kind of belief states instead of single world states. Hence, we are able to significantly reduce the number of world states (see the evaluation section) determined for the process which is already sufficient at planning time.

4.3 Quantifying the values of context-aware attributes

Having determined context information for the process, we now proceed to determine the *quantified* values of context-aware attributes for every service object $s_{ij} \in S_i$ with the help of the quantification functions τ^{γ}, context information ($ws_{ik} \in BS_i$) and the values of ca_{ij}. It should be noted that based on the world state $ws_{ik} \in BS_i$, one *unique quantified* NFP vector $p_{ij}(ws_{ik}) = \left[p_{ij}^1, \dots, p_{ij}^N\right]^T$ for a service object $s_{ij} \in S_i$ can be determined. Hence, subject to the considered world state ws_{ik} the vector $p_{ij}(ws_{ik})$ is usually a different one (see Figures 3 and 4). As the utility of a service object is determined based upon the vector p_{ij} (see term 7), context interdependencies (see effect ❷) that exist for a service composition are already considered in the utility value of a service object. Context-aware attributes where the value is quantified by means of the Boolean type function represent a special case during quantification. Here, the quantified value (e.g. ‘1’ and ‘0’) allows for a direct verification of the service object’s feasibility. For instance, if the value of the context-aware attribute business hours is just quantified to ‘0’ (e.g. a restaurant is closed at the determined time of day), it is obvious that this particular service object cannot be part of a feasible service composition. Thus, in this case (i.e. for the service composition at hand), this particular service object remains directly unconsidered. In that way, the number of decision variables and constraints and thus the complexity in the optimisation model (see y_{ik} in term 9) can be reduced.

4.4 Optimisation model

With the quantified values of context-aware attributes, we now can proceed with the optimisation model to determine the optimal service composition. Based upon the notation and the problem statement introduced in the model setup, our optimisation model is defined as follows:

$$\arg\max_{sc_l} \sum_{S_i \in sc_l} \sum_{s_{ij} \in S_i} \sum_{ws_{ik} \in BS_i} U\left(s_{ij}, p_{ij}(ws_{ik})\right) * x_{ij} * y_{ik} \tag{9}$$

$$S.t.: \sum_{s_{ij} \in S_i} x_{ij} = 1 \ with\, x_{ij} \ \in \{0,1\} \ \forall \ S_i \in sc_l \tag{10}$$

$$\sum_{ws_{ik} \in BS_i} y_{ik} = 1 \ with\, y_{ik} \ \in \{0,1\} \ \forall \ BS_i \in sc_l \tag{11}$$

$$\sum_{ws_{ik} \in BS_i} y_{ik} * \sum_{(s_{(i-1)j}, ws_{(i-1)k^*}) \in \Upsilon_{\mathrm{ik}}} \left(x_{(i-1)j} * y_{(i-1)k^*}\right) = 1 \ with\, i \in [2; I] \qquad (12)^6$$

$$\sum_{S_i \in sc_l} \sum_{s_{ij} \in S_i} p_{ij}^{duration} * x_{ij} \leq \ Q_c^{duration} \tag{13}$$

$$\sum_{S_i \in sc_l} \sum_{s_{ij} \in S_i} p_{ij}^{favorites} * x_{ij} \geq \ Q_c^{favorites} \tag{14}$$

$$\sum_{S_i \in sc_l} \sum_{s_{ij} \in S_i} \sum_{ws_{ik} \in BS_i} p_{ij}^{comp.costs} * x_{ij} * y_{ik} \leq \ Q_c^{comp.costs} \tag{15}$$

$$\sum_{S_i \in sc_l} \sum_{s_{ij} \in S_i} \sum_{ws_{ik} \in BS_i} p_{ij}^{distance} * x_{ij} * y_{ik} \leq \ Q_c^{distance} \tag{16}$$

Term (9) shows that the service composition with the highest accumulated utility is determined as the optimal one, while the utility of a service object s_{ij} depends on the quantified NFP vector p_{ij} and on the considered world state ws_{ik}. Term (10) assures that exactly one service object s_{ij} for each service class S_i is selected ($x_{ij} = 1$ indicates that service object s_{ij} has been selected to accomplish the service composition; $x_{ij} = 0$

[6] Please note, that this constraint is mandatory as soon as *consecutive* context information is considered. In case that only *non-consecutive* context information is taken into account the following constraint is necessary: $\sum_{ws_{ik} \in BS_i} y_{ik} * \sum_{s_{(i-1)j} \in \Theta_{ik}} x_{(i-1)j} = 1 \ with\, i \in [2; I]$ where Θ_{ik} is a set of service objects which are involved in the creation of ws_{ik}.

if not). Term (11) assures that exactly one world state ws_{ik} for each belief state BS_i is selected ($y_{ik} = 1$ indicates that world state ws_{ik} has been considered for the quantification of the context-aware attributes of service object s_{ij}; $y_{ik} = 0$ if not). Further, term (12) assures that only world states are considered that are feasible with respect to the considered combination of world state and service object in the preceding service class. Here, Υ_{ik} is a set of all such combinations (i.e. $(s_{(i-1)j}, ws_{(i-1)k})$) involved in the creation of the world state ws_{ik}. Q_c is the constraints vector $Q_c = [Q_c^1, \dots, Q_c^N]^T$ that is used to represent the user constraints regarding the different aggregated end-to-end values of the attributes. These constraints are defined either as lower bounds or upper bounds[7]. Hence, term (13) ensures that the aggregated duration is less than or equal to the constraints of the user. Term (14) is to guarantee that the aggregated scores of the favourites are greater than or equal to the constraints of the user (e.g. each service object needs to have at least a score of 3 based on a 5 point rating scale). It should be noted that, as the duration and favourites are non-context-aware attributes, the aggregated value only depends on the selected service object s_{ij}. Hence, for the determination of the aggregated values, only x_{ij} is relevant. In contrast to that, the distance and composition costs are context-aware attributes. Hence, their quantified values, and thus the aggregated values, are dependent on both the considered service object s_{ij} and world state ws_{ik}. Consequently, for the determination of the aggregated values, x_{ij} and y_{ik} are relevant. In that way, we are able to address effect ❸. Terms (15) and (16) ensure that the aggregated distance and composition costs are less than or equal to the constraints of the user. Please note that as the feasibility of service objects

[7] Please note that the optimization model can easily be extended by constraints that are defined as equality conditions if required.

regarding the context-aware attribute business hours (i.e. in general all context-aware attributes where the value is quantified by means of the Boolean type function) is already verified during quantification, no further constraint is necessary here.

4.5 Determination of the optimal service composition

For the resolution of the optimisation model and subsequently to determine the optimal service composition, several analytical as well as heuristic approaches can be used. The choice of the approach applied depends mainly on the question of whether an optimal solution is needed or not. For the determination of an exact solution, either the approach of integer programming (see Gao et al., 2006) or the algorithm MCSP (see Yu, Zhang, & Lin, 2007) can be used. In contrast to that, heuristic approaches such as genetic algorithms (see Ai & Tang, 2008a; Yuan et al., 2013) or ant colony algorithms (see Wu & Zhu, 2013) may have good scalability in terms of the process size, but cannot guarantee that the optimal solution is found.

5 Evaluation

In this section, we evaluate the feasibility of the proposed approach to contribute to challenges ❶-❸. We therefore validate in a first step (i) the feasibility of the state transition algorithm as well as of the optimisation model. Then, (ii) we demonstrate the importance of considering the effects of context interdependencies. After that, (iii) we evaluate the performance of the so-called quantifying step (i.e. state transition and determination of the quantified values of context-aware attributes). Finally, (iv) we demonstrate the practical applicability of our approach by means of a real-world example.

5.1 (i) Feasibility of the state transition algorithm as well as the optimisation model

In addition to the manual analysis of the source code (structured walk through) by persons other than the programmers, we made a series of tests using the JUnit framework, including runs with extreme values, regression tests, unit tests and integration tests, thus checking number of created world states in detail for different specifications of both non-context-aware and context-aware attributes. The implemented algorithms did not show any defects at the end of the test phase. Further, in order to assure that our optimisation model determines the optimal solution, we compared the results of the Integer-Programming solver *Gurobi* with the results of an exhaustive enumeration with respect to different specifications, including different process sizes as well as different numbers of context-aware and non-context-aware attributes. Please note, to use Gurobi, the optimisation model given in terms (8-16) has to be linearised first. With this analysis, we are able to ensure that our optimisation model works properly, as the results showed no difference between our model and the exhaustive enumeration regarding the determined solution.

5.2 Evaluation methodology for steps (ii) and (iii)

To evaluate steps (ii) and (iii), we used three different scenarios reflecting three different process sizes. The first scenario is used to illustrate the effects of context interdependencies *(step ii)* by comparing the results (optimal service composition) of our approach with a local selection approach (e.g. Kirsch-Pinhero et al., 2008) and the BCCbSS approach (see Yu & Reiff-Marganiec, 2009b) which is capable of considering context interdependencies partially.

- In the first scenario, the number of service objects J_i per service class is 5 and the number of service classes I is increased in steps of 5 from 5 to 100. The context-aware-attribute that is taken into account for this scenario is the GPS position (i.e. the quantified value is distance). We used randomly generated GPS positions within in a radius of 10 km to represent a metropolitan area.

For the first scenario we used *Gurobi* to determine our optimal solution.

The second and third scenarios are used to evaluate the quantifying step *(iii)* with respect to its feasibility and performance, and the number of created states.

- In the second scenario, the number of service classes I is 10 and the number of service objects J_i per service classes is increased in steps of 50 from 50 to 500.
- In the third scenario, the number of service objects J_i per service class is 80 and the number of service classes I is increased in steps of 2 from 2 to 50.

For both scenarios, we start by taking only *non-consecutive context information* (represented by the context-aware attributes GPS position and composition costs) into account. We then add *consecutive context information* (represented by the context-aware attribute business hours). All the values for the (non-)context-aware attributes were generated randomly except the values of the duration. Here, we used the discrete codomain of (30, 60, 90 and 180) in order to deal with the problem of state explosion.

We simulated each of the three scenarios 50 times. For the first scenario – with a focus on the context information location (distance) – we determined the additional utility when considering the effects of context interdependencies. To be precise, we compared the results from our approach with those from a local selection approach and the BCCbSS approach. These additional utilities ΔU_1 and ΔU_2 are defined as:

$$\Delta U_1 = 1 - \left(\frac{\sum_{s_i \in sc_l} \sum_{s_{ij} \in s_i} p_{ij}^{distance} (our\ approach)}{\sum_{s_i \in sc_l} \sum_{s_{ij} \in s_i} p_{ij}^{distance} (localselection)} \right) \qquad (17)$$

$$\Delta U_2 = 1 - \left(\frac{\sum_{s_i \in sc_l} \sum_{s_{ij} \in s_i} p_{ij}^{distance} (our\ approach)}{\sum_{s_i \in sc_l} \sum_{s_{ij} \in s_i} p_{ij}^{distance} (BCCbSS)} \right) \qquad (18)$$

For scenarios two and three, we determined the average computation time as well as the absolute number of created world states.

All of the analyses were conducted on a machine with an Intel Core I7 processor with 3.6 GHz, 16 GB RAM, Gurobi 6.0 and Java 1.8.

5.3 (ii) The importance to consider the effects of context interdependencies

To show the effects of context interdependencies, we plotted the two graphs in Figure 5, illustrating the additional average utility achieved (see ΔU_1 and ΔU_2). The first finding is that considering the effects of context interdependencies has a significant positive impact on the utility. The results show that, on average, ΔU_1 equals 20% and ΔU_2 equals 14%. Even for a small number of service classes, our approach was able to achieve superior results compared to the local selection and the BCCsBB approach. In some cases, the additional utility is even over 60% (e.g. $\Delta U_1 = 65\%$ and $\Delta U_2 = 63\%$). This is not surprising, as with the approach presented here context interdependencies are considered through the entire process, while in existing approaches context interdependencies are either only considered between two successive service classes (see BCCsBB approach) or left completely unconsidered (see local selection approach). The scalability of our approach with respect to the process size is, in comparison with the scalability of the local selection and the BCCbSS approach, slightly worse. However, even for larger processes (e.g. 100 service classes) our approach took on average just 0.87 sec to determine the solution, which is more than acceptable *at planning time* (e.g. 1 day or one hour before accomplishing the process). In situations where the latter may not be the case, heuristics (see Ai & Tang, 2008a) for the resolution of the optimisation problem can be applied (see above). Indeed, this can be easily realised, since, compared to a situation in which an exact solution is not required, no adaptations have to be made to the state transition and the optimisation model.

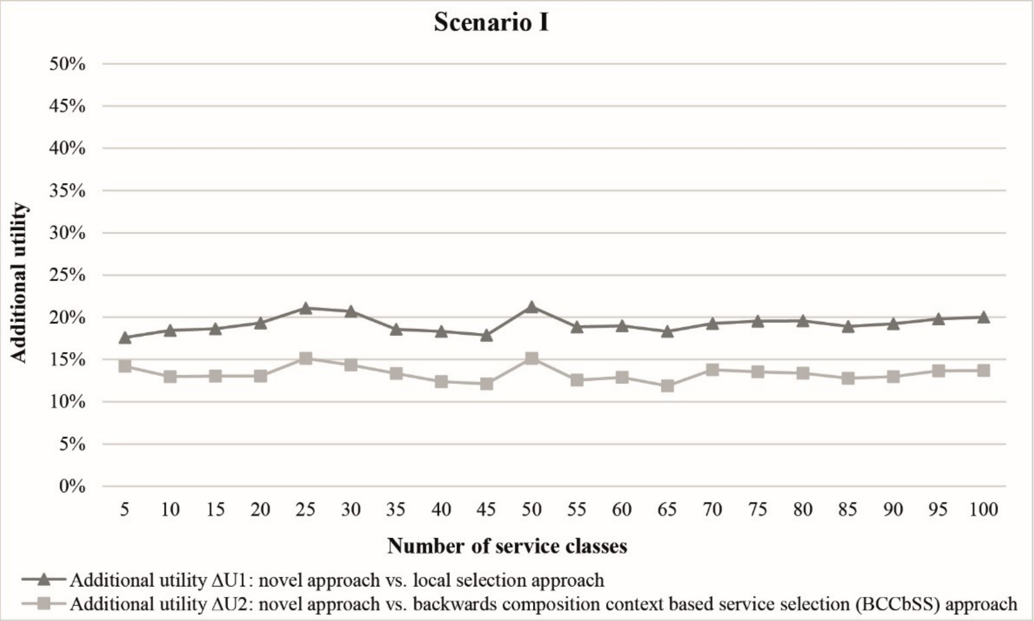


Figure 5 Additional utility of the novel approach

5.4 (iii) Performance of the quantifying step

Figures 6 to 9 illustrate the number of created world states and the computation time of the quantifying step of our approach for scenarios two and three.

We start our discussion with the effects of the usage of *non-consecutive context information* on the *number of created world states*. Provided that only non-consecutive context information is considered, exactly one new world state per service object is being created (see Figure 3). As a result, we expect a linear growth of world states as the process size increases. Indeed, the results of our simulation experiment reveal this growth. The number of world states increases in a linear fashion as either the number of service classes (see Figure 6 – dark grey line) or the number of service objects per service class (see Figure 7 – dark grey line) is increased.

Next, we discuss the effects on the *number of created world states* if *consecutive context information* is additionally considered. The results (see Figure 6 – light grey line) of our simulation experiment show that the number of world states increases in a linear fashion if the number of service objects per service class is increased. This is not surprising, as the main driver for the number of world states is the number of service classes (see Figure 4) in cases where consecutive context information is considered. As this number is fixed (i.e. *I* = 10), the total number of world states increases by a constant factor in the case that the number of service objects is increased. In contrast to that is the growth of world states in case the number of service classes is increased. Here, the results (see Figure 7 – light grey line) show that the number of created world states increases in an over-proportional manner. However, with the usage of discrete state variables (i.e. using a kind of belief states), we are able to slow down this growth compared to a situation in which we would work with single world states and state variables with possibly infinite sets of values (continuous domain). Hence, the results provide some support that our idea to handle the problem of state explosion works.

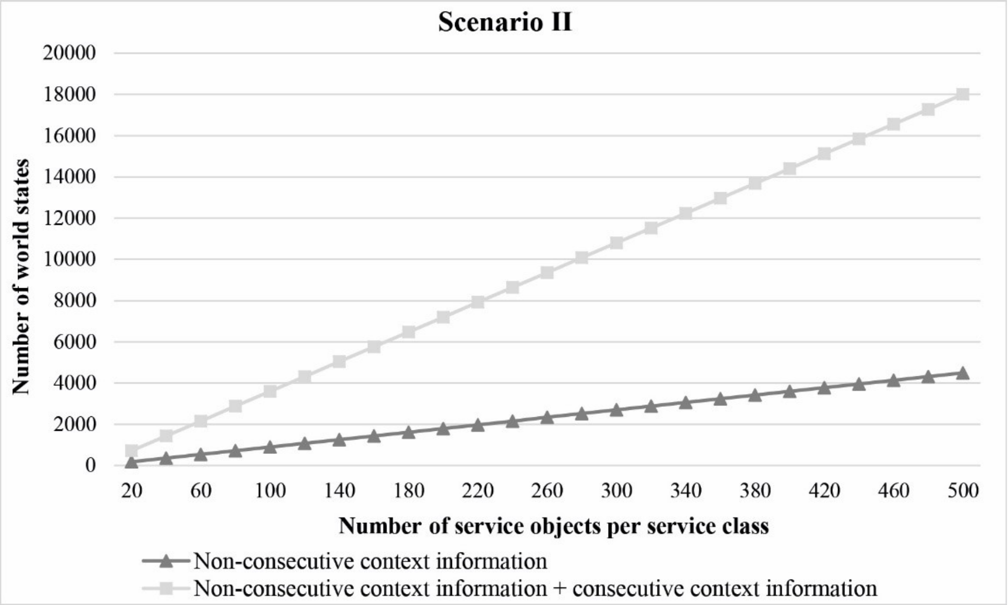


Figure 6 Number of created world states vs. number of service objects per service class

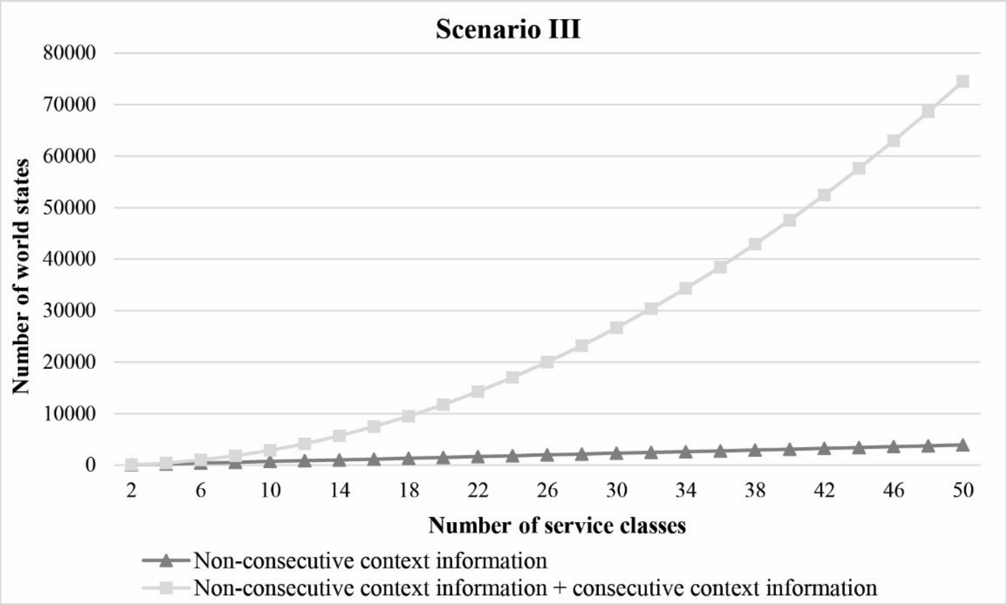


Figure 7 Number of created world states vs. number of service classes

We now take a closer look at the computation time of our state creation algorithm needed to determine all world states and subsequently to quantify the values of context-aware attributes. We start with a discussion on the *computation time* dependent on of the usage of *non-consecutive context information.* Figures 8 and 9 (dark grey line) show that the computation time increases very slowly, as the process size is increased. This may be due to the fact that the number of world states grows in a linear fashion as the process size is increased, in the case that only non-consecutive context information is considered. Even for large processes (i.e. 10 service classes and 500 service objects per service class), the state creation algorithm on average only took 15 sec to determine all world states and subsequently to quantify the values of non-context-aware attributes.

Contrary to the sole usage of non-consecutive context information, the results (see Figures 8 and 9 - light grey line) show that the computation time increases in an over-proportional fashion as the process size is increased when *consecutive context information* is additionally considered. This may account for the fact that, with the usage of consecutive context information, the state creation algorithm needs to take both the service object and the last considered world state (see line 7 in Table 1) into account to create a new world state. However, with an average computation time of 160 sec for large process sizes (i.e. 10 service classes and 500 service objects per service class), the observed computation times are more than manageable from a planning point of view.

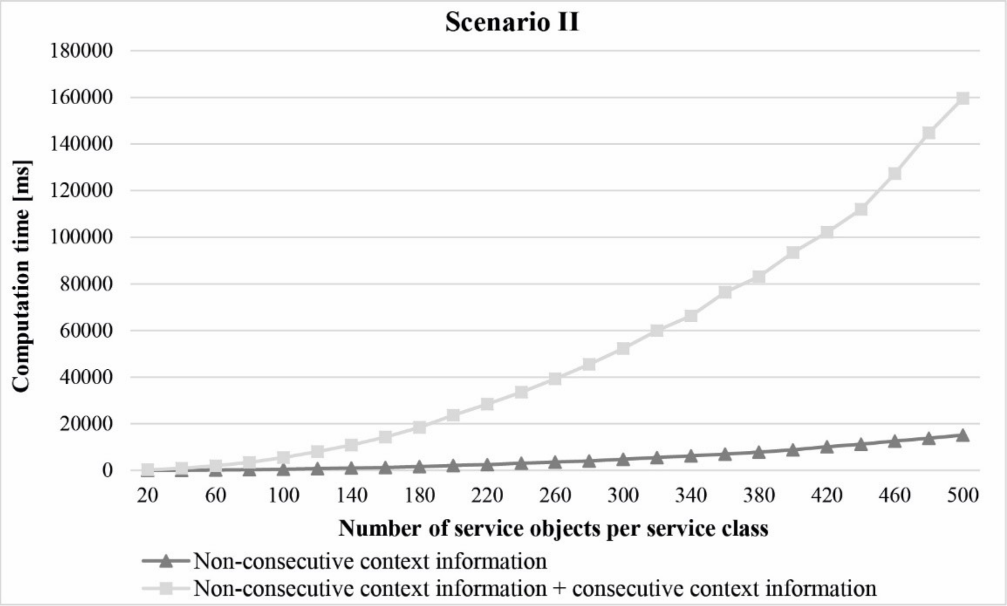


Figure 8 Computation time vs. number of service objects per service class

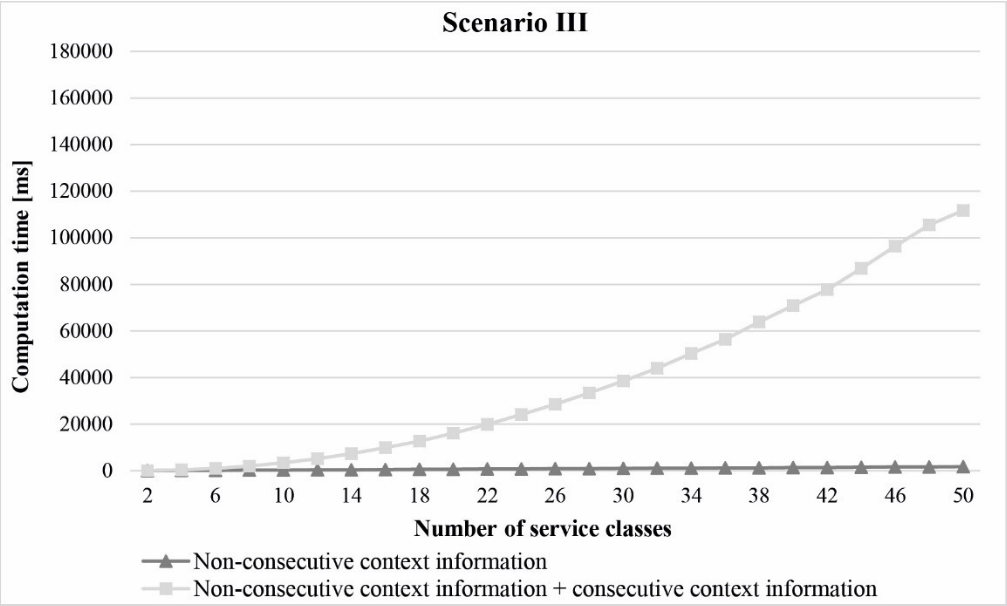
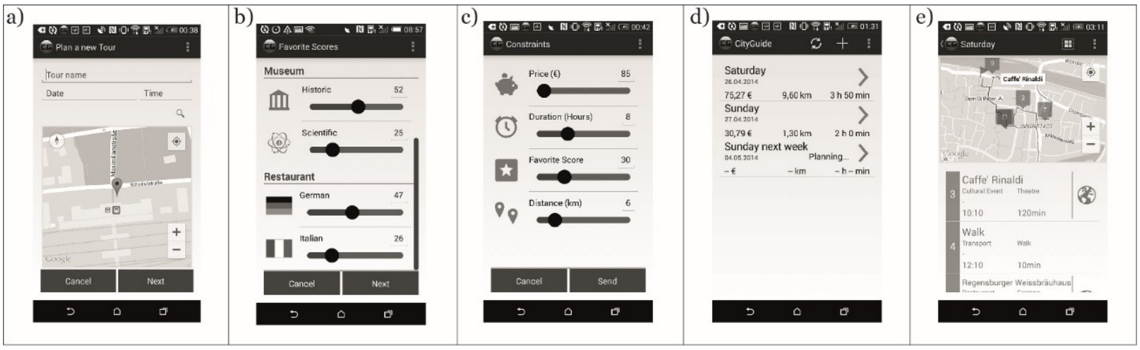


Figure 9 Computation time vs. number of service classes

5.5 (iv) Practical applicability

In this step, we want to demonstrate the practical applicability of our approach by means of a real-world example. We developed a prototypical Android app that allows the user to plan a day trip in a city. For this purpose, we chose the World Heritage City of Regensburg, Bavaria, Germany. The day trip is modelled as a process having 15 different activities (= service classes) such as, for instance: visiting a museum or a sight, transportation, having lunch, having dinner, visiting a café and visiting a bar or a night club, etc. In contrast to steps *(ii)* and *(iii)*, we used the information services Google Maps, Yelp and TripAdvisor to determine for each activity suitable real world service objects as well as their non-functional properties (e.g. location, business hours, food category, etc.). Moreover, with the help of the app, the mandatory input for the selection such as the end-to-end constraints of the user as well as the initial state that holds context information for the beginning of the process can be determined. As Figure 10 (b and c) illustrates, users can set up the end-to-end constraints for the different non-functional properties and also provide information about their favourites (e.g. Italian food, historical museums, etc.). To determine the initial state (e.g. the values of the GPS position and time of day), either the sensors of the mobile device can be used or the user himself/herself can set up the necessary values (see Figure 10 a). The latter, for instance, can be applied if the trip is going to start the following day, etc. While the selection process is ongoing the user is informed about the current status of selection (see Figure 10 d). Once the selection process is finished, the results of the optimisation are displayed (see Figure 10 e) by means of a map.



Tour name
Date Time

Museum
Historic 52
Scientific 25
Restaurant
German 47
Italian 26

Price (€) 85
Duration (Hours) 8
Favorite Score 30
Distance (km) 6

Saturday
26.04.2014
75,27 € 9,60 km 3 h 50 min
Sunday
27.04.2014
30,79 € 1,30 km 2 h 0 min
Sunday next week
04.05.2014 Planning...
– € – km – h – min

3 Caffe' Rinaldi
Cultural Event Theatre
10:10 120min
4 Walk
Transport Walk
12:10 10min
Regensburger Weissbräuhaus

Figure 10 Prototypical implementation of the City Trip App

By tapping on a service object (e.g. Caffé Rinaldi) the user is provided with additional information (e.g. the web site of a restaurant). Compared to current information services (i.e. also commercial apps like Google Maps, Yelp), the app using our approach has several advantages. First, it has the capability to select service objects not just for a single activity, but rather for an entire process where the effects of context interdependencies are considered. Furthermore, the user is able to set up end-to-end constraints that must be considered for the entire process, which is important from a practical point of view. Finally, it allows for well-founded decision support in order to deal with the problem of information overload.

6 Conclusion, limitations and future research

In this paper, we address the context-aware service selection that aims to select the optimal service composition based upon context information. For this selection problem, several approaches can be found in the literature. However, the effects ❶-❸ of context interdependencies have either not been considered or only been considered partially by existing selection approaches so far. Here, our approach aims to contribute to resolving these challenges. It is highly relevant in cases where multiple service objects are interdependent and several pieces of context-information must be considered.

To design our approach, we used the concept of states to model and organise context information. Based upon the latter, we determined the quantified values of context-aware attributes of a service object, and, subsequently its utility. Based on this, the utility of a service composition can be calculated. Here, we find that the solutions determined by our approach, that jointly takes the effects of context interdependencies into account, significantly outperforms current selection approaches with respect to the utility of the selected service composition. Additionally, we illustrate the strength and practical benefits of the presented approach by means of a real world-example.

Our findings offer some important practical implications. The extent of the effects of context interdependencies can be significant, as our results show. In some cases, the difference between the utility resulting from our approach and the one resulting from existing selection approaches was over 60%. Moreover, using our approach (e.g. with a mobile app) allows for well-founded decision support of the user. On the one hand, this is achieved by considering an entire process instead of just a single activity, where additionally the user's end-to-end constraints are taken into account. On the other hand, our approach selects and subsequently displays the optimal service composition rather than a bunch of service objects. In that way, it helps to deal with the problem of information overload. Given that impact, we believe that users and practitioners would substantially benefit from using our approach when selecting service objects for a process based on context information.

Moreover, we have to discuss some limitations which are the starting point of further research: first, our approach works with data from existing information services. In order to prevent the user from being frustrated due to wrong or missing information, high data quality (accuracy, currency, completeness, etc.) is required, which may not always be the case. Hence, collected data must be checked with respect to their quality.

This can be accomplished, for instance, by using existing quality approaches (see Kim & Lee, 2006; Manzoor, Truong, & Dustdar, 2014). Second, for large processes, especially with a very high number of service objects as well as a great deal of consecutive context information that needs to be considered, our approach suffers from a higher computation time. Here, a future research goal is to show how existing heuristics (see Alrifai et al., 2012; Wu & Zhu, 2013) can be used for the quantifying step and the subsequent selection step. Additionally, we intend to consider of processes with different control flow patterns. Furthermore, there might be activities (e.g. having a meal) in a process that occur more than once. For such activities, it is necessary to consider that an already selected service object (e.g. regarding a certain restaurant) may have a lower utility for the user when selected the second time (e.g. normally a user does not want to visit the same restaurant for lunch and dinner). In future work, we will consider these situations. Finally, a further research goal is how endogenous context information (e.g. the weather) can be considered, and how the optimal service composition can be adapted to this context information already at planning time (e.g. using probabilities regarding different weather conditions).

Our approach presents an appropriate foundation for this and for the aforementioned enhancements, thus serving as a suitable basis for further research.

Funding

This research was funded by the Austrian Science Fund (FWF) [P 23,546-G11]

7 References

Aart, C. V., Wielinga, B., & Hage, W. R. V. (2010). Mobile Cultural Heritage Guide: Location-Aware Semantic Search. *Knowledge Engineering and Management by the Masses, 6317,* 257–271.

Ai, L., & Tang, M. (2008a). A Penalty Based Genetic Algorithm for QoS-aware Web Service Composition with Inter-Service Dependencies and Conflicts. In M. Mohammadian (Ed.), *Proceedings of the 2008, IEEE International Conference on Computational Intelligence for Modelling, Control and Automation* (pp. 738–743). IEEE. Vienna, Austria.

Ai, L., & Tang, M. (2008b). QoS-based Web Service Composition Accommodating Inter-Service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm. In R. v. Engelen, M. Govindaraju, & M. Cafaro (Eds.), *Proceedings of the 2008, International Conference on eScience* (pp. 119–126). IEEE. Indianapolis, Indiana.

Alrifai, M., Risse, T., & Nejdl, W. (2012). A Hybrid Approach for Efficient Web Service Composition with End-to-End QoS Constraints. *ACM Transactions on the Web, 6*(2), 1–31.

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

Badidi, E., & Larbi, E. (2010). A Quality of Context Driven Approach for the Selection of Context Services. In IEEE Computer Society (Ed.), *Proceedings of the 2010, IEEE International Conference on e-Business* (pp. 1–6). IEEE. Athens, Greece.

Baldauf, M., Schahram, D., & Florian, R. (2007). A Survey on Context-Aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing, 2*(4), 263–277.

Bertoli, P., Cimatti, A., Roveri, M., & Traverso, P. (2006). Strong Planning under Partial Observability. *Artificial Intelligence, 170*(4-5), 337–384.

Chen, G., & Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. *Technical Report TR2000-381, 1*(2), 1–16.

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

Dannewitz, C., Pentikousis, K., Rembarz, R., Renault, É., Strandberg, O., & Ubillos, J. (2008). Scenarios and Research Issues for a Network of Information. In European Alliance for Innovation (Ed.), *Proceedings of the 2008, International Mobile Multimedia Communications Conference* (pp. 1–7). ICST. Oulu, Finnland.

Daurer, S., Molitor, D., Spann, M., & Manchanda, P. (2013). The Impact of Smartphones, Barcode Scanning, and Location-Based Services on Consumers' Search Behavior. In R. Baskerville & M. Chau (Eds.), *Proceeding of the 2013, International Conference on Information Systems*. AIS. Milan, Italy.

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

Dhar, S., & Varshney, U. (2011). Challenges and Business Models for Mobile Location-Based Services and Advertising. *Communications of ACM, 54*(5), 121–128.

Fujii, K., & Suda, T. (2009). Semantics-Based Context-Aware Dynamic Service Composition. *ACM Transactions on Autonomous and Adaptive Systems, 4*(2), 1–31.

Gao, A., Yang, D., Tang, S., & Zhang, M. (2006). QoS-Driven Web Service Composition with Inter Service Conflicts. In X. Zhou, J. Li, H. T. Shen, M. Kitsuregeawa, & Y. Zhang (Eds.), *Proceedings of the 2006, International Conference on Frontiers of WWW Research and Development* (pp. 121–132). IEEE. Harbin, China.

Gao, Y., Zhang, B., Na, J., Yang, L., Dai, Y., & Gong, Q. (2005). Optimal Selection of Web Services for Composition Based on Interface-Matching and Weighted Multistage Graph. In H. Shen & K. Nakano (Eds.), *Proceedings of the 2005, IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies* (pp. 336–338). IEEE. Dalian, China.

Gerpott, T. J., & Berg, S. (2011). Determinants of the Willingness to Use Mobile Location-Based Services. *Business & Information Systems Engineering, 3*(5), 279–287.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. San Francisco: Elsevier Inc.

Haddad, J. E., Manouvrier, M., Reiff-Marganiec, S., & Rukoz, M. (2010). Context-Based Transactional Service Selection Approach for Service Composition. *Notes de Recherche No 49. LAMSADE - CNRS FRE 3234 - Universite Paris Dauphine.*,

Heath, T. P., Fassler, M. J., Mubayi, P., Balakhnin, A., Schwartz, D., Hu, W., . . . Grambling, S. (2014). *eCommerce Expected to Accelerate Globally in 2014*. Retrieved from http://www.wisburg.com/wp-content/uploads/2014/03/%E9%AB%98%E7%9B%9B-eCommerce-expected-to-accelerate-globally-in-2014.pdf

Heinrich, B., Bewernik, M.-A., Henneberger, M., Krammer, A., & Lautenbacher, F. (2008). SEMPA - A Semantic Business Process Management Approach for the Planning of Process Models. *WIRTSCHAFTSINFORMATIK, 50*(6), 445–460.

Heinrich, B., Bolsinger, M., & Bewernik, M. (2009). Automated Planning of Process Models: The Constructions of Exclusive Choices. In H. Chen & S. Slaughter (Eds.), *Proceedings of the 2009, International Conference on Information Systems*. AIS. Phoenix, Arizona.

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

Heinrich, B., & Schön, D. (2015). Automated Planning of Context-Aware Process Models. In AIS (Ed.), *Proceedings of the 2015, European Conference on Information Systems*. AIS. Munster, Germany.

Hinkelmann, K., Maise, M., & Thönssen, B. (2013). Connecting Enterprise Architecture and Information Objects Using an Enterprise Ontology. In A. Gerber & P. v. Deventer (Eds.), *Proceedings of the 2013, International Conference on Enterprise Systems* (pp. 1–11). IEEE. Cape Town, South Africa.

Hwang, C.-L., & Yoon, K. (1981). Methods for Multiple Attribute Decision Making. In C.-L. Hwang & K. Yoon (Eds.), *Lecture Notes in Economics and Mathematical Systems. Multiple Attribute Decision Making - Methods and Applications A State-of-the Art Survey* (pp. 58-191). Springer Berlin Heidelberg.

Kim, Y., & Lee, K. (2006). A Quality Measurement Method of Context Information in Ubiquitous Environments. In G. Lee, D. Slezak, T.-h. Kim, P. Sloot, H.-k. Kim, S. I. Ko, & M. Szczuka (Eds.), *Proceedings of the 2006, International Conference on Hybrid Information Technology* (pp. 576–581). IEEE. Cheju Island, Korea.

Kirsch-Pinhero, M., Yves, V., & Yolande, B. (2008). Context-Aware Service Selection Using Graph Matching. In F. De Paoli, I. Toma, A. Maurino, M. Tilly, & G. Dobson (Eds.), *Proceedings of the 2008, Workshop on Non Functional Properties and Service Level Agreements in Service Oriented Computing*. CEUR-WS.org. Dublin, Ireland.

Leberknight, C. S. (2010). A Location Aware Service to Minimize Travel Costs Using Dynamic Information. In D. Leidner & J. Elam (Eds.), *Proceedings of the 2010, Americas Conference on Information Systems*. AIS. Lima, Peru.

Legner, C. (2009). Do Web Services Foster Specialization? An Analysis of Commercial Web Services Directories. *Proceedings of the 2009, International Conference on Business Informatics, 1,* 67–76.

Lin, D., & Ishida, T. (2012). Dynamic Service Selection Based on Context-Aware QoS. In L. Moser, M. Parashar, & P. Hung (Eds.), *Proceedings of the 2012, IEEE International Conference on Services Computing* (pp. 641–648). IEEE. Honolulu, Hawaii.

Luo, X., Gu, B., Fang, Z., & Xu, Y. (2013). Quantifying the Dynamic Sales Impact of Location-Based Mobile Promotion Technologies. In R. Baskerville & M. Chau (Eds.), *Proceeding of the 2013, International Conference on Information Systems*. AIS. Milan, Italy.

Manzoor, A., Truong, H.-L., & Dustdar, S. (2014). Quality of Context: Models and Applications for Context-Aware Systems in Pervasive Environments. *The Knowledge Engineering Review, 28*(02), 154–170.

Okazaki, S., & Mendenz, F. (2013). Perceived Ubiquity in Mobile Services. *Journal of Interactive Marketing, 27,* 98–111.

O'Sullivan, J., Edmond, D., & Hofstede, A. T. (2002). What's in a Service? Towards Accurate Description of Non-Functional Service Properties. *Distributed and Parallel Databases, 12*(2-3), 117–133.

Ramacher, R., & Mönch, L. (2012). Cost-Minimizing Service Selection in the Presence of End-To-End QoS Constraints and Complex Charging Models. In L. Moser, M. Parashar, & P. Hung (Eds.), *Proceedings of the 2012, IEEE International Conference on Services Computing* (pp. 154–161). IEEE. Honolulu, Hawaii.

Schumann, L., & Stock, W. G. (2014). The Information Service Evaluation (ISE) Model. *Webology, 11*(1), 1–20.

Setten, M. v., Pokraev, S., & Koolwaaij, J. (2004). Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In P. M. E. De Bra & W. Nejdl (Eds.), *Proceedings of the 2004, International Conference on Adaptive Hypermedia and Adaptive Web Based Systems* (pp. 235–244). Springer Berlin Heidelberg. Eindhoven, The Netherlands.

Shen, Y., Wang, M., Tang, X., Luo, Y., & Guo, M. (2012). Context-Aware HCI Service Selection. *Mobile Information Systems*, (8), 231–254.

Tripadvisor. (2015). *Fact Sheet*. Retrieved from http://www.tripadvisor.com/PressCenter-c4-Fact_Sheet.html

Vanrompay, Y., Kirsch-Pinhero, M., & Yolande, B. (2009). Context-Aware Service Selection with Uncertain Context Information. In R. Rouvoy & M. Wagner (Eds.), *Proceeding of the 2009, International Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services* (pp. 1–11). EASST. Lisbon, Portugal.

Vienna Tourist Board. (2015). *Vienna Card*. Retrieved from http://www.wien.info/en/travel-info/vienna-card

Vos, H. d., Haaker, T., & Teerling, M. (2008). Consumer Value of Context Aware and Location Based Services. In AIS (Ed.), *BLED 2008 Proceedings* (pp. 50–62). AIS. Bled, Slovenia.

Vrček, N., Bubaš, G., & Bosilj, N. (2008). User Acceptance of Location-Based Services. *World of Science, Engineering and Technology, International Science Index 17, 2*(5), 260–265.

Wu, Q., & Zhu, Q. (2013). Transactional and QoS-Aware Dynamic Service Composition Based on Ant Colony Optimization. *Future Generation Computer Systems, 29*(5), 1112–1119.

Xu, L., & Jennings, B. (2010). A Cost-Minimizing Service Composition Selection Algorithm Supporting Time-Sensitive Discounts. In IEEE Computer Society (Ed.), *Proceedings of the 2010, IEEE International Conference on Services Computing* (pp. 402–408). IEEE. Miami, Florida.

Yelp. (2015). *Press*. Retrieved from http://www.yelp-press.com/phoenix.zhtml?c=250809&p=irol-press

Yu, H. Q., & Reiff-Marganiec, S. (2009a). Automated Context-Aware Service Selection for Collaborative Systems. In P. van Eck, J. Gordijn, & R. Wieringa (Eds.), *Proceedings of the 2009, International Conference on Advanced Information Systems Engineering* (pp. 261–274). Springer Berlin Heidelberg. Amsterdam, The Netherlands.

Yu, H. Q., & Reiff-Marganiec, S. (2009b). A Backwards Composition Context Based Service Selection Approach for Service Composition. In IEEE Computer Society (Ed.), *Proceedings of the 2009, IEEE International Conference on Services Computing* (pp. 419–426). IEEE. Bangalore, India.

Yu, T., Zhang, Y., & Lin, K.-J. (2007). Efficient Algorithms for Web Services Selection with end-to-end QoS Constraints. *ACM Transaction on the Web, 1*(1), 1–26. doi:10.1145/1232722.1232728

Yuan, Y., Zhang, X., Sun, W., Cao, Z., & Wang, H. (2013). Optimal Web Service Composition Based on Context-awareness and Genetic Algorithm. In IEEE Computer Society (Ed.), *Proceedings of the 2013, International Conference on Information Science and Cloud Computing Companion* (pp. 660–666). IEEE. Guangzhou, China.

Zhang, D., Boonlit, A., & Yaser, M. (2009). User-Centered Context-Aware Mobile Applications -The Next Generation of Personal Mobile Computing. *Communications of the Association for Information Systems, 24*(1), 28–46.

Zheng, X., Shi, Y., Wang, X., & Xu, C. (2013). A Context-Aware Service Selection Mechanism Based on Hidden Markov Model. In Y. Fan, X. Xu, & J. Zhou (Eds.), *Proceedings of the 2013, IEEE International Conference on Service Science* (pp. 196–201). IEEE. Shenzhen, China.