Incorporating Reputation Information into Decision-Making Processes in Markets of Composed Services

Alexander Jungmann
C-LAB
University of Paderborn
Paderborn, Germany
Email: alexander.jungmann@c-lab.de

Ronald Petrlic
CISPA
Saarland University
Saarbrücken, Germany
Email: ronald.petrlic@uni-saarland.de

Sonja Brangewitz
Department of Economics
University of Paderborn
Paderborn, Germany
Email: sonja.brangewitz@wiwi.upb.de

Marie Christin Platenius
Heinz Nixdorf Institute
University of Paderborn
Paderborn, Germany
Email: m.platenius@upb.de

***Abstract*—One goal of service-oriented computing is to realize future markets of composed services. In such markets, service providers offer services that can be flexibly combined with each other. However, although crucial for decision-making, market participants are usually not able to individually estimate the quality of traded services in advance. To overcome this problem, we present a conceptual design for a reputation system that collects and processes user feedback on transactions, and provides this information as a signal for quality to participants in the market. Based on our proposed concept, we describe the incorporation of reputation information into distinct decision-making processes that are crucial in such service markets. In this context, we present a fuzzy service matching approach that takes reputation information into account. Furthermore, we introduce an adaptive service composition approach, and investigate the impact of exchanging immediate user feedback by reputation information. Last but not least, we describe the importance of reputation information for economic decisions of different market participants. The overall output of this paper is a comprehensive view on managing and exploiting reputation information in markets of composed services using the example of On-The-Fly Computing.**

Keywords–Reputation, Service Market Interactions, Economic Decisions, Service Composition, Service Matching, Learning Service Recommendation, Game Theory.

I. PREFACE

This paper is a revised and expanded version of our paper entitled 'Towards a Flexible and Privacy-Preserving Reputation System for Markets of Composed Services' (SERVICE COMPUTATION) [1]. In addition to the original paper's scope, this expanded version covers new research results regarding the incorporation of reputation information into processes that are crucial in markets of composed services: fuzzy service matching, service composition including a learning service recommendation component, and economic decisions with respect to service market interactions. Moreover, the requirements imposed on the reputation system are expanded to account for context-specific reputation and expert ratings.

II. INTRODUCTION

A major goal of On-The-Fly (OTF) Computing is the automated composition of software services that are traded in dynamic markets and that can be flexibly combined with each other [2][3]. A user formulates a request for an individual software solution, receives an answer in terms of a composed service, and finally executes the composed service.

As an illustrative example, let us assume that someone wants to post-process a holiday video. However, it does not pay off to use a monolithic software solution because such software provides a lot of dispensable functionality, and is therefore too expensive to buy for just this purpose. What this person needs is an individually customized software composed of only those services, which together are able to satisfy his needs. A famous web-based application for individual post-processing tasks is Instagram, which provides different image processing services that can be applied to an uploaded photo or video [4]. However, the variety of available services is restricted and the selection of appropriate services has still to be done manually.

Now, let us consider a market of image processing services. A person, who wants to post-process his video, becomes a user within this market by formulating a request describing what he expects from the composed service (e.g., the functionality to create videos with reduced image noise and an increased brilliance homogeneously distributed throughout the entire video). Subsequently, a post-processing solution that satisfies the user's request is automatically composed based on image processing services that are supplied by different market participants. In this scenario, the user only has to pay for the actually utilized functionality. Figure 1 shows an exemplary use case in terms of a single photo for such a market of image processing services. The original photo depicted in Figure 1a was captured and shall be post-processed in order to change the appearance according to individual user preferences. To achieve the different effects shown in Figure 1b to Figure 1d, the functionality of a single service is usually not sufficient. In fact, composed services have to be constructed based on image processing services that are available in the market. In

(a) (b) (c) (d)

Figure 1. Original image (a) is processed by different composed services to achieve the different effects shown in (b), (c), and (d).

order to compose services that satisfy specific requirements, the quality of services regarding functional as well as non-functional properties must be taken into account.

However, for market participants who are willing to buy those services, it is difficult to estimate the quality of services before the services are actually used. For example, an image processing service's response time can be predicted to a certain extent, but it is very dependent on the specific context, e.g., its execution environment and its current load. Other markets such as eBay or Amazon solve this problem by using a reputation system. Within such a system, the experiences other users made in previous transactions are collected. Thereby, the reputation information provides new users an indicator for the service quality they can expect. As an example, let us consider that many users were entirely satisfied with a specific image processing service and rated it with five stars, for example. As a consequence, this service gained a high reputation, which makes it more attractive for future users. Not only the requesters, but also the whole market benefits from considering reputation, because the providers of high-quality products are rewarded with a high reputation, thereby increasing their chances for future sales. On the other hand, low-quality or even deceptive service providers will only be able to sell their services at low prices or will vanish from the market after some time, which again pays off for all customers. Existing reputation systems used by eBay or Amazon, for example, do not explicitly consider ratings for composed services. Other reputation systems, such as those to rate trips or hotels, often ask the user to evaluate different aspects. However, single services cannot be combined with each other as flexibly as needed in the OTF Computing market. Thus, a reputation system for composed services is still an open challenge.

The contribution of our original paper covers the identification of requirements for a reputation system for markets of composed services such as OTF Computing [1]. Furthermore, it covers the conceptual design of our proposed solution in terms of a flexible reputation system. The extended contribution of this revised version additionally covers, besides an expanded list of requirements, a more detailed description of selected processes that are essential for realizing markets of composed services. Each of these processes incorporates reputation information according to our proposed reputation system. Technical details of a prototypical implementation of our reputation system, however, are not part of the contribution and are consequently beyond the scope of this paper. The contribution of this paper is not necessarily restricted to OTF Computing alone. Results of our work can also be adopted to other areas, in which reputation of combinable products is vital.

To the best of our knowledge, there are currently no existing reputation system approaches that can be directly applied to markets of composed services like, e.g., the OTF Computing market. Reputation has already been considered in the area of service composition: A survey is presented by Gómez Mármol et al. [5]. However, each of the existing approaches only deals with a subset of the requirements we identified. For example, context-specific reputation and the consideration of expert ratings is missing in almost all related approaches and often only the reputation of services but no reputation of other market participants is considered. Furthermore, privacy protection is not considered by already existing approaches. Reputation systems that take privacy protection into account explicitly, either entail a high overhead, or privacy is only a "property", which is said to be achieved—but not enforced cryptographically.

This paper is organized as follows. Section III introduces OTF Computing while mainly focusing on those aspects that are relevant for the work at hand. Furthermore, it motivates the significance of reputation in OTF Computing. Section IV gives a detailed problem description by subsequently introducing crucial requirements for a reputation system in OTF Computing. Section V presents our conceptional solution in terms of a flexible reputation system that covers all identified requirements. Existing approaches that only partially cover these requirements are discussed in Section VI. Section VII introduces our fuzzy service matching approach that focuses on reputation information about services. Section VIII, in turn, mainly deals with the incorporation of reputation information about composed services into our composition approach. Economic decisions with respect to service market interactions under consideration of available reputation information are investigated in Section IX. Based on the heretofore presented results, Section X describes remaining research challenges that will be addressed in our future work. Finally, the paper concludes with Section XI.

III. On-The-Fly Computing

A major goal of OTF Computing is automated composition of flexibly combinable services that are traded in markets. A user's request for an individual software solution should be resolved by automatically composing a solution on demand. OTF Computing addresses the entire process, starting with fundamental concepts for organizing large-scale service markets up to the final execution of a composed service. Embedding automatic service composition into service markets is one key challenge for realizing OTF Computing. The whole OTF Computing process is very complex, as a number of different aspects need to be taken account of in an integrated fashion. In this paper, we cover three of those aspects exemplarily. At that time, we do not have an implementation of the whole OTF Computing scenario. We do have prototype implementations of the individual processes, though, which should be combined in the future in order to be able to prove the practicality of OTF Computing as a whole.

A. Automatic Service Composition

In general, we interpret automatic service composition as the sequential application of composition steps. A composition step may, for example, correspond to selecting a service in order to realize a placeholder within a workflow [6]. Regarding our initial example in terms of image processing services, a placeholder could correspond to a class of services, which provide similar functionality (such as smoothing filters). For execution, a specific service (e.g., Gaussian smoothing) must then be selected. A composition step, however, may also correspond to a single step within a composition algorithm based on Artificial Intelligence (AI) planning approaches [7][8][9][10].

For simplicity, let us assume that a workflow is available and that a service composition step corresponds to selecting a service. We divide a single composition step into two separate processes, which subsequently reduce the amount of qualified service candidates. First of all, a *Service Matching* process determines to what extent a particular service fulfills a placeholder's functional (e.g., signatures and behavior) as well as non-functional requirements (e.g., quality properties such as response time or reliability) [11][12]. Based on the matching result, services that provide significantly different functionality or that violate important non-functional restrictions can be discarded directly. Subsequent to the matching process, a *Service Recommendation* process identifies (and ranks) the best service candidate(s) out of the set of remaining services. During the recommendation process, explicitly given non-functional objectives regarding the final composed service (e.g., maximizing the performance while simultaneously minimizing the costs) as well as implicit knowledge from previous composition processes (e.g., a certain service is more qualified in a particular context than others) are incorporated. The incorporation of knowledge from previous composition processes is realized by means of Reinforcement Learning (RL) [13], and requires feedback about the quality of the execution result [14].

B. Market Infrastructure Perspective

Figure 2 shows the transactional view on the entire OTF Computing process, reduced to those processes that are relevant for the work at hand. *OTF Provider Selection* and *Service Provider Selection* are decision-making processes regarding transactions within the market. Three different classes of market participants are involved in the overall process: users, OTF providers, and service providers. A user formulates a request for an individual software solution and sends it to an OTF provider of his choice (*Step 1*). The selected OTF provider processes the request and automatically composes a solution based on elementary services that are supplied by independent service providers.

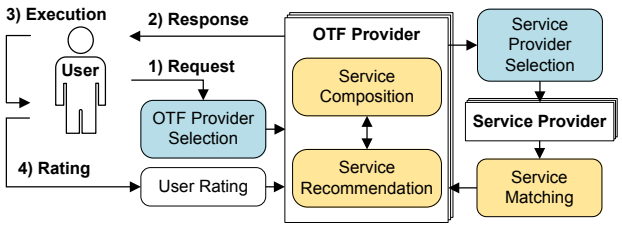


Figure 2. Overall OTF Computing process.

For each composition step, an OTF provider asks a selected subset of service providers for elementary services. The previously mentioned matching process is part of the OTF architecture and takes place before an OTF provider receives answers about appropriate elementary services. The matching process operates as a filter ensuring that only services that fulfill the desired requirements to a certain extent are returned. The recommendation process, in turn, is part of the OTF provider-specific composition process and highly depends on the context of the request.

As soon as a composed service is created, it is passed on to the user (*Step 2*), who subsequently executes it (*Step 3*). After execution, the user rates his degree of satisfaction regarding the quality of the execution result (*Step 4*). In the current setting, the value of the user rating is immediately returned to the associated OTF provider. By transforming the value into a reward and incorporating it into the RL process within the recommendation system, the OTF provider improves its internal composition strategy (recommendation process) for future user requests [15].

C. Reputation as a Signal for Quality

In a dynamic market of software services, information about quality (e.g., service quality or the quality of OTF providers) is essential. A user may resort only to OTF providers of a certain quality (e.g., with respect to customer support), while simultaneously accepting only composed services of a certain quality level (e.g., composed services with high reliability and trustworthiness). OTF providers, in turn, have to build composed services consisting of elementary services with a quality level according to a user's request. Information about quality, however, is either difficult to estimate before a transaction actually took place, or cannot be simply trusted if the quality information is provided by the associated market participant itself (e.g., when a service provider specifies the quality of his own services). Our solution to overcome these issues is to replace the previously mentioned and fairly simple user rating procedure (cf. Figure 2) with a flexible reputation system, which aggregates user ratings into single reputation values and provides them to market participants. Reputation can then be incorporated as an estimation of quality into the different decision-making processes.

IV. Problem Description and Requirements

Our goal is to explicitly incorporate reputation information as an estimation of quality into the OTF Computing process. Using goal-oriented requirements engineering [16], we systematize our reputation system requirements by investigating the role of reputation from different perspectives.

A. Reputation Information Within the OTF Process

As shown in Figure 2, the OTF Computing process is initiated by a user's request. To enable users to choose an OTF provider they want to establish a business relationship with, i.e., to buy a composed service from, reputation information about OTF providers must be available.

(*R1*) *OTF Provider Reputation:* The reputation system must provide reputation information about OTF providers.

The selected OTF provider has to ensure that the requested composed service satisfies the user's requirements regarding reputation. For this purpose, the reputation of service providers and the reputation of their supplied elementary services has to be considered during the composition process. In order to enable OTF providers to select service providers they want to retrieve elementary services from, reputation information about service providers must be available.

(*R2*) *Service Provider Reputation:* The reputation system must provide reputation information about service providers.

Reputation of elementary services influences the reputation of composed services. For example, if a composed image processing service uses a well-known, reputable implementation of a specific image filter, it can be assumed, that the composed service's reputation will be higher than the reputation of a service composed of unknown elementary services. Thus, the service matching processes (cf. Figure 2) as well as the service recommendation process have to consider the reputation of elementary services. While the matching process has to determine to what extent an elementary service fulfills certain requirements considering reputation, the recommendation process has to determine the best composition steps including reputation. Reputation information, however, cannot be simply extrapolated from service providers to elementary services, since a service provider may supply services of varying quality. Therefore, reputation information about elementary services must be available, too.

(*R3*) *Service Reputation:* The reputation system must provide reputation information about elementary services that have been consumed as a part of a composed service.

The recommendation process additionally rates alternative composition steps based on experience gained from previous composition processes. Reputation information about previously composed services is needed as feedback for the recommendation process in order to adapt its recommendation strategy by means of RL. An OTF provider's experience, however, can be considered a business secret that must not be revealed to other market participants.

(*R4*) *Composed Service Reputation:* The reputation system must provide reputation information about composed services without revealing business secrets of OTF providers.

Up to now, we focused on the overall reputation of (composed) services and providers. However, in reality, reputation is rather context-specific [17]. For example, an image processing service could have a good reputation regarding the response time but a bad reputation regarding security. Thus, the reputation system should maintain vectors instead of single values for ratings and reputation. This provides requesters with the possibility to specify more detailed requests, e.g., "I want an image processing service with a high reputation with respect to security, but its reputation for response time is not that important to me".

(*R5*) *Context-specific Reputation:* The reputation system must distinguish between reputation values based on different contexts, i.e., based on different properties of the rated service or provider.

Users only interact with OTF providers and not with service providers directly (cf. Figure 2). As a consequence, a user's ratings mainly contain information about OTF providers and their composed services. Only once in a while may a user be able to additionally rate elementary services. For example, when using a composed service for an image processing task, users may not be aware of all elementary services, e.g., of the filter service that reduces image noise. However, they may be able to rate an elementary service that implements an image compression algorithm, since the way the algorithm effects the execution result can be directly observed in terms of the size and quality of the generated image or video.

(*R6*) *Incomplete User Rating:* The reputation system has to consider that a user is most often just able to rate OTF providers and their composed services, while a user is only sometimes able to rate elementary services and never able to rate service providers.

In certain scenarios, users might be especially interested in ratings by *experts*, i.e., people who have a well-known expertise in that domain. The reputation system should allow for a flexibility when it comes to dealing with ratings provided by experts. As an example, the system could support weighting experts' ratings more than ratings provided by non-experts, i.e., "ordinary" users.

(*R7*) *Expert Ratings:* The reputation system shall provide a flexible mechanism of handling experts' ratings.

B. Technical Requirements

The reputation system needs to provide access to the different reputation values mentioned in the previous section for the different parties illustrated in Figure 2. Those parties have diverse and variable needs for reputation value computations and access as well as interaction preferences. For the service recommendation process, recent ratings are more important to accelerate the learning process and, therefore, reputation value computations that put a higher weight on those ratings are desired (e.g., rather a geometric mean than an average with equal weights). In contrast, for a user, it might be preferable that a certain composed service has a very low failure rate and, thus, during the provider selection process, reputation values that include historic values to a sufficient extent and put a higher weight on negative ratings have to be considered. The reputation system's functionality to process user ratings and to provide them as reputation information has to satisfy the diverse needs of the requesting parties.

(*R8*) *Flexible Processing of User Ratings:* The reputation system must support flexible processing of user ratings.

Certain restrictions may be applied: Concerning requirement (*R4*), reputation information about composed services shall be

retrievable only by the OTF provider that originally accomplished the service composition process.

(*R9*) *Access Control:* The reputation system must implement access control to reputation values.

Furthermore, the reputation system shall support different interaction models. Parties, such as the OTF provider's service recommendation component, need new reputation information as soon as it is available. New reputation information has to be automatically forwarded by the reputation system without explicitly asking for it. Other processes that rarely need to retrieve reputation information, such as users or the service matching component, shall be able to access those data actively on demand to reduce the data traffic.

(*R10*) *Interaction:* The reputation system must support alternative interaction concepts. Reputation information must either be provided on demand triggered by a request event, or actively sent to a party as soon as new reputation information is available.

Furthermore, security and privacy protection are crucial issues—as we have already investigated more generally for the OTF Computing as well [2]. If users could arbitrarily rate any services (without having used them), the reputation system would not constitute any benefit. If any party would be able to manipulate the reputation values, users could not trust the provided values and, thus, the reputation system's benefit would be lost as well.

(*R11*) *Rating Authorization:* Only authorized users, i.e., users that performed a transaction with an OTF provider, are allowed to rate that transaction. If a rating shall count as an "expert rating" (compare (*R7*)), a special authorization is needed: The expert needs to be recognized and authenticated as an expert during the rating process.

(*R12*) *Correctness:* The computed reputation value provided by the reputation system must be correct, i.e., it must not be possible for any party to manipulate the reputation value (computation).

Depending on the traded services in the market, users might only be willing to rate transactions if they can stay anonymous. They do not want to (publicly) reveal, which services were consumed by them. It has been shown in the past that designing a reputation system that provides user anonymity is a challenging task [18].

(*R13*) *Anonymity of Rating User:* No party shall be able to relate (individual) ratings to users. Even expert raters shall get the possibility to stay anonymous—if they want to; still, they need to be authenticated as experts (in order for their ratings to count more, for example) but their ratings shall not be linkable to them.

(*R14*) *Unlinkability of User Rating to Transaction:* The OTF provider must not be able to relate a rating to a transaction (previously executed with a certain user)—in order to achieve user anonymity.

V. A Flexible Reputation System

This section introduces the conceptual design of our proposed solution in terms of a flexible reputation system. First, the system's internal processes as well as its interaction capabilities are described. Afterward, we illustrate in particular how the system meets each requirement listed in Section IV. An overview of our proposed solution is given in Figure 3. It shows the internal structure of our flexible reputation system as well as the interactions with the OTF Computing process.

A. Basic Internal Structure

The reputation system is modeled as a stand-alone and independent component within the OTF Computing environment. The reputation values are derived by processing user ratings of services, composed services, as well as OTF providers. The internal structure can be divided into three main sections.

The *Accumulated Ratings* section provides functionality for accumulating raw values of incoming user ratings over time. To increase robustness, these values can be stored by means of a distributed storage system. The number of values to be stored is not necessarily restricted. However, depending on the available storage space and the amount of incoming values, outdated values may either be discarded or at least consolidated into a lower amount of values in the long run.

The *Aggregation System* provides functionality for processing a set of raw values in order to generate an aggregated representation. However, one can flexibly choose the set of raw values to be incorporated into the process, the actual aggregation function to be applied (e.g., arithmetic/geometric averaging, identifying the maximum or approximating the future trend by time series analysis) and the final representation (e.g., single scalars such as mean or median, or density functions in terms of their statistical parameters).

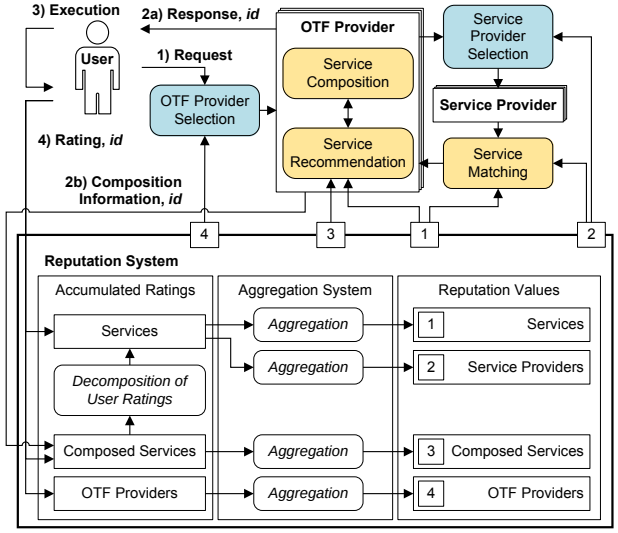


Figure 3. Proposed OTF Computing Reputation System. Internal structure and interactions with the OTF Computing process are depicted.

The *Reputation Values* section finally provides the interfaces for accessing the different reputation values of services, service providers, composed services, and OTF providers. When accessing reputation values, the set of raw user ratings to be considered, the actual aggregation function, as well as the final representation can be flexibly specified. Reputation values are not stored within the system, but always computed on demand dependent on the previously mentioned specifications. This flexibility allows requests for reputation information to adapt to more complex reputation requirements imposed by users. For example, a user may want an image processing service with a reputation value higher than 4 based on at least 20 user ratings that are not older than 6 months. Another user may want an image processing service, which has an average reputation value of 4, while no elementary service should have a reputation value less than 2.

B. Integration into the OTF Computing Process

Reputation values are consumed by the *Service Matching*, the *Service Recommendation*, the *Service Provider Selection*, and the *OTF Provider Selection* processes within the overall OTF Computing process. Besides flexibility regarding how a reputation value is internally computed, our proposed reputation system also provides flexible interaction capabilities. On the one hand, reputation values can be accessed by a *pull* approach whenever they are needed. Following this approach, the requester inherits the active role by asking for reputation data if and only if it is necessary. This solution is efficient when reputation information is needed less frequently (e.g., when a user wants to choose an OTF provider). On the other hand, a *push* approach shifts the active role to the reputation system. Reputation information is sent to a party as soon as new data is available. This approach also allows for creating a local cache of the latest reputation values without flooding the reputation system with redundant requests for possibly new information.

Figure 4 shows the interaction with the proposed reputation system using the example of the service matching process (matcher). During the OTF Computing process, the matcher is called for each elementary service that possibly satisfies an OTF provider's request (cf. Section III-B). In this context, Figure 4 illustrates the access of reputation information for exactly one elementary service by a *pull* approach.

The reputation matching process is initiated by providing the request information and the description of an elementary service and by calling the *match* operation. For the sake of simplicity, the request in the depicted example only shows an extract: An image processing service should have a minimum reputation value of 4. This request shall now be matched against an elementary service with id `ImagePro1`. The matcher asks the reputation system for a reputation value of service `ImagePro1` aggregated by means of an aggregation function with id `f_id`. Hence, the aggregation system fetches the relevant user rating values (3,4,3,5,5) from the storage, selects the corresponding aggregation function (here, arithmetic averaging), and computes an aggregated reputation value of 4. Based on this result, the matcher decides that the service matches to the request. We describe the challenges of reputation matching and our matching approach in more detail in Section VII.

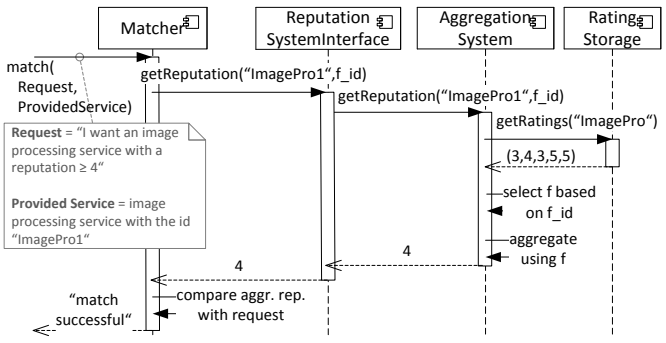


Figure 4. Simplified example interaction with the reputation system.

After a composed service was executed (*Step 3* in Figure 3), users are encouraged to provide feedback on their transactions. They are asked to rate composed services, OTF providers, and single services. The feedback in terms of user ratings is the foundation for generating reputation information within the reputation system. If the rating is provided by an expert, this information needs to be stored as well, as expert ratings can be given more weight than ordinary ratings—this depends on the requirements of the scenario and can flexibly be implemented as part of the reputation value computation as discussed before. To be able to identify, which composed service a rating belongs to, OTF providers attach an *id* to their response (*Step 2a* in Figure 3). This *id* corresponds to the particular structure of a composed service, meaning that identical composed services have identical *id*s. During the rating process for a composed service, this *id* is forwarded to the reputation system (*Step 4* in Figure 3).

Elementary services that are consumed as part of a composed service cannot always be rated separately by the user. In fact, due to complex user requests, we expect that this is rarely possible. Thus, in order to still be able to provide reputation values for elementary services and to benefit from all information available, our reputation system decomposes user ratings of composed services. To enable this decomposition, the *id* the OTF provider sends with his response (cf. Figure 3) is reused: Simultaneously with his response to the user (*Step 2a* in Figure 3), the OTF provider sends the same *id* together with *composition information* to the reputation system (*Step 2b* in Figure 3).

As pointed out above, our reputation system for OTF Computing shall provide flexibility, which also means that different implementations for the components are supported. We have already shown that such an implementation of a reputation system for the OTF Computing can be done in a *secure* and *privacy-preserving* way—respecting the requirements stated in Section IV [19]. In contrast to related work, as covered in Section VI, this approach only requires a single *reputation provider*, which is in line with the requirements of OTF Computing, and does not need any other components (such as a bulletin board). The approach is based on the Paillier cryptosystem [20] to provide a reputation value as an aggregation of individual user ratings without revealing anything about the individual ratings to any party. At the moment we are investigating if expert ratings can also be implemented in a privacy-respecting way. For that purpose,

we are investigating *group signatures* in more detail, as that mechanism shall allow the authentication of experts without revealing their identities.

C. Satisfying OTF Computing Requirements

Our proposed solution in terms of a flexible reputation system fulfills all requirements listed in Section IV. This section points out how the reputation system fulfills each of these requirements in particular.

The proposed reputation system enables users to rate OTF providers, composed services, and—if possible—elementary services. Assured by the transferred *id*, in this context, only users that are involved in a particular transaction taking place in the OTF Computing market, i.e., users that have requested, received and executed a particular composed service, are allowed to participate in the rating process. This ensures ratings by authorized users (*R11*). How to realize the rating process in particular (i.e., what kind of questions have to be asked and how a user rating value is represented) is beyond the scope of this paper.

Correctness of the provided reputation values is ensured by design. Reputation values are computed on demand by the system itself based on a pre-defined set of aggregation functions. Furthermore, the entire system is an independent component within the OTF Computing environment. As a consequence, manipulations of the computation process by other participants are eliminated (*R12*).

Anonymity of users (*R13*) as well as unlinkability of user ratings to transactions (*R14*) is ensured by the accumulation and aggregation functionality. For reasons of privacy protection, i.e., in order to not reveal individual user ratings, the reputation system always collects individual ratings and aggregates them. Although the single user ratings are stored within the reputation system, they are not accessible to market participants so that individual ratings are not traceable. In this context, it is important that the amount of accumulated user ratings is high enough and that the aggregation operation sufficiently condenses the user ratings such that it can be guaranteed that no information on individual ratings can be recovered. If not enough user ratings are included in the aggregation process (e.g., when not enough user ratings are available yet, or if a request explicitly specifies to only consider just a few user ratings), the reputation system will not provide a value but will raise an exception.

All processes that need reputation information within the entire OTF Computing process have access to the reputation system. The flexibility of our proposed solution enables each market participant to freely choose an interaction approach (*push* or *pull*) that is most appropriate with respect to the market participant's internal processes (*R10*). Furthermore, the process of generating reputation values can be adjusted by each market participant individually by specifying the set of user ratings to be considered, the actual aggregation function to be applied, and the final representation of the aggregated value (*R8*).

Reputation information about OTF providers (*R1*) is provided by the reputation system in a straight-forward manner. Users rate their satisfaction regarding the transaction with an OTF provider. These ratings are accumulated and aggregated by the reputation system and can be accessed by other users. The process of generating reputation information about composed services (*R4*) is similar. Users rate their satisfaction regarding the execution process and the execution result of a composed service. These ratings, again, are accumulated and aggregated by the reputation system. In comparison to the reputation of OTF providers, however, reputation information about composed services is OTF provider-related. In order to preserve business secrets, only the OTF providers themselves can access the reputation values of their own composed services - after successful authentication (*R9*).

Besides being directly rated by users, ratings of elementary services also have to be derived from ratings for composed services (*R3*). For this purpose, OTF providers send information about their composed service to the reputation system. In order to not reveal their business secrets, this composition information, however, only consists of abstract, structural information. Only the set of elementary services included in a composed service is exposed, but not, for example, when and how often a particular service is called. This way, the provider's business secrets are protected, while it also allows for a mapping of the rating for a composed service to single services (*R6*).

Since users only interact with OTF providers, user ratings for service providers cannot be provided to the reputation system (*R6*). To overcome this problem, the aggregation system extrapolates from reputation information about elementary services to information about the associated service providers during the aggregation process (*R2*).

While composing services, reputation information about elementary services have most likely to be aggregated in order to choose composed services not only based on their (aggregated) non-functional properties, but also based on their overall reputation. How to determine this overall reputation, however, depends on the user requirements and the composition strategy of the respective OTF provider. If a user requires, e.g., all elementary services to satisfy a minimal reputation value, an OTF provider has to check the reputation value of each service individually. Another user might be satisfied with an average reputation value above a specific threshold. In this case, an OTF provider has to determine the average reputation value by aggregating all single values. Subsequently, the aggregated value and the threshold value have to be compared. In either case, aggregation of reputation values within the composition process is not part of the reputation system itself. A further investigation of how to integrate reputation information into service composition is addressed in Section VIII.

VI. Related Work

There is a lot of literature on reputation, both in economics and computer science. Our interpretation of reputation is used for instance by Shapiro [21] or as well by Bar-Isaac and Tadelis [22], who summarize the economic literature on reputation. Design aspects related to mathematically modeling a reputation system and challenges that arise with online transactions, are explicitly discussed by Friedman et al. [23] and Dellarocas [24], for example. Another comparison of trust and reputation models without referring to services or

service composition is given by Gómez Mármol and Martinez Pérez [25].

More closely related, we identify three involved fields, *Reputation Systems*, *Privacy-Preserving Systems* and *Service Composition*, and their overlaps with each other as shown in Figure 5. In the following, we present related work, which has been done within these overlaps in more detail. It is noteworthy that no work covers all of the three different fields (the overlapping marked **x** in Figure 5). To the best of our knowledge, we are the first to take all three fields into account.

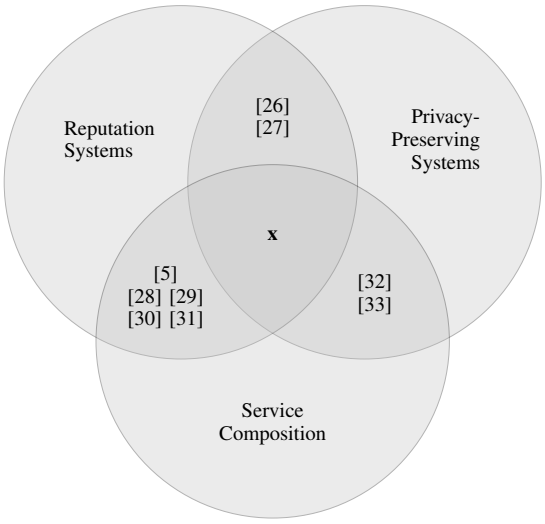


Figure 5. Overview of related work.

A. Reputation Systems and Privacy-Preserving Systems

Researchers have come up with *privacy-preserving* reputation systems in the past. Androulaki et al. [27] propose a reputation system that allows anonymous ratings. However, there is no authorization mechanism in their approach, i.e., anybody could rate any service. Kerschbaum et al. [26] present a system, which requires two centralized mutually mistrusting reputation providers in order to achieve anonymous user ratings. Users encrypt their ratings and send them to the first reputation provider, which collects a number of ratings and then publishes them to a bulletin board. The second reputation provider retrieves the ratings from the bulletin board to decrypt and aggregate them before providing a (computed) reputation value. The approach is based on the Paillier cryptosystem [20]. However, the approach is too inflexible and complex to be used in our OTF Computing setting. We want to keep a lean OTF infrastructure with only one reputation provider and no other additional components, such as a bulletin board, used only by the reputation system.

B. Reputation Systems and Service Composition

In general, existing approaches in this area focus on technical issues, e.g., how exactly reputation of composed services is computed. Our work does not focus on these issues but rather provides a holistic overview of how reputation is used in different processes of OTF Computing.

A good overview of existing research in the crosscutting area of reputation systems and service composition is provided by Gómez Mármol and Kuhnen [5]. In their survey, they compared 12 approaches on reputation-based web service composition. According to their analysis, most of the considered approaches suffer from security issues like sybil attack. Furthermore, many of those approaches offer some flexibility in a way that they support different aggregation strategies. The latter is similar to our approach. However, these approaches are not applicable to OTF Computing because they do not consider reputation of the different roles, i.e., OTF Provider and Service Provider. Also other requirements are not covered, e.g., context-specific reputation, expert ratings, or different ways of interaction. Moreover, the matching approaches used in the considered approaches to select services for a composition based on reputation are rather simple and do not support complex requests or different fuzziness sources.

Ali et al. present a reputation system with an integrated *Service Composer* [28]. They combine reputation with service composition by evaluating reputation metrics whenever services are composed. The *reputation computation phase* calculates reputation for elementary services as well as for composite services. In another approach, Motallebi et al. integrate *Component Reputation* and *Component Trust* in order to derive the reputation of a composed service from trust values for single services [29]. They do this by taking into account the frequency of invocations of these services. Both approaches covers only some of our requirements for a reputation system in OTF Computing. For example, neither service providers are considered, nor is privacy or security a topic within their publications.

Malik and Bouguettaya present the framework RATEWeb [30], which aims at facilitating service composition considering a service's reputation. In this approach, the service comsumer is responsible for maintaining reputation values, i.e., the reputation system is distributed. This contradicts with our idea of OTF providers that use "global" reputation values within service composition. However, their reputation metrics are very comprehensive in a way that they consider different aspects of computation, e.g., rater credibility, majority rating, and temporal sensitivity. We focus on a more flexible method to be configured by the user at runtime. In future, we should cover a similar range of aspects in our requirements, though.

The reputation propagation framework by Huang et al. [31] considers "various entities in the [service] ecosystem", e.g., not only a service's reputation but also the providers' reputation is considered. This makes it similar to our approach. However, they do not describe how reputation is considered within other processes, e.g., service composition. Furthermore, the matching process itself is rather simple, based on one numerical value. An interesting idea is that they take the domain of a service into account. This might also be an interesting addition to our approach in the future.

C. Service Composition and Privacy-Preserving Systems

Tbahriti et al. identify privacy preservation as one of the most challenging problems in *Data-as-a-Service (DaaS)* services composition [32]. DaaS is about combining web services for data publishing and sharing. In their proposed approach, *privacy policies* specify how collected data is treated and *privacy requirements* specify how the service-consuming

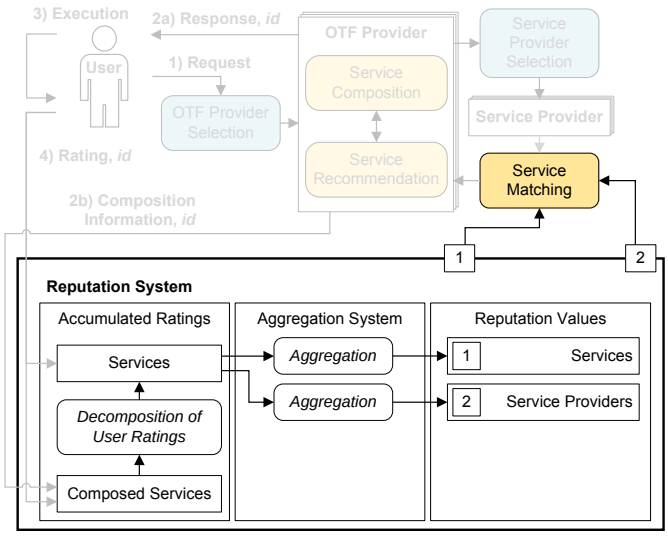


Figure 6. Overview: Service Matching

services are expected to treat the provided data. Similarly, Costante et al. come up with a solution for web service selection and composition that takes privacy into account [33]. Users are able to specify their privacy preferences, which are checked against the service providers' privacy policies. Only in the case of a successful match are the service providers' services selected and used for composition. Both approaches do not take into account reputation of elementary or composed services.

In contrast to related work, we pursue a *privacy-by-design* approach that builds privacy protection into the reputation system for OTF Computing. This allows us to prove that privacy is achieved rather than to rely on guarantees made by the participants.

VII. Fuzzy Reputation Matching

In general, service matching is the process of comparing a request for a certain service to descriptions of the services provided in a service market. For each provided service, a matching process delivers a matching result that indicates how much the service satisfies the given request. Here, we do not distinguish between composed services and elementary services: the provided service can be of both types. The matching process includes functional properties, e.g., signatures or protocols, as well as non-functional properties, e.g., quality properties of the service, like performance. A service's reputation is part of the non-functional properties. The most important difference between reputation matching and matching of other service properties is the source of the information the request has to be matched to: While properties like signatures and protocols have to be specified by the service providers themselves, reputation data has to be determined by a reputation system managed by a trusted third party – the OTF provider in our case. This is due to the fact that the ratings a reputation value is aggregated from are security- and privacy-sensitive (see Section IV). The provider must be prevented from manipulating such data (also covered by *R12*).

Section V-B illustrated a very simple reputation matching example. In practice, we expect requesters to require a more complex matching approach. The reason is that their requirements, on the one hand, can be more much complex but, on the other hand, they can also be fuzzy. Similar facts hold for the information provided in the reputation system. In the following, we explain how a more complex reputation matching works, which kinds of fuzziness can occur, and how we cope with them. As depicted in Figure 6, in this section, we only focus on the matching process and not on the process that gathers the ratings needed to calculate reputation. The reputation matching process depends on reputation values for services and service providers.

A. Reputation Matching

A request for a service's reputation consists of a list of conditions that can be fulfilled or not. As an example, consider the request in Figure 7. This request consists of four conditions, `c1` – `c4`. These conditions can be evaluated based on data from the reputation system. For a full match, all four conditions have to be true. If not all conditions are true, the matching approach returns a result that denotes to what extent the request is satisfied. Based on this result, the requester (or the OTF Provider's recommendation system, see Section VIII) can compare and select between different services. The more complex a request is, i.e., the more details a requester specifies, the more accurately can the matcher determine results that actually fit the requester's interests. However, with an increasing complexity, also the set of services matching the request to a high extent becomes smaller. We explain the example request depicted in Figure 7 in more detail in the following.

Each of the conditions in a request checks several properties related to service reputation (*R3*) and a service provider's reputation (*R2*). For example, `c1` checks whether the overall reputation of a service is greater or equal to `4` (based on a five star range as it is common in today's app stores). As a further restriction to this reputation value, this value must have been aggregated based on at least 100 ratings. Such restrictions are useful as a reputation value's reliability increases with the number of ratings it has been calculated from. The conditions `c2` and `c3` check context-specific reputation values (*R5*), i.e., the reputation with respect to the perceived response time of the service (`c2`) and the reputation with respect to the perceived security of a service (`c3`). Furthermore, `c2` uses an approximation operator ($\approx$). It means that the lower bound should be approximately `4` but the requester is tolerant regarding slight deviations. For example, a reputation of `3.95` would also be accepted. We will give more details about such approximated conditions in the next subsections related to fuzzy matching. In `c3`, we can see a restriction with respect to time. In this example, the reputation value should have been created based on at least 50 ratings, which have been given during the last three months. These kinds of restrictions are based on the idea that recent ratings are more reliable than old ones. This especially happens if the rated service has been updated or if the environment of the raters changed (e.g., the global sensitivity to security increased due to some incident). In contrast to `c1` – `c3`, `c4` is about the reputation of the service's provider. Another special characteristic of `c4` is the restriction `newer > older`. Based on the same idea

Request:

c1: Rep(Service) ≥ 4
based on ≥ 100 ratings

c2: Rep_{RT}(Service) ≈ 4
based on ≥ 50 ratings

c3: Rep_{Sec}(Service) ≥ 3
based on ≥ 50 ratings
of the last 3 months

c4: Rep(Provider) ≥ 3
newer > older

Reputation System:

Entity	Context	# Ratings	
Service: ImagePro1	Overall	300	Rep(300 ratings) = 4.5
Service: ImagePro1	ResponseTime	80	Rep (80 ratings) = 3.5
Service: ImagePro1	Security	30	Rep (30 ratings) = 2
Provider: IPServices	Overall	1300	Rep (newer > older) = 4
...	...	...	...
Service: PicProcessor	Overall	90	Rep (90 ratings) = 3.35
Service: PicProcessor	ResponseTime	30	Rep (30 ratings) = 1.5
Service: PicProcessor	Security	30	Rep (30 ratings) = 4.5

Figure 7. Exemplary request and extract of reputation system contents

as explained above, this restriction is related to the age of the ratings the creation of the reputation value is based on. However, in this case, the requester did not specify a specific threshold as above. Instead, ratings are weighted based on their age. For example, ratings from the current month have a higher impact on the reputation value than ratings given a year ago. Please note that flexible feedback processing (*R8*) is an essential prerequisite for such complex requests as they affect the aggregation function the evaluated reputation values are based on.

The right part of Figure 7 shows an exemplary extract of the contents of a reputation system in a tabular notation. These contents, amongst others, are used to evaluate the conditions of the request explained above. For example, reputation values based on different contexts for the services `ImagePro1` and `PicProcessor` are depicted. We also see the reputation of the service provider `IPServices`, which is the provider of both the `ImagePro1` and the `PicProcessor` services. The third column depicts how many ratings are available per service in total. The rightmost column depicts some exemplary reputation values calculated based on these ratings. Please note that these are dynamic values not stored in the reputation system but derived from the ratings stored in the system based on a selected aggregation function.

After the required reputation values considering all requested restrictions have been determined, an exact matching of each condition is a simple numerical comparison. For example, for `ImagePro1`, `c1` evaluates to `true` because the overall reputation value can be calculated based on 300 ratings and turns out to be 4.5 (cf. the first row of the reputation system depicted in Figure 7). In contrast, `PicProcessor` can already be sorted out as the overall reputation can only be determined on 90 ratings (and it is only 3.35, anyway). From this example, we can learn that an exact matching approach approach like this always depends on complete knowledge, e.g., the requested number of ratings. In the following, we explain why this is an unrealistic assumption and how we can deal with incomplete knowledge using fuzzy matching.

B. Fuzziness Types

Since the reputation of a service is not an objective measure, such as signatures or protocols, uncertainty or *fuzziness* might easily be introduced into the matching process. The more detailed the request, the more possibility there is for induced fuzziness.

Fuzziness can be introduced into the matching process due to several reasons. In our earlier work, we classified these reasons into different fuzziness types [11][12] including Requester-induced Fuzziness and Provider-induced Fuzziness:

- *Requester-induced Fuzziness:* Requesters are often tolerant with regard to slight variations between the stated requirements and the provided service. Especially "soft" constraints, like the conditions in a reputation request, are likely to be vague. For example, a requester wanting a reputation of 4, might also be satisfied with a service having a reputation of 3.95 if all other properties match well.
- *Provider-induced Fuzziness:* We expect service providers to not provide all information a service matching approach needs to determine an exact match in most cases. Reasons for this include (a) they do not want to publish many details in order to protect business interests and (b) they cannot know all required service properties because these details are difficult to determine, e.g., the overall performance of a service. Regarding reputation matching, provider-induced fuzziness occurs if the reputation system does not have the data needed to evaluate a condition on a given request. For example, if a service is rather new in the market, there cannot be many ratings for this service. Another reason might be unrateability of the provider or of a service that has only been used as part of a composition (*R6*).

Our main idea is to enable matching despite these uncertainties and to make them visible to the requester. Thus, the matching approach should not only return a matching result but also a measure of each of the three fuzziness types. As a benefit, the requester can make a more informed decision. In most cases, requesters will prefer services with good matching results that come with a low amount of fuzziness. Furthermore, if the requester-induced fuzziness is high, the requester can even react to it by modifying his request and restarting his search. Similarly, even the provider can try to react if he finds that matching results returned for his services are often inflicted with a high amount of provider-induced fuzziness in order to increase its sales opportunities.

There are different ways of how to represent a matching result that reflects induced fuzziness. For example, in our previous work, we assigned percentage fuzziness scores to privacy policy matching [34]. In the following, we will show an

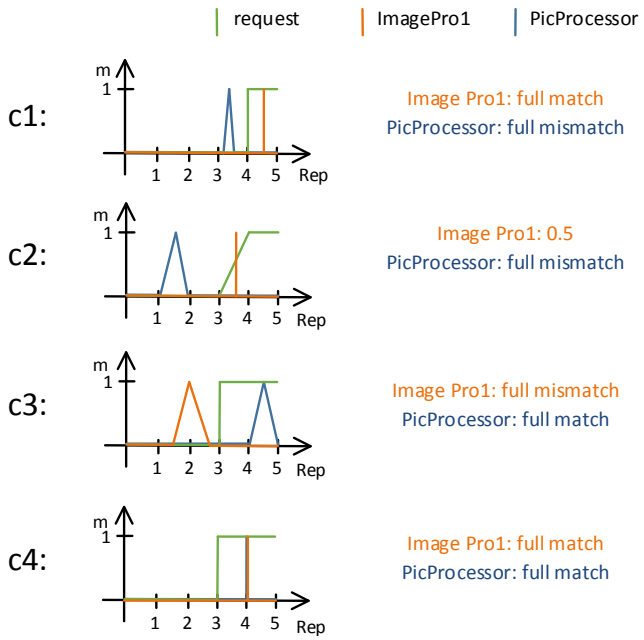


Figure 8. Exemplary Matching Results

approach appropriate for reputation matching based on fuzzy logic, more specifically fuzzy sets [35], as fuzzy logic has already been shown to be useful in order to model uncertainty in many works (e.g., [36], [37]).

C. Calculation of Fuzzy Matching Results

The example request depicted in Figure 7 covers requester-induced fuzziness as well as provider-induced fuzziness. Requester-induced fuzziness occurs in `c2`, indicated by the approximation operator $\approx$. When analyzing the service `ImagePro1`, provider-induced fuzziness occurs in `c3` because the reputation system does not have 50 ratings regarding security of this service but only 30. When analyzing the service `PicProcessor`, provider-induced fuzziness occurs in `c1`, `c2`, and `c3` as there is only a little amount of ratings for this service.

In order to evaluate `c1` – `c4`, these conditions are transformed into membership functions. Conditions inflicted with fuzziness are fuzzified into fuzzy sets. Figure 8 depicts these sets. The x axes denote reputation values in a scale from 0 to 5, while the y axes represent the membership as a number between 0 and 1. The sets for the request are depicted in green color. For example, the threshold for the requested reputation in `c1` is 4. Thus, the membership is 0 from 0 to 4 and 1 between 4 and 5. This means, if a service's reputation value is higher than 4, it matches completely. For `c3` and `c4`, the thresholds are 3. `c2` is transformed into a fuzzy set as there is no hard threshold. Depending on the risk affinity of the requester, the transition can be more or less steep. However, for simplicity reasons, at the moment, we model the steepness based on a constant value of 1. Thus, in this example, we modeled the fuzzy constraint with a smooth transition between full and no membership from 3 to 4. Modeled as a tuple, a fuzzy set can be denoted by its lower left corner, its upper left corner, its upper right corner, and its lower right corner. Accordingly, this fuzzy set is denoted by the values (3,4,5,5).

The orange and the blue lines are the membership functions for the two provided services to be matched (`ImagePro1` and `PicProcessor`). For example, the reputation value of `ImagePro1` is 4.5. This leads for `c1` to a membership of 1 at 4.5 and 0 otherwise. As stated above, for `PicProcessor`, there are not enough ratings to evaluate `c1`, so we model the provided reputation value as a fuzzy set. The fuzzy set is derived from the information we have and from the amount of missing information. In this example, we know that the reputation is 3.35 based on 90 ratings. It is uncertain how the reputation value would develop with 10 more ratings (a value based on 100 ratings was requested). However, assuming an averaging aggregation function for the reputation, the value cannot deviate much. Thus, based on the ratio of missing ratings and available ratings, we span a triangular fuzzy set. The peak is the value we calculated based on the currently available ratings and the transition, i.e., the steepness on both sides is determined by the number of missing ratings and the number of requested ratings. In this example, the peak is at 3.35 and the steepness on both sides is 0.1 (10 missing ratings out of 100 ratings requested: $10/100$). This leads to a fuzzy set of (3.25,3.35,3.45). In `c2`, there is more uncertainty about the reputation value of `PicProcessor` as even more ratings are missing compared to the request: The reputation value calculated based on the currently available ratings is 1.5, 20 ratings are missing and 50 are requested. In this case, we use the current reputation value of 1.5 and a deviation of 0.4 ($20/50$), which denotes a fuzzy set of (1.1,1.5,1.9).

Based on these membership functions, the provided services are compared to the request in order to decide whether the services match. Regarding `c1`, we have a full match for `ImagePro1` because its membership is fully covered by the request's membership function. `PicProcessor` does not match with respect to `c1` because there is no overlap with the set represented by the request's membership function. The evaluation of `c2` shows that the reputation value for `ImagePro1` intersects the request's membership function at a value of 0.5 on the y axis. Accordingly, the matching result for this condition is 0.5. `PicProcessor` does not match `c2`. Regarding `c3`, `ImagePro1` matches, while `PicProcessor` does not match. Furthermore, both services match with respect to `c4`.

The results for the four conditions are aggregated to one final matching result per service. Taking the average again, `ImagePro1` matches with a result of

$$(1 + 0.5 + 0 + 1)/4 = 0.625\ ,$$

and `PicProcessor` matches with a result of

$$(0 + 0 + 1 + 1)/4 = 0.5\ .$$

As a conclusion, the requester should choose `ImagePro1`.

D. Configurability

In Section VII-C, we described how we derived membership functions from the request that describes a user's requirements on reputation. There are some configuration possibilities to adapt these functions. For example, the steepness of the

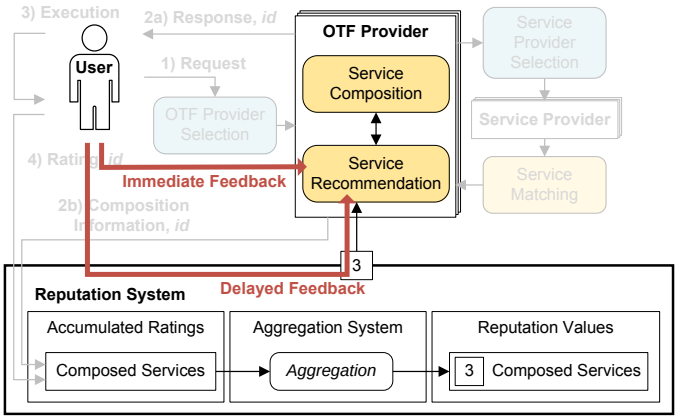


Figure 9. Overview: Service Composition and Recommendation

fuzzy sets can be determined based on different heuristics. Here, we only described one possibility. In general, choosing an appropriate heuristic means dealing with the trade-off between precision and recall: the steeper the transitions from membership to non-membership and vise versa, the less false negatives can be expected, while the less steep the transitions, the less false positives can be expected.

Another possibility for configuration is the aggregation function. Here, we described an averaging aggregation method. However, also maximizing or minimizing methods or more complex, e.g., hierarchical methods [38], are possible.

VIII. Reputation for Composed Services

This section focuses on the internal processes of a single OTF provider while taking user feedback for composed services into account (see Figure 9). In this context, the service composition component realizes a sequential service composition approach (cf. Section VIII-A). The service recommendation component, in turn, realizes a learning approach in order to improve the quality of composed services over time (cf. Section VIII-B). In this section, quality of composed services corresponds to how good a composed service satisfies a desired functionality. Assuming that alternative realizations for a desired functionality exist, the alternative that approximates the desired functionality the best is interpreted as the composed service with the highest quality.

After briefly introducing the fundamental techniques of both service composition approach and service recommendation approach, we present experimental results for demonstrating how delayed feedback in terms of reputation values influence the learning behavior, which originally incorporates immediate feedback. We use the image processing example introduced in the beginning of this paper as concrete use case for composing, executing and rating composed services. Incorporating reputation values for single services, however, is not considered in this section.

A. Sequential Service Composition Model

As already stated in Section III-A, we interpret automatic service composition as the sequential application of composition steps. In this section, we introduce our sequential composition model based on composition rules. However, our composition model represents only one possible realization of the service composition component depicted in Figure 9.

A composition rule compactly defines a formally correct modification during the service composition process. The syntax of composition rules is identical to the syntax of production rules, which contain *non-terminal* symbols and *terminal* symbols [39]. In our context, non-terminal symbols correspond to functional parts of the composed service that still have to be realized. Terminal symbols correspond to concrete services. For example, the composition rules $X \rightarrow s_1 Y \mid s_2$ define that a required functionality X can be composed by a service s_1 and a required functionality Y. Non-terminal Y, in turn, has to be realized in a subsequent composition step. Alternatively, X can be directly realized in terms of service s_2.

For illustration purposes, let us consider the image processing example introduced in Section II. Let us assume that Figure 1a can be transformed into Figure 1b by a sequence of two post-processing services. In fact, four different steps are actually necessary to achieve the desired result: Modification of contrast, brightness, saturation, and sharpness. However, for reasons of comprehensibility and without loss of generality, we consider only two processing steps Y and Z to be necessary in order to achieve the desired functionality X. The chronological order, in which both steps are applied cannot be neglected, but is relevant due to dependencies between processing results. In short, YZ may produce different results then ZY. As a consequence, we write $X \rightarrow YZ \mid ZY$. Let us assume, that each processing step can be performed by two different services, resulting in four services: s_1, s_2, s_3, and s_4. Services s_1 and s_2 as well as services s_3 and s_4 implement similar functionality and are equivalent with respect to their formal specifications. Required functionality Y can be realized by either s_1 or s_2, while Z can be realized by either s_3 or s_4. In short, we write $Y \rightarrow s_1 \mid s_2$ and $Z \rightarrow s_3 \mid s_4$.

Figure 10a depicts the complete list of composition rules mentioned heretofore. Each rule is assigned a unique identifier $\widehat{r}_i$. A derivation of rules corresponds to composing a solution for desired functionality X. For example, the derivation

$$X \xrightarrow{\widehat{r}_2} ZY \xrightarrow{\widehat{r}_3} Zs_1 \xrightarrow{\widehat{r}_5} s_3 s_1$$

composes the solution $s_3 s_1$ by successively applying rules $\widehat{r}_2$, $\widehat{r}_3$, and $\widehat{r}_5$. In fact, this sequence of single decision-making steps is similar to workflow composition. First of all, an abstract workflow has to be selected by either applying rule $\widehat{r}_1$ or rule $\widehat{r}_2$. Afterwards, the functional placeholders Y and Z have to be realized by choosing between rules $\widehat{r}_3$, $\widehat{r}_4$, and rules $\widehat{r}_5$, $\widehat{r}_6$, respectively.

According to Rao and Su [40], the second major realm of approaches besides workflow management that have emerged in order to tackle the service composition problem is Artificial Intelligence (AI) planning. From the AI planning perspective, service composition can be interpreted as a search problem in a state transition system [41]. Figure 11a depicts the state space for our example. In the most general sense, a state s_i corresponds to a set of conditions that hold as long as state s_i is occupied. Actions (arrows), in turn, correspond to services. Applying a service to a state s_i corresponds to changing the

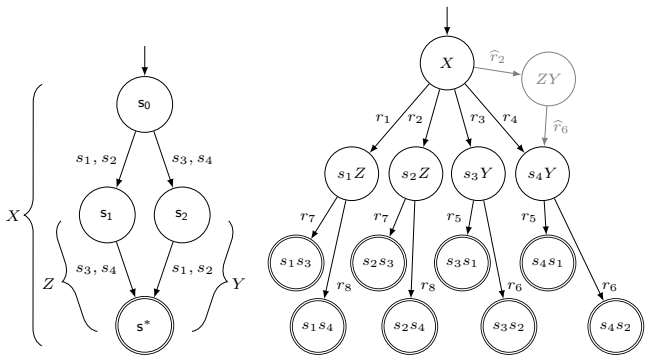
	$\widehat{r}_1$	$\widehat{r}_2$
X	YZ	ZY
	$\widehat{r}_3$	$\widehat{r}_4$
Y	s_1	s_2
	$\widehat{r}_5$	$\widehat{r}_6$
Z	s_3	s_4

(a)

	r_1	r_2	r_3	r_4
X	s_1Z	s_2Z	s_3Y	s_4Y
	r_5	r_6		
Y	s_1	s_2		
	r_7	r_8		
Z	s_3	s_4		

(b)

Figure 10. Composition rules (a) according to workflow composition and (b) according to forward search from AI planning.



(a) Planning State Space (b) Composition Rule State Space

Figure 11. Relationship between planning based composition and sequential application of composition rules.

conditions encoded in state $\mathsf{s_i}$ [42]. A desired functionality is usually specified in terms of pre- and postconditions. The preconditions correspond to the initial state $\mathsf{s_0}$, while the post-conditions correspond to the goal state $\mathsf{s_*}$. Search algorithms are then applied to find a sequence of services that transforms $\mathsf{s_0}$ into $\mathsf{s_*}$.

The planning based composition model can be gradually transformed into our composition model during the search process [43]. In terms of forward search, a non-terminal symbol N represents the unresolved path from the currently occupied state $\mathsf{s_i}$ to the goal state $\mathsf{s_*}$; we write $N = \overrightarrow{\mathsf{s_i s_*}}$. For example, when the forward search enters state $\mathsf{s_0}$, non terminal symbol $X = \overrightarrow{\mathsf{s_0 s_*}}$ is constructed. If the forward search then proceeds to state $\mathsf{s_2}$ by applying service s_3, non-terminal symbol $Y = \overrightarrow{\mathsf{s_2 s_*}}$ and composition rule $r_3 : X \rightarrow s_3 Y$ are constructed. If the search process subsequently applies service s_1 and proceeds to state $\mathsf{s_*}$, rule $r_5 : Y \rightarrow s_1$ is constructed. Figure 11b shows the complete state space based on our composition model for our example. The associated composition rules are listed in Figure 10b.

There are two major benefits when using our composition model. First, our model facilitates a unified composition process. Both workflow composition and AI planning can be modeled in terms of composition rules and can be combined in the same state space. Rule $\widehat{r}_3$, e.g., is equivalent to rule r_5. The sequential application of rules $\widehat{r}_2$ and $\widehat{r}_6$, in turn, is combined in rule r_4 (cf. Figure 11b). Second, the underlying decision-making process in our model satisfies the Markov property: Decisions do not depend on history but are memoryless. All relevant information is encoded in a single state in terms of the current composition structure. Whenever a decision between alternative composition rules has to be made, decision-making bases on the heretofore composed solution. Choosing between alternatives is up to the applied RL approach.

B. Improving Quality by Means of Learning

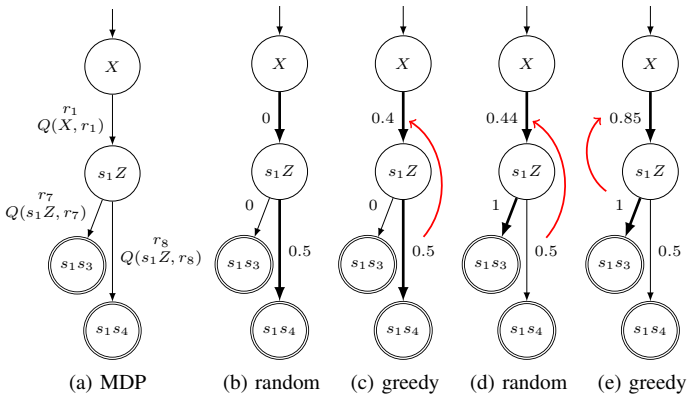
The recommendation component (cf. Figure 9) incorporates RL techniques for improving the quality of composed services over time. In general, RL addresses the problem faced by an autonomous agent that must learn to reach a goal through sequential trial-and-error interactions. Depending on its actions, an agent receives a reward value that is incorporated into the decision-making processes in order to adjust the selection of actions in the future. In case of episodic RL, an agent is not learning continuously but periodically in terms of episodes. An episode defines the period between initial state and final state.

In our context, an episode covers an entire composition process. After each composition process, a user gives feedback by rating the quality of the *execution result*. In terms of our example, a user rates his satisfaction regarding the result image that is produced by the composed service. To roughly simulate user ratings in our experiments, the original image (Figure 1a) is automatically processed by the composed service. The resulting image is then compared with the manually produced desired image (Figure 1b). Based on the normalized distance value $d \in [0, 1]$ of the original image and the desired image, the user rating $r = 1 - d$ is computed. The higher the rating, the better the execution result and, consequently, the higher the quality of the composed solution. The user rating is subsequently incorporated as a final reward into the sequential composition process to improve the selection of alternative composition steps during future composition processes.

Technically, the sequential composition process as described in the previous section is modeled as Markov Decision Process (MDP) [44]. Figure 12a shows a snippet of the state space depicted in Figure 11b. The annotated quality values $Q(\mathsf{s}, r)$ are an estimation of how good it is to apply a composition rule r when currently occupying state s. The higher these so-called Q-values, the more promising the corresponding actions for the composition task. In our work, we apply Q-Learning to adjust the Q-values over time based on user feedback [45]. Q-Learning is a model-free RL algorithm that directly approximates Q-values by means of its update function

$$Q(\mathsf{s}_t, r_t) \leftarrow Q(\mathsf{s}_t, r_t) + \alpha \left[\gamma \max_r [Q(\mathsf{s}_{t+1}, r)] - Q(\mathsf{s}_t, r_t) \right], \quad (1)$$

with current state s_t, next state s_{t+1}, current composition rule r_t, next composition rule r_{t+1}, discount factor γ, and learning rate α. In addition to the update function, we incoporate an ϵ-greedy action selection strategy. Action selection strategies are crucial in order to balance between exploitation of already learned knowledge and exploring new and probably better states. In case of ϵ-greedy, composition rules are greedily selected with probability $1 - \epsilon$: the composition rule with the highest Q-value is selected (exploitation phase). With probability ϵ, however, a composition rule is randomly selected (exploration phase). For our experiments, we chose a commonly used setting; that is, $\gamma = 0.9$, $\alpha = 0.9$, and $\epsilon = 0.1$.



(a) MDP (b) random (c) greedy (d) random (e) greedy

Figure 12. Demonstration of Q-Learning.

Figures 12b-12e illustrate the actual learning process. Each figure shows state space and associated Q-values *after* a composition process was completed and a user rating was incorporated as final reward. Thick arrows indicate the chosen path from initial state to final state. Q-values $Q(X, r_1)$, $Q(s_1Z, r_7)$, and $Q(s_1Z, r_8)$ are initialized with 0.

Figure 12b: Service s_1s_4 was composed and executed. During the composition process, composition rule r_8 was chosen randomly. The execution result was rated with value 0.5. The rating value was immediately integrated as final reward by adjusting Q-value $Q(s_1Z, r_8)$. Note: Final reward is always incorporated unmodified and replaces the Q-value of the lastly applied composition rule.

Figure 12c: The composition process again produced composed service s_1s_4. Composition rule r_8, however, was not selected randomly, but greedily based on Q-value $Q(s_1Z, r_8)$ that was modified in the previous composition process. After selecting r_8 and before transitioning to state s_1s_4, update rule (1) is applied to adjust Q-value $Q(X, r_1)$. Q-value $Q(s_1Z, r_8)$, however, is not changed again, since it is equivalent to the rating result, which is the same as before.

Figure 12d: Composition rule r_7 was randomly selected during the composition process. Executing composed service s_1s_4 results in an image that is identical to the desired result image. Hence, the rating value is 1. Q-value $Q(s_1Z, r_7)$ is immediately updated. During the composition process, however, this value was not yet available. Due to the max operator, Q-value $Q(X, r_1)$ was again updated based on Q-value $Q(s_1Z, r_8)$.

Figure 12e: The composition process operated in a greedy manner again. Furthermore, Q-value $Q(X, r_1)$ significantly increased, since it was updated based on Q-value $Q(s_1Z, r_7)$ this time.

By consecutively applying the update rule when moving through the state space and by continually incorporating ratings of consecutive composition processes, ratings are propagated throughout the state space and Q-values finally converge. Generally speaking, the overall composition process adapts its composition strategy to produce a composed service that approximates the desired functionality that is implicitly determined by the feedback.

We conducted two initial experiments to obtain reference results before investigating the impact of reputation information in the subsequent experiments. The objective in each experiment was to identify the composed service that reproduces the desired image shown in Figure 1b at best. Four different image processing operations had to be applied in order to appropriately modify contrast, brightness, color, and sharpness, respectively. For each operation, six different services with slightly different functionality were provided. The chronological order of the four operations was not defined in advance. To obtain sound results, each experiment comprised 3000 consecutive composition processes and was repeated 100 times. For each experiment, the ratings per composition process were plotted in terms of mean value and 95% confidence interval of all 100 independent runs. For reasons of clarity, the resulting plots were additionally smoothed.

Figure 13 depicts the results of the two initial experiments. In the first experiment (green plot), no feedback was provided at all. Decisions between alternative composition steps were made only randomly. As a consequence, the rating values do not increase over time but remain almost the same. The results of the first experiment serve as worst case. In the second experiment (black plot), immediate feedback was incorporated as described above. Rating values were directly integrated into the composition process as soon as they were available. The impact of learning is clearly visible. The period, in which the rating values increased, indicate the phase, in which Q-values kept changing during the experiment – the actual learning period. After approximately 1200 composition processes, the Q-values converged. The optimal value 1 was not achieved due to the ϵ-greedy action selection strategy: Composition steps were still selected randomly once in a while. The results of the second experiment serve as best case for the Q-Learning setting in our example.

C. OTF Provider-related Reputation Values

Considering our proposed reputation system, user ratings are accumulated and aggregated into reputation values to ensure preservation of privacy (cf. Section V-C). Furthermore, we consider reputation information about composed services to be OTF provider-related in order to strictly enforce preservation of OTF providers' business secrets. To evaluate the impact of OTF provider-related reputation values on the learning process, we delayed the incorporation of our simulated user ratings.

Usually, when dealing with markets of composed services,

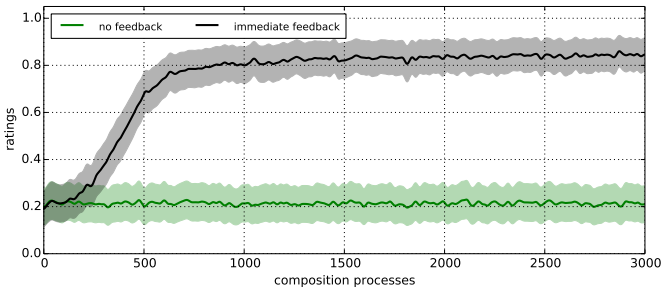


Figure 13. No learning (green) vs. learning (black).

users have individual preferences and consequently deliver different ratings for the same composed solution. By analyzing the different ratings, e.g., a new learning context can be identified and an independent learning process with a different learning objective can be initialized. To cope with the privacy preservation mechanisms, an aggregation function that still allows these steps has to be found. In our next experiment, however, we assumed that ratings are identical for identical composed services in order to analyze the mere influence of delayed reward. We conducted the experiment with the same settings as in the experiment with immediate feedback. However, we delayed the incorporation of ratings until a second rating for the same composed service was available. Figure 14 shows the results (blue plot) in comparison to the best and the worst case, respectively. The impact is enormous. After 3000 composition processes, the results of the best case were still not achieved. Nevertheless, a learning behavior is still clearly visible.

A possible way to improve the learning behavior is to change the learning process itself, including the update rule as well as the action selection strategy. For example, the update rule could be applied more often in order to increase the propagation speed of Q-values within the MDP. Modifying the learning process, however, is beyond the scope of this paper, but needs a more thorough investigation. Another common mean is to provide more samples, from which the process can learn. In our context, a sample comprises a composed service and its corresponding rating value. If more samples were available, reputation values could be provided earlier and the impact of the delay would be weakened. Regarding our OTF Computing market, OTF providers might establish a cooperation in order to share business secrets (the ratings for services they composed). The reputation system can be extended to accumulate and aggregate ratings for composed services that belong to a group of OTF providers. A new challenge that emerges is the asynchronous provision of new feedback. Until now, we assumed that feedback is incorporated synchronously – after a composition process has finished. However, new ratings from other OTF providers may be available anytime.

D. Publicly Accessible Reputation Values

In the previous section, we assumed that the complete composition information is stored in the reputation system in order to unambiguously identify and assign ratings to identical compositions. An alternative approach to increase the number of samples is to make rating values for composed services publicly accessible for every OTF provider. To not reveal all business secrets of the OTF providers, information about composed services that is stored in the reputation system must be abstracted. As a consequence, user ratings for different composed services most likely end up in the same accumulation process. When, for example, only information about the utilized services is considered, but not information such as data and control flow, ratings for composed services that contain the same set of services are mixed up. We call this effect ambiguous feedback.

To investigate the impact of ambiguous feedback, we conducted three additional experiments with different delays. During these experiments, composed services that contained the same set of services were considered to be identical. As a consequence, more ratings for a composed service were accumulated over time. The ratings for actually identical composed services, however, were not identical anymore. For example, the ratings for the composed service $s_1 s_2 s_3 s_4$ were accumulated together with the ratings for the composed service $s_4 s_3 s_2 s_1$. According to the defined delay, we aggregated the latest ratings in terms of their average. This value was then incorporated as ambiguous feedback into the learning process.

Figure 15 compares the results of the previous experiment and the first experiment with ambiguous feedback. In both experiments, reputation values were not provided until two rating values were available. At first view, the results were quite unexpected. The learning process with delayed, ambiguous feedback (orange plot) converged significantly faster than the learning process that incorporated just delayed feedback (blue plot). In fact, the experiment with ambiguous feedback even shows a very similar learning curve like the best case experiment with immediate feedback (black plot); shifted to the right by a several hundred composition processes. The reason for the outcome of this experiment is most likely the special characteristics of our chosen image processing experiment. The chronological order, in which the provided services were executed in order to achieve the chosen goal was not as important as assumed. As a consequence, the ratings for the set of services that were utilized in a composed service tended to be very similar. Although still delayed, feedback is accumulated faster and not as ambiguous as assumed.

Figure 16 compares the results of all three experiments with ambiguous feedback. The results of the experiment with

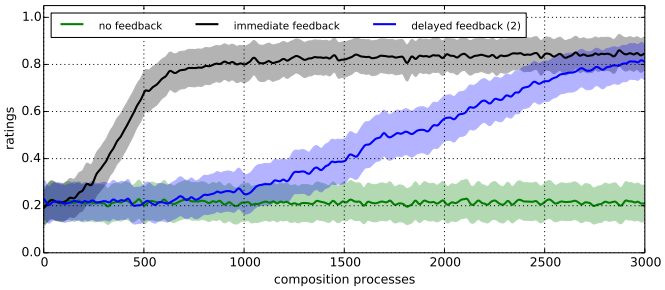


Figure 14. No learning (green), learning with immediate feedback (black), and learning with delayed feedback (blue) until two ratings were available.

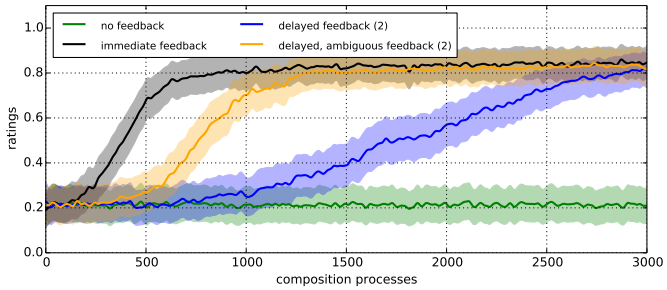


Figure 15. Delayed feedback (blue) based on two OTF provider related ratings and ambiguous feedback (orange) based on two publicly accessible ratings.

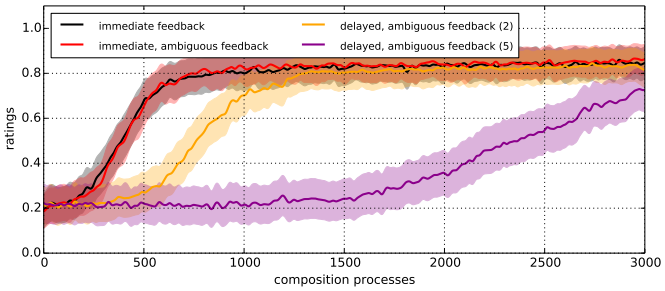


Figure 16. Comparison of all experiments with ambiguous feedback.

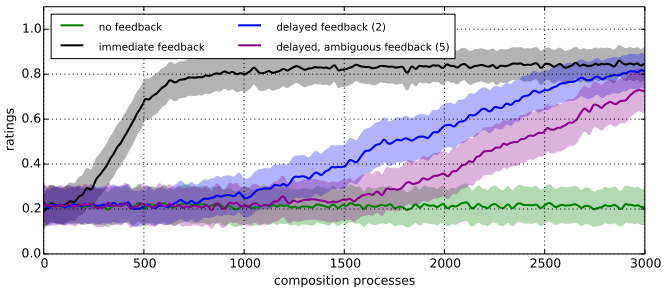


Figure 17. Comparison of delayed feedback (blue) based on two OTF provider-related rating values and ambiguous feedback (magenta) based on five publicly accessible rating values.

immediate, ambiguous feedback clearly show that mixing up ratings for composed services that are considered to be identical as long as they include the same set of services has no significant negative effect at all. The learning curves of both the experiment with immediate feedback and the experiment with immediate, ambiguous feedback do not significantly differ. Even the results of the last experiment (magenta plot), in which reputation values are not accessible until 5 rating values are available seem to be acceptable when directly comparing them to the delayed but OTF provider-related ratings (blue plot), as shown in Figure 17.

E. Remarks

In order to focus on the influence of reputation information under optimal conditions, we only considered a static context during the experiments. Neither the services available on the market, nor the specific user request changed. If we assume a MDP with a finite state space, Q-Learning is theoretically able to identify the optimal policy — as long as states are visited infinitely often [46]. The learning speed in terms of convergence rate can be improved by implementing a more sophisticated action selection strategy such as a dynamic ϵ-greedy strategy, where ϵ is adjusted based on temporal information.

In our case, however, we deal with a theoretically infinite state space: A composed service can be of arbitrarily length. In cases with low complexity such as our specific image processing example, the optimal solution may still be identified or at least be well approximated, as the experimental results have shown. In more complex scenarios, however, the state explosion problem inevitably emerges. A possible solution to deal with this problem is a dynamic state space approach. Similar states are merged into a single abstract state, while abstract states are split up if necessary. As a consequence, the amount of states can be restricted. Furthermore, in case of an abstract state, Q-values do not apply to a single concrete state, but to a set of states. This generalization effect might also be exploited to improve the learning speed [13].

In the long run, OTF Computing intends to consider dynamic market environments. Services may enter or leave a market anytime. In this case, finding the optimal solution is hardly possible. In fact, the Q-Learning approach will most likely only converge temporarily. However, finding the optimal solution (best case) is not even necessary to provide users a composed service that is better than a randomly selected solution. Users already benefit from less optimal solutions that are identified during the learning phase.

IX. Economic Decisions including Reputation

This section demonstrates how the information provided by the reputation system enters into the economic decision problems of users, OTF providers and services providers and how this influences the interaction in the OTF Computing market. Users and OTF providers interact when a user sends his request to an OTF provider (*Provider Selection*). OTF providers interact with service providers to purchase elementary services for a user's request (*Service Provider Selection*). This is shown in Figure 18. We suppose here that users and service providers do not directly communicate with each other and every interaction takes place via an OTF provider. Each of the market participants faces a different economic optimization problem. The primary goal of a user is to find the most preferred composed service(s) at an appropriate price whereas OTF and service providers are interested to maximize their profits by trading composed or single services.

An important feature of the OTF Computing market are information asymmetries between the market participants on qualitative characteristics of services and composed services.

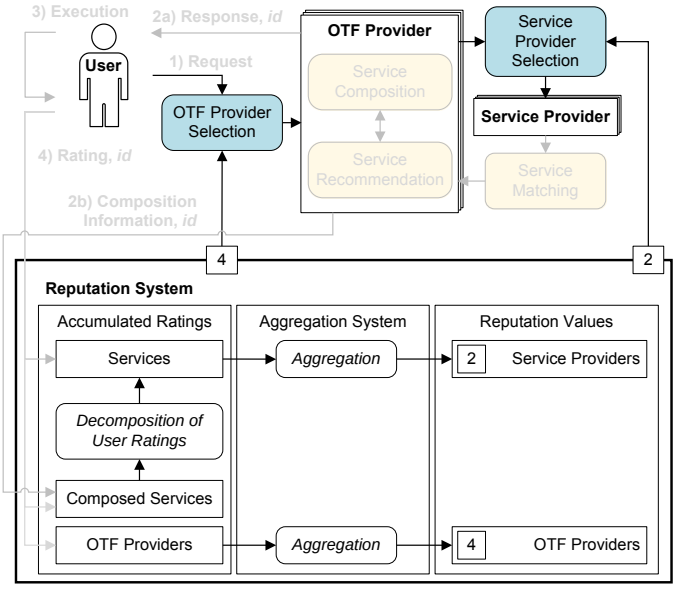


Figure 18. Overview: Economics

Therefore, a reputation system is essential to reduce these information asymmetries and to keep track of past behavior related to these unobservable characteristics. The information provided by the reputation system influences strategic decisions of users, OTF providers, and service providers. More precisely, the reputation information has an impact on the users' demands for composed services and, thus, also on the OTF providers' demands for elementary services. We model market prices depending on reputation values by analyzing both the market participants' decision problems and their interaction in the OTF Computing market.

For the rest of this section, suppose there are $N = 1, ..., n$ OTF providers typically indexed by i and $M = 1, ..., m$ service providers typically indexed by j. In addition, we assume for the notational simplicity that each service provider provides exactly one elementary service, which is available at different quality levels.

A. Economic Decisions of Users

The economic decision problem of a *user* is to formulate his request such that he gets the best composed service he can afford. The user's preferences are defined on a decision set of available composed services $S = \cup_{i \in N} \left\{ S_1^i, S_2^i, S_3^i, ... \right\}$. Hereby, the "available services" may be interpreted user-specific being those services that a user is aware of and considers in his decision problem. This need not necessarily be all composed services in the market. The user's preferences may also be expressed by weighting different characteristics of composed services. In this case, the distance between two composed services is used to define the user's preferences. If every two composed services from the decision set S can be compared with each other (*completeness*) and if S_1^i is preferred over $S_2^{i'}$ and $S_2^{i'}$ is preferred over $S_3^{i''}$ implies that S_1^i is preferred over $S_3^{i''}$ (*transitivity*) for $i, i', i'' \in N$, then the user's preferences are called rational [47, p. 42]. From a user's point of view there may exist exactly one ideal composed service (single peaked preferences) or his preferences may be monotonically increasing or decreasing for certain characteristics. Often utility functions describing the user's preferences are used to formally analyze economic decisions. In the OTF Computing market the user's utility function

$$u : S \to \mathbb{R} \tag{2}$$

assigns to a composed services $S_\ell^i \in S$ a valuation $u\left(S_\ell^i\right)$.

Within the OTF Computing process, the information provided by the reputation system about OTF providers influences the users' purchase decisions (*Provider Selection* in Figure 18). A user only selects those OTF providers that are reliable to properly answer his request. Therefore, taking the reputation information into account influences the user's evaluation of a composed service and may have the effect that certain alternatives are no longer acceptable as the reputation values are not sufficiently high. Let R_{OTF} and R_{SP} be the sets of possible reputation values of OTF and service providers. The user's utility function including the OTF provider's reputation value is given by

$$u : S \times R_{\mathrm{OTF}} \to \mathbb{R}. \tag{3}$$

Moreover, market prices in the OTF Computing market can be defined depending on the current reputation values. The user's decision problem is to buy a composed service that maximizes his utility for a fixed monetary budget that he is willing or able to spent. Therefore, given the OTF providers' prices $\left(p_{S_\ell^i}\left(r_{\mathrm{OTF}_i}\right)\right)_{S_\ell^i \in S}$, that depend on his current reputation value $r_{\mathrm{OTF}_i} \in R_{\mathrm{OTF}}$, a user selects those composed services that maximize the valuation minus the price for a composed service over the set of available composed services $S_\ell^i \in S$, formally the user's objective is

$$\max_{S_\ell^i \in S} \quad u\left(S_\ell^i, r_{\mathrm{OTF}_i}\right) - p_{S_\ell^i}\left(r_{\mathrm{OTF}_i}\right). \tag{4}$$

If a user is willing to buy several units of a composed service, his utility function and the price may also depend on the quantity he buys.

Besides the price of a composed service, quality considerations play an important role when a user compares different composed services in the OTF Computing market. If composed services are available at multiple quality levels, the user's decision problem can be explicitly extended by a quality dimension. Let Q be the set of possible quality levels and let Q_i denote the promised quality of composed service S_ℓ^i. Then, the user's utility function is

$$u : S \times R_{\mathrm{OTF}} \times Q \to \mathbb{R}. \tag{5}$$

The user's valuation of service S_ℓ^i in promised quality Q_i is $u\left(S_\ell^i, r_{\mathrm{OTF}_i}, Q_i\right)$ and therefore a user takes into account that each composed service may be available in different quality levels. A seminal contribution for the interplay of price and quality considerations in case of monopoly is [48], for instance. In the OTF Computing market, this quality is assumed to be unobservable before the transaction actually takes place. Therefore, the reputation information may help the user to find an appropriate OTF provider.

Beyond incorporating quality and reputation considerations, a different challenge for a user may be not to reveal his entire preferences in order not to be exploited in the market. Therefore, when he interacts with an OTF provider, he has to decide how much information he is willing to communicate about his preferences. This is important when his request is posed as well as when feedback on the composed service is provided. The revelation of preferences and the according incentives from an economic perspective are closely related to technical issues of privacy protection (cf. challenge *Manipulation Resistance versus Privacy Protection* in Section X).

B. Economic Decisions of OTF Providers

An *OTF provider* establishes a link between service providers and users. Thus, an OTF provider has to simultaneously consider the users' demands in the market for composed services as well as the available elementary services in the input market. Typically, there is a huge number of users with heterogeneous preferences in the OTF Computing market. Thus, from an OTF provider's perspective, the total users' demand is an aggregation of the individual demands from all users in the market resulting from their optimal buying decisions.

The profit maximization problem of an OTF provider is two-fold: On the one hand, it is crucial for an OTF provider to understand the users' preferences, to deduce their willingness to pay and to sell his composed services at profit maximizing prices. On the other hand, an OTF provider has to consider the supply of available services including their prices, that may either be given in the market or need to be negotiated with the service providers. The OTF provider's challenge is to find a combination of elementary services that minimizes the OTF provider's input costs for composed services (*Service Provider Selection* in Figure 18) and that at the same time sufficiently satisfies the users' requests (*Provider Selection* in Figure 18). It may be profitable for an OTF provider not to offer the entire range of composed services, but to focus on particular groups of users. The payments finally received for composed services need to be such that the prices paid for the elementary services as well as the effort put into the service composition process can be compensated. Hereby, the prices are typically dependent on the current reputation values provided by the reputation system.

The decision problem of an OTF provider $i \in N$ can be formalized as follows. Assume that the current reputation values are given by $r_{\mathrm{OTF}_i} \in R_{\mathrm{OTF}}$ and $r_{\mathrm{SP}_j} \in R_{\mathrm{SP}}$ for all $j \in M$ and denote the vector of the service providers' reputation values by $r_{\mathrm{SP}} = \left(r_{\mathrm{SP}_j}\right)_{j \in M}$. Let S^i be the set of composed services OTF provider i is able to build. Suppose, the OTF provider uses the composition strategy k_ℓ^i chosen from the set of possible composition strategies K^i to produce a composed service $S_\ell^i \in S^i$. This requires the use of elementary services s_{ij} for $j \in M_{k_\ell^i} \subseteq M$. Technical details of the *Service Composition* and *Service Recommendation Process* can be found in Section VIII. Given the sales price $p_{S_\ell^i} : R_{\mathrm{OTF}} \to \mathbb{R}_+$ with $r_{\mathrm{OTF}_i} \mapsto p_{S_\ell^i}\left(r_{\mathrm{OTF}_i}\right)$, prices of the elementary services $p_{s_{ij}} : R_{\mathrm{SP}} \to \mathbb{R}_+$ with $r_{\mathrm{SP}_j} \mapsto p_{s_{ij}}\left(r_{\mathrm{SP}_j}\right)$ and costs of the service composition $c : S_{\mathrm{OTF}_i} \times K \to \mathbb{R}_+$ with $\left(S_\ell^i, k_\ell^i\right) \mapsto c\left(S_\ell^i, k_\ell^i\right)$ the decision problem of an OTF provider for a user's request is to choose a composed service $S_\ell^i \in S$ and a composition strategy $k_\ell^i \in K^i$ to maximize his profit. This is the price the OTF provider receives from the user minus the costs he has. Formally, OTF provider i's profit is

$$\pi_{\mathrm{OTF}_i} : S^i \times K \times R_{\mathrm{OTF}} \times \prod_{j \in M} R_{\mathrm{SP}} \to \mathbb{R} \tag{6}$$

with

$$\begin{aligned}\pi_{\mathrm{OTF}_i}&\left(S_\ell^i, k_\ell^i, r_{\mathrm{OTF}_i}, r_{\mathrm{SP}}\right)\\ &= p_{S_\ell^i}\left(r_{\mathrm{OTF}_i}\right) - c(S_\ell^i, k_\ell^i) - \sum_{j \in M_{k_\ell^i}} p_{s_{ij}}\left(r_{\mathrm{SP}_j}\right)\end{aligned} \tag{7}$$

and his objective is

$$\max_{S_\ell^i \in S^i,\, k_\ell^i \in K^i} \quad \pi_{\mathrm{OTF}_i}\left(S_\ell^i, k_\ell^i, r_{\mathrm{OTF}_i}, r_{\mathrm{SP}}\right). \tag{8}$$

If there are several heterogeneous users, then OTF provider i's profit maximization problem needs to be considered over all users, to which he sells his composed services.

Summing up, the reputation information provided for elementary as well as composed services influences the OTF provider's economic decision problem via market prices driven by the users' demands.

In addition, within the service provider selection process, an OTF provider has to keep in mind that after he delivered the composed service to a user, he will receive a rating indicating the user's satisfaction. If the user's expectations are not met and the rating is negative, then this influences the future sale opportunities of an OTF provider. The incorporation of future profits makes the economic decision problem of an OTF provider even more complex. The profit maximization needs to be considered over current and future sales. Hence, the OTF provider's profit should be described as a discounted sum of profits that are expected in the long run from repeated interaction in the market.

C. Economic Decisions of Service Providers

The *Service Providers'* main challenge is to offer their elementary services such that an OTF provider decides to use these services for a composed service (*Service Provider Selection* in Figure 18). From a service provider's point of view, the difficulty is to provide services such that they match with those of other service providers technically as well as qualitatively. For a detailed description of the *Service Matching Process*, we refer to Section VII. As there is a huge number of elementary services available in the market, the service providers face an intensive competitive pressure. However, the services of different service providers are typically not perfectly exchangeable and, consequently, the OTF provider's decision for an elementary service depends on the availability of complementary services used for a composed service.

A service provider's profit in the OTF Computing market consists of the price he receives for his elementary service minus the costs to produce it. An economic decision of a service provider is the quality of the elementary service he delivers to an OTF provider. Suppose the set of possible quality levels is given by Q_{SP}. The profit of service provider $j \in M$ is

$$\pi_{\mathrm{SP_j}} : Q_{\mathrm{SP}} \times R_{\mathrm{OTF}} \times R_{\mathrm{SP}} \to \mathbb{R} \tag{9}$$

with

$$\pi_{\mathrm{SP_j}}\left(q_{ij}, r_{\mathrm{OTF}_i}, r_{\mathrm{SP}}\right) = p_{s_{ij}}\left(r_{\mathrm{SP}_j}\right) - c(q_{ij}). \tag{10}$$

The service provider's decision problem is to choose a quality level $q_{ij} \in Q_{\mathrm{SP}}$ that maximizes his profits,

$$\max_{q_{ij} \in Q_{\mathrm{SP}}} \pi_{\mathrm{SP_j}}\left(q_{ij}, r_{\mathrm{OTF}_i}, r_{\mathrm{SP}}\right). \tag{11}$$

Similarly to the OTF providers, a service provider maximizes his long run profits from repeated interaction in the market. The quality choice of a service provider has an impact on the quality of the composed service. After the execution of the composed service the OTF provider receives a rating and this rating influences the reputation values of the services and service providers used for the composition. Thus, a negative rating for a composed service may have the effect that the price a service provider receives from an OTF provider decreases. Accordingly, a service provider has to take the long run consequences of his short run decisions into account for his profit maximization.

D. Interaction in the OTF Computing Market

The interaction in the OTF Computing market can be formally described by means of game theoretic models. We start to explain the short run interaction for given current reputation values $r_{\mathrm{OTF}_i} \in R_{\mathrm{OTF}}$ and $r_{\mathrm{SP}_j} \in R_{\mathrm{SP}}$ for all $j \in M$.

If the short run decisions of users, OTF provider, and service providers are considered to be simultaneous, a *non-cooperative normal form game* can be used to describe the short run interaction. The players are users, OTF providers, and service providers. The strategic decision of a user is the selection of an OTF provider. An OTF provider selects the composed services he offers and the according composition strategies, while a service provider chooses the quality of the elementary services he is asked to deliver to the OTF provider. The payoffs of the market participants are the utility for a user and the profits for OTF and service providers derived in the previous subsections.

A well-known solution concept for non-cooperative normal form games is the *Nash equilibrium* [49]. In a Nash equilibrium, no market participant is able to increase his payoff by deviating from his strategic choice given the others' strategic choices. This means that in a Nash equilibrium

- a user cannot increase his utility by choosing a different OTF provider given the OTF providers' choices on composed services and composition strategy and given the service providers' quality choices,
- an OTF provider cannot increase his profit by changing the composed service or the composition strategy given the user's selection of an OTF provider and given the service providers' quality decisions,
- and a service provider cannot increase his profit by delivering his elementary service in a different quality given the user's selection of an OTF provider and given the OTF providers' choices on composed services and composition strategy.

We illustrate the normal form game by means of our running example. Consider the market of image processing services. Suppose there is one user who approaches an OTF provider to post-process his video. The OTF provider finds an elementary service that, however, from his point of view does not perfectly match the user's request (cf. Section VII). On the one hand, this matching result is imprecise as it is still dependent on the service provider's effort put into the performance of the video post-processing service. On the other hand, during the service composition process (cf. Section VIII), the OTF provider may put additional own effort to improve the quality of the video processing service. This example fits into the previously described economic framework as follows: There is one user, one OTF provider, and one service provider in the OTF Computing market. In addition, there is only one composed service available. As there is exactly one OTF provider who receives the user's request and there exists one composed service, the strategy set of the user is a singleton and can be described by $S = \{S_1\}$. The OTF provider has two different composition strategies $K = \{k_L, k_H\}$ (or different effort levels) and there are two different quality levels the service provider may choose for the elementary service $Q_{\mathrm{SP}} = \{q_L, q_H\}$. The qualities of the composed service are assumed to be $Q = \{LL, LH, HL, HH\}$. In addition, we take $u(S_1, LL) = 9$, $u(S_1, LH) = u(S_1, HL) = 10$, $u(S_1, HH) = 12$, $p_{S_1}(r_{\mathrm{OTF}_i}) = 8$, $c(S_1, k_L) = 1$, $c(S_1, k_H) = 2$, $p_{s_{11}}(r_{\mathrm{SP}_1}) = 4$, $c(q_L) = 2$, and $c(q_H) = 3$. The payoffs are shown in Figure 19. The entries of the payoff vectors are ordered such that the first value corresponds to the user's utility, the second to the OTF provider's profit and the third to the service provider's profit. The OTF provider chooses the row of the payoff matrix in Figure 19, the service provider the column and the user the matrix. The Nash equilibrium (S_1, k_L, q_L) with payoffs $(1, 3, 2)$ is marked in red.

S_1		q_L	q_H
	k_L	$(1, 3, 2)$	$(2, 3, 1)$
	k_H	$(2, 2, 2)$	$(4, 2, 1)$

Figure 19. Simultaneous Short Run Interaction

In this example, the user has no other option than to buy the composed service the OTF provider offers. Even if the user is interested in buying a composed service of high quality, OTF and service provider are willing to sell low quality as this is less expensive for them. In a more complicated scenario, the user may send a request to a different OTF provider or punish the OTF provider with a negative rating, which may lower the sale price in the long run.

Alternatively, the short run decisions of the market participants can be considered to be sequentially. In this case, a *non-cooperative extensive form game* should be used to describe the interaction in the OTF Computing market. The strategic choices are now sequential: First, the users select the OTF providers. Then, after receiving the users' requests, the OTF providers react and select composed services and composition strategies. Finally, the service providers choose the quality of the elementary services. The final payoffs are as previously described. A solution concept used for sequential decisions is the *subgame-perfect Nash equilibrium*, which is a refinement of the original Nash equilibrium for sequential interaction [50][51].

Figure 20 shows the previous example in an extensive form. The subgame perfect Nash equilibria can be obtained in our case by *backwards induction*. We first determine the best choices of the service provider who makes the last decision. Taking this decision as given, we determine the optimal choice of the OTF provider and finally the user decides. The optimal choices are market in red in Figure 20. The subgame perfect Nash equilibrium is (S_1, k_L, q_L) with payoffs $(1, 2, 3)$. Independently of the choice of the OTF provider, it is the best option for the service provider to produce an elementary service of low quality. The OTF provider now takes this choice of the service provider as given and deduces that his optimal action is to use composition strategy k_L.

Up to now, the prices in the OTF Computing market were assumed to be given depending on the reputation values of the OTF and service providers. There may exist a rule governing the evolution of prices depending on the strategic choices of

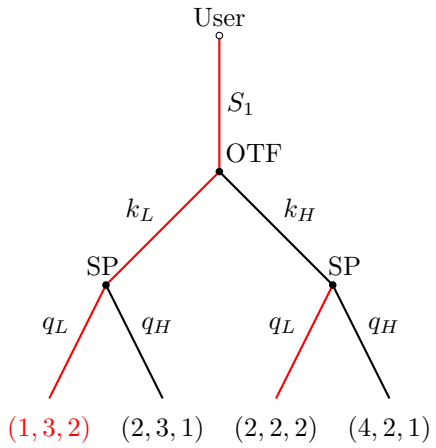


Figure 20. Sequential Short Run Interaction

the market participants and the rating finally received from the users.

X. Future Research Challenges

The introduction of a reputation system in the OTF Computing in Section V is conceptual and provides flexibility for further specifications. The research challenges resulting from the requirements imposed in Section IV have been highlighted in our previous contribution [1] and were investigated in more detail in Sections VII, VIII, and IX of this contribution. This section is devoted to further describe the trade-offs and challenges that are planed to be analyzed more intensively in future work.

Fuzzy Matching of Reputation Values: In Section VII, we analyzed in more detail how reputation should be matched. Since the reputation of a service is not an objective measure, such as signatures or protocols, uncertainty might be introduced into the matching process. For example, as noted in our fuzzy matching survey [12], the user stating the request might tolerate variations (e.g., "I want a service with *approximately* five stars"), or the request might include requirements, for which the corresponding information on the provider side do not exist yet (e.g., there has not been much feedback yet because the service is new on the market and thus the reputation is unclear). Section VII illustrated our fuzzy matching approach for a specific form of reputation values. However, there are still open issues in this area. For example, restrictions based on expert ratings (*R7*) could be introduced. Furthermore, we plan to quantify fuzziness based on necessity and possibility measures [52] in order to give even more feedback on the matching result.

Efficiency in Learning versus Privacy Protection: In Section VIII, we explained how a delayed feedback affects the convergence behavior of the learning process. The reinforcement learning approach, which is used to improve the quality of composed services, originally incorporates feedback immediately after a composition process. If the feedback is absent, the learning process is hampered. However, for reasons of privacy protection, no direct feedback is given to any party. Only an aggregated value of the accumulation of several individual ratings (feedback) is provided, as described in Section V. How the results of ambiguous feedback have to be interpreted in a more general context beyond the one described in Section VIII is still an open question. The influence of the execution order has to be thoroughly investigated in a different experiment in order to estimate, in which context a learning process benefits and suffers from ambiguous feedback, respectively. The presented results, however, are promising and gave a clear direction for our ongoing research. Furthermore, a combination of OTF provider-related ratings and publicly accessible ratings might be a good mean to improve the ratio between unambiguous feedback and ambiguous feedback. For example, an OTF provider might incorporate OTF provider-related rating values when available. When not available, ambiguous feedback can be incorporated. To measure the degree of ambiguity in order to enable an OTF provider to decide whether to use such feedback or not can be provided by the reputation value in terms of common measures of dispersion such as standard deviations or associated confidence intervals. In addition to these investigations, incorporating varying user ratings due to varying preferences is still an open challenge as well.

Economic Decisions: In Section IX, we described the economic decisions on the OTF Computing market and studied their interaction. Alternatively to an exogenously fixed rule of price evolution, the prices may also be strategically chosen by the OTF and service providers. Two possibilities are *price* (or *Bertrand*) and *quantity* (or *Cournot*) competition. In the first case, the providers compete by choosing profit maximizing prices and, in the second case, the providers compete by announcing prices such that the demanded quantities are profit maximizing. These two forms of competition are compared in [53], for example. The model in [53] allows for quality differences between the services and uses a parameter to describe substitutabilities or complementarities between the services. Within the model, the main observation is that for complementary services the providers' profits are higher if the providers compete in prices whereas for substitutable services with small quality differences quantity competition yields higher provider profits [53, Proposition 2]. However, if the services are substitutes and the quality differences are large, the provider with the quality advantage earns higher profits in case of price competition. An interesting direction for future research is to apply these models of strategic pricing to the OTF Computing market including a reputation system.

More complex pricing structures such as *two-part tariffs* with a fixed fee and usage depended price or *three-part tariffs* with an additionally included volume have been analyzed in [54] applied to a cloud computing context. The main observation is that in case of symmetric providers and homogeneous services the competitive pressure forces the equilibrium prices to decrease until they are equal to the providers' costs. Hence, for the OTF Computing market, on which typically both substitutable and complementary services are traded and pricing structures may be even more complex, further investigations are needed.

A different point that our simple example already indicates is that optimal decisions in the short run may not coincide with those in the long run when the impact of the reputation information in the market prices is taken into account. Therefore, the long run interaction in the market have to be further

investigated. The simultaneous or sequential short run game in the OTF Computing market has to be considered as repeated game or as a stochastic game [55]. Without a reputation system and from a contract design perspective, the long run decisions in the OTF Computing market are analyzed in [56]. It is shown, that the OTF provider's evaluation of future profits plays a crucial role for the long run equilibrium quality level in the market for composed services. If the OTF provider's future expectations are too pessimistic, then composed services of low quality are going to be offered in the OTF Computing market even when high quality is more efficient from a welfare perspective. In [57], simulation techniques for complex strategic decisions are applied to the OTF Computing market. However, further analyzes to investigate the long run decisions in the market for composed services and to understand the impact of reputation information on the market participants' strategic behavior are necessary. This is one main direction of our current and future research.

Benefit of Privacy Protection: As discussed in this paper and [1], the design of a privacy-preserving solution entails a multitude of trade-offs that need to be taken into account, e.g., the trade-off between privacy and learning mechanism efficiency. Thus, it needs to be investigated whether market participants are interested in implementing a privacy-preserving solution at all. We need to prove that privacy protection is a benefit of OTF Computing and that users rather use such a market than any other, which does not provide such strong privacy guarantees. Concerning the introduced reputation system, we want to examine whether users are more willingly providing ratings when their privacy is protected—which is not the case in any other state-of-the-art reputation system in use today.

Manipulation Resistance versus Privacy Protection: An important further issue is to obtain truthful user feedback. Ratings may be dishonest or randomly chosen [23]. So far we assumed that users have no incentives to strategically manipulate their feedback and moreover we supposed that feedback on a transaction is always provided. Truthful rating behavior is induced by *incentive compatible reputation mechanisms* [58] (and the references mentioned therein). To ensure privacy protection, several ratings need to be accumulated and aggregated. It has already been analyzed how the aggregation of ratings impacts the efficiency of a reputation mechanism [59] and how it influences incentives for truthful rating behavior [60]. An important next step now is to further understand the interplay of incentive compatibility and privacy protection. Therefore, a challenging question is whether and how it is possible to design reputation systems that induce truthful feedback and respect privacy protection.

XI. Conclusion

In the context of OTF Computing, we interpret reputation information as a signal for quality in markets of composed services. From an economic perspective, the buying decision of a user and the future sale opportunity of an OTF provider crucially depend on the current reputation value. For that reason, we proposed a conceptual design of a reputation system that collects information about experiences users make with composed services in transactions. We identified and described requirements for such a reputation system from a technical, and economic perspective, and presented research challenges that automatically emerge from conflicting objectives. In this regard, we focused on three aspects, which are among others crucial for markets of composed services: service matching, service composition and economic decisions of market participants (users, OTF providers, and service providers).

In case of service matching, we introduced our fuzzy matching approach that is able to cope with reputation values that are – in comparison to other non-functional properties such as performance or costs – not of objective nature. Nevertheless, there are still open challenges such as the integration of restrictions based on expert ratings. Furthermore, we intend to quantify fuzziness based on necessity and possibility measures in order to give even more feedback on the matching result.

The impact of reputation information for our RL based composition and recommendation approach has to be investigated in a more general context. For our image processing example with particular characteristics, however, we presented promising results. A major challenge in this context is the collection and processing of user ratings from different sources (e.g., OTF provider-related vs. publicly available) in order to increase the amount of learning samples in terms of reputation information.

From the economic point of view, we analyzed the individual decision problems of the market participants in the OTF Computing market as well as their interaction. Hereby, we explicitly included the information provided by the reputation system. Moreover, we outlined by means of a simple example how strategic decisions may influence the quality of composed services traded in equilibrium in the market. Some preliminary results to analyze the economic interaction in the OTF Computing market have already been obtained in previous and ongoing work as outlined in Section X. This can be seen as a first step for a comprehensive analysis of the impact of reputation on the OTF Computing market.

The contribution of this paper is not necessarily restricted to OTF Computing alone. Results of our work can also be adopted to other areas, in which reputation of combinable products is crucial. For our future work, our reputation system can be considered as an interface to bring together approaches and results from different (sub-)disciplines such as economics, software engineering, distributed systems, artificial intelligence, and security.

Acknowledgment

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901).

References

[1] S. Brangewitz, A. Jungmann, R. Petrlic, and M. C. Platenius, "Towards a flexible and privacy-preserving reputation system for markets of composed services," in Proceedings of the Sixth International Conferences on Advanced Service Computing (SERVICE COMPUTATION), 2014, pp. 49–57.

[2] R. Petrlic, A. Jungmann, M. C. Platenius, W. Schaefer, and C. Sorge, "Security and privacy challenges in on-the-fly computing," in Tagungsband der 4. Konferenz Software-Technologien und -Prozesse (STeP), 2014, pp. 131–142.

[3] "Collaborative Research Center 901 - On-The-Fly Computing," 2014, URL: http://sfb901.uni-paderborn.de [accessed: 2014-11-28].

[4] "Instagram," 2014, URL: http://www.instagram.com [accessed: 2014-11-28].

[5] F. Gómez Mármol and M. Q. Kuhnen, "Reputation-based web service orchestration in cloud computing: A survey," Concurrency and Computation: Practice and Experience, 2013.

[6] N. Hiratsuka, F. Ishikawa, and S. Honiden, "Service selection with combinational use of functionally-equivalent services," in Proceedings of the 18th IEEE International Conference on Web Services (ICWS), 2011, pp. 97–104.

[7] J. Peer, "Web service composition as ai planning - a survey," University of St. Gallen, Switzerland, Tech. Rep., 2005.

[8] P. Bartalos and M. Bieliková, "Semantic web service composition framework based on parallel processing," in Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC), 2009, pp. 495–498.

[9] P. Bartalos and M. Bieliková, "Automatic dynamic web service composition: A survey and problem formalization," Computing and Informatics, vol. 30, no. 4, 2011, pp. 793–827.

[10] M. Aiello, E. el Khoury, A. Lazovik, and P. Ratelband, "Optimal QoS-Aware Web Service Composition," in Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC), 2009, pp. 491–494.

[11] M. C. Platenius, "Fuzzy service matching in on-the-fly computing," in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE). ACM, 2013, pp. 715–718.

[12] M. C. Platenius, M. von Detten, S. Becker, W. Schäfer, and G. Engels, "A survey of fuzzy service matching approaches in the context of on-the-fly computing," in Proceedings of the 16th International ACM Sigsoft Symposium on Component-based Software Engineering (CBSE). ACM, 2013, pp. 143–152.

[13] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, Massachusetts: MIT Press, 1998.

[14] A. Jungmann and B. Kleinjohann, "Learning Recommendation System for Automated Service Composition," in Proceedings of the 10th IEEE International Conference on Services Computing (SCC), 2013, pp. 97–104.

[15] A. Jungmann, B. Kleinjohann, and L. Kleinjohann, "Learning service recommendations," Int. J. Business Process Integration and Management, vol. 6, no. 4, 2013, pp. 284–297.

[16] A. Van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE), 2001, pp. 249–262.

[17] Y. Wang and J. Vassileva, "A review on trust and reputation for web service selection," in 27th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2007, pp. 25–25.

[18] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in Proceedings of the IEEE Symposium on Security and Privacy (SP), 2008, pp. 111–125.

[19] R. Petrlic, S. Lutters, and C. Sorge, "Privacy-preserving reputation management," in Proceedings of the 29th Symposium On Applied Computing. ACM, 2014.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proceedings of the 17th international conference on Theory and application of cryptographic techniques. Springer, 1999, pp. 223–238.

[21] C. Shapiro, "Premiums for high quality products as returns to reputations," The Quarterly Journal of Economics, vol. 98, no. 4, 1983, pp. 659–680.

[22] H. Bar-Isaac and S. Tadelis, "Seller reputation," Foundations and Trends in Microeconomics, vol. 4, no. 4, 2008, pp. 273–351.

[23] E. Friedman, P. Resnick, and R. Sami, "Manipulation-resistant reputation systems," in Algorithmic Game Theory, Chapter 27. Cambridge University Press, 2007.

[24] C. Dellarocas, "Reputation mechanism design in online trading environments with pure moral hazard," Information Systems Research, vol. 16, no. 2, 2005, pp. 209–230.

[25] F. Gómez Mármol and G. Martinez Pérez, "Trust and reputation models comparison," Internet Research, vol. 21, no. 2, 2011, pp. 138–153.

[26] F. Kerschbaum, "A verifiable, centralized, coercion-free reputation system," in Proceedings of the 8th ACM workshop on Privacy in the electronic society (WPES), 2009, pp. 61–70.

[27] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, "Reputation systems for anonymous networks," in Proceedings of the 8th International Symposium on Privacy Enhancing Technologies. Springer, 2008, pp. 202–218.

[28] A. Shaikh Ali, S. Majithia, O. F. Rana, and D. W. Walker, "Reputation-based semantic service discovery," Concurrency and Computation: Practice and Experience, vol. 18, no. 8, 2006, pp. 817–826.

[29] M. R. Motallebi, F. Ishikawa, and S. Honiden, "Component trust for web service compositions," in AAAI Spring Symposium Series, 2012.

[30] Z. Malik and A. Bouguettaya, "Rateweb: Reputation assessment for trust establishment among web services," The VLDB Journal, vol. 18, no. 4, 2009, pp. 885–911.

[31] K. Huang, J. Yao, Y. Fan, W. Tan, S. Nepal, Y. Ni, and S. Chen, "Mirror, mirror, on the web, which is the most reputable service of them all?" in Service-Oriented Computing. Springer, 2013, pp. 343–357.

[32] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn, "Privacy-aware daas services composition," in Database and Expert Systems Applications, 2011, pp. 202–216.

[33] E. Costante, F. Paci, and N. Zannone, "Privacy-aware web service composition and ranking," in Proceedings of the 20th IEEE International Conference on Web Services (ICWS), 2013, pp. 131–138.

[34] M. C. Platenius, S. Arifulina, R. Petrlic, and W. Schäfer, "Matching of incomplete service specifications exemplified by privacy policy matching," in 4th International Workshop on Adaptive Services for the Future Internet. Springer, 2014, in press.

[35] L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, no. 3, 1965, pp. 338–353.

[36] H. Lam, F. H. F. Leung, and P. K.-S. Tam, "Stable and robust fuzzy control for uncertain nonlinear systems," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 30, no. 6, 2000, pp. 825–840.

[37] R.-E. Precup, M. L. Tomescu, and S. Preitl, "Fuzzy logic control system stability analysis based on lyapunovs direct method," International Journal of Computers, Communications & Control, vol. 4, no. 4, 2009, pp. 415–426.

[38] Y. Yi, T. Fober, and E. Hüllermeier, "Fuzzy operator trees for modeling rating functions," International Journal of Computational Intelligence and Applications, vol. 8, no. 04, 2009, pp. 413–428.

[39] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2009.

[40] J. Rao and X. Su, "A survey of automated web service composition methods," in Proceedings of the First International Conference on Semantic Web Services and Web Process Composition (SWSWPC). Springer-Verlag, 2005, pp. 43–54.

[41] M. Ghallab, D. Nau, and P. Traverso, Automated planning: theory & practice. San Francisco, CA, USA: Morgan Kaufmann, 2004.

[42] D. Doliwa, W. Horzelski, M. Jarocki, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, and A. Zbrzezny, "Planics - a web service composition toolset," Fundam. Inf., vol. 112, no. 1, 2011, pp. 47–71.

[43] A. Jungmann, F. Mohr, and B. Kleinjohann, "Combining automatic service composition with adaptive service recommendation for dynamic markets of services," in Proceedings of the IEEE 10th World Congress on Services (SERVICES), 2014, pp. 346–353.

[44] M. L. Puterman, Markov decision processes: Discrete stochastic dynamic programming. Hoboken, NJ, USA: Wiley-Interscience, 2005.

[45] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, 1992, pp. 279–292.

[46] T. M. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.

[47] A. Mas-Colell, M. D. Whinston, and J. R. Green, Microeconomic Theory. Oxford University Press, 1995.

[48] A. M. Spence, "Monopoly, quality, and regulation," The Bell Journal of Economics, vol. 6, no. 2, 1975, pp. 417–429.

[49] J. Nash, "Non-cooperative games," Annals of Mathematics, vol. 54, no. 2, 1951, pp. 286–295.

[50] R. Selten, "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit: Teil I: Bestimmung des dynamischen Preisgleichgewichts," Zeitschrift für die gesamte Staatswissenschaft, vol. 121, no. 2, 1965, pp. 301–324.

[51] R. Selten, "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit: Teil II: Eigenschaften des dynamischen Preisgleichgewichts," Zeitschrift für die gesamte Staatswissenschaft, vol. 121, no. 4, 1965, pp. 667–689.

[52] D. Dubois, H. Nguyen, and H. Prade, "Possibility theory, probability and fuzzy sets misunderstandings, bridges and gaps," in Fundamentals of Fuzzy Sets, ser. The Handbooks of Fuzzy Sets Series. Springer US, 2000, vol. 7, pp. 343–438.

[53] J. Häckner, "A note on price and quantity competition in differentiated oligopolies," Journal of Economic Theory, vol. 93, no. 2, 2000, pp. 233–239.

[54] J. Künsemöller, S. Brangewitz, H. Karl, and C.-J. Haake, "Provider competition in infrastructure-as-a-service," in Proceedings of the IEEE International Conference on Services Computing (SCC), 2014, pp. 203–210.

[55] G. J. Mailath and L. Samuelson, Repeated games and reputations: long-run relationships. Oxford university press, 2006.

[56] S. Brangewitz, C.-J. Haake, and J. Manegold, "Contract design for composed services in a cloud computing environment," in Proceedings of the 2nd International Workshop on Cloud Service Brokerage (CSB), 2014, in press.

[57] M. Feldotto and A. Skopalik, "A simulation framework for analyzing complex infinitely repeated games," in Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH). SciTePress, 2014, pp. 625–630.

[58] S. Phoomvuthisarn, "A survey study on reputation-based trust mechanisms in service-oriented computing," Journal of Information Science and Technology, vol. 2, no. 2, 2011, pp. 1–12.

[59] C. Dellarocas, "How often should reputation mechanisms update a trader's reputation profile?" Information Systems Research, vol. 17, no. 3, 2006, pp. 271–285.

[60] C. Aperjis and R. Johari, "Optimal windows for aggregating ratings in electronic marketplaces," Management Science, vol. 56, no. 5, 2010, pp. 864–880.