Reusability for Trust and Reputation Systems

Johannes Sänger, Günther Pernul

Reusability for trust and reputation systems

Johannes Sänger and Günther Pernul

Department of Information Systems
University of Regensburg
Regensburg, Germany
{johannes.saenger,guenther.pernul}@wiwi.uni-regensburg.de
http://www-ifs.uni-regensburg.de

Abstract. Reputation systems have been extensively explored in various disciplines and application areas. A problem in this context is that the computation engines applied by most reputation systems available are designed from scratch and rarely consider well established concepts and achievements made by others. Thus, approved models and promising approaches may get lost in the shuffle. In this work, we aim to foster reuse in respect of trust and reputation systems by providing a hierarchical component taxonomy of computation engines which serves as a natural framework for the design of new reputation systems. In order to assist the design process we, furthermore, provide a component repository that contains design knowledge on both a conceptual and an implementation level.

Keywords: trust, reputation, reusability, trust pattern

1 Introduction

In the last decade, trust and reputation have been extensively explored in various disciplines and application areas. Thereby, a wide range of metrics and computation methods for reputation-based trust has been proposed. While most common systems have been introduced in eCommerce, such as eBay's reputation system[1] that allows to rate sellers and buyers, considerable research has also been done in the context of peer-to-peer networks, mobile ad hoc networks, social networks or ensuring data accuracy, relevance and quality in several environments [1]. Computation methods applied range from simple arithmetic over statistical approaches up to graph-based models involving multiple factors such as context information, propagation or personal preferences. A general problem is that most of the new introduced trust and reputation models use computation methods that are designed from scratch and rely on one novel idea which could lead to better solutions [2]. Only a few authors built on proposals of others. Therefore, approved models and promising approaches may get lost in the shuffle.

In this work, we aim to encourage reuse in the development of reputation systems by providing a framework for creating reputation systems based on reusable

[1] http://www.ebay.com

components. Design approaches for reuse have been given much attention in the software engineering community. The research in trust and reputation systems could also profit from benefits like effective use of specialists, accelerated development and increased reliability. Toward this goal, we propose a *hierarchical taxonomy* for components of computation engines used in reputation systems. We, thereto, decompose the computation phase of common reputation models to derive single building blocks. The classification based on their functions serves as a natural framework for the design of new reputation systems. To facilitate the reuse of the identified components we, moreover, set up a *component repository* containing artifacts on both a conceptual and an implementation level. On the conceptual level, we describe each building block as a design pattern-like solution. On the implementation level, we provide already implemented components by means of web-services.

The rest of this paper is based on the design science research paradigm involving the guidelines for conducting design science research by Hevner et al. [3] and organized as follows: Firstly, we give an overview of the general problem context, the relevance and motivation of our work. We, thereby, identify the research gap and define the objectives of our research. In the following section, we introduce our hierarchical component taxonomy of computation engines used in reputation systems. Subsequently, we point out how our component repository is conceptually designed and implemented. Finally, we summarize the contribution and name our plans for future work.

2 Problem context and motivation

With the success of the Internet and the increasing distribution and connectivity, trust and reputation systems have become important artifacts to support decision making in network environments. To impart a common understanding, we firstly provide a definition of the notion of trust. At the same time, we explain the properties of trust that are important with regard to this work. Then, we point out how trust can be established applying computational trust models. Focusing an reputation-based trust, we explain how and why the research in reputation models could profit from reuse. We, thereby, identify the research gap and define the objectives of this work.

2.1 The notion of trust and its properties

The notion of trust is a topic that has been discussed in research for decades. Although it has been intensively examined in various fields, it still lacks a uniform, generally accepted definition. Reasons for this circumstance are the multifaceted terms trust is associated with like credibility, reliability or confidence as well as the multidimensionality of trust as an abstract concept that has a cognitive, an emotional and a behavioral dimension. As pointed out by [4], trust has been described as being structural in nature by sociologists while psychologists viewed trust as an interpersonal phenomenon. Economists, however, interpreted trust

as rational choice mechanism. The definition often cited in literature regarding trust and reputation online that is referred to as *reliability trust* was proposed by Gambetta in 1988 [5]:

"*Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.*"

Multiple authors furthermore include security and risk which can lead to more complex definitions. Anyway, it is generally agreed that trust is multifaceted and dependent on a variety of factors. Trust is *dynamical, context specific, subjective, propagative, non-transitive, composable* and *event sensitive* as Sherchan et al. [6] point out. These properties are important with respect to this work, since they form the basis for many applied computation techniques in trust and reputation systems described in section 3.2. Reusable components could extend current models by the ability to gradually include these properties.

2.2 Reputation-based trust

In the recent years, several trust models have been developed to establish trust. Thereby, two common ways can be distinguished, namely policy-based and reputation-based trust establishment [7]. Policy-based trust is often referred to as a *hard security mechanism* due to the exchange of hard evidence (e.g. credentials). Reputation-based trust, in contrast, is derived from the history of interactions. Hence, it can be seen as an estimation of trustworthiness (*soft security*). In this work, we focus on reputation-based trust. Reputation is defined as follows:

"*Reputation is what is generally said or believed about a person's or thing's character or standing.*" [8]

It is based on referrals, ratings or reviews from members of a community and can, therefore, be considered as a collective measure of trustworthiness [8]. Trustworthiness as a global value is objective. The trust an agent puts in someone or something as a combination of personal experience and referrals, however, is subjective.

2.3 Research gap: Design of reputation systems with reuse

It has been argued (e.g. by [2]) that most reputation-based trust models proposed in the academic community are built from scratch and do not rely on existing approaches. Only a few authors continue their research on the ideas of others. Thus, many approved models and promising thoughts go unregarded. The benefits of reuse, though, have been recognized in software engineering for years. However, there are only very few works that proposed single components to enhance existing approaches. Rehak et al. [9], for instance, introduced a generic mechanism that can be combined with existing trust models to extend their capabilities by efficiently modeling context. The benefits of such a component that can easily

be combined with existing systems are obvious. Nonetheless, research in trust and reputation still lacks in sound and accepted principles to foster reuse.

To gradually close this gap, we aim to provide a framework for the design of new reputation systems with reuse. As described above, we thereto propose a hierarchical component taxonomy of computation engines used in reputation systems. Based on this taxonomy, we set up a repository containing design knowledge on both a conceptual and an implementation level. The uniform and well-structured artifacts collected in this repository can be used by developers to select, understand and apply existing concepts on the one hand, as well as encourage researchers to provide novel components on a conceptual and an implementation level, on the other hand. In this way, the reuse of ideas, concepts and implemented components as well as the communication of reuse knowledge should be achieved.

3 A hierarchical component taxonomy for computation methods in reputation systems

To derive a taxonomy from existing models, our research includes two steps: (1) the analysis of the generic process of reputation systems and (2) the identification of logical components of the computation methods used in common trust and reputation models. A critical question is how to determine and classify single components. We thereto follow an approach to function-based component classification, where the taxonomy is derived from the functions identified components fulfill.

3.1 The generic process of reputation systems

The generic process of reputation systems, as depicted in Figure 1, can be divided into three steps: (1) *collection & preparation*, (2) *computation* and (3) *storage & communication*. Those steps were adapted from the three fundamental phases of reputation systems identified by [10] and [11]: feedback generation/collection, feedback aggregation and feedback distribution. Feedback aggregation as the central part of every trust and reputation system was furthermore divided into three process-steps *filtering*, *weighting* and *aggregation* taken together as computation. The context setting consists of a trustor who wants to build a trust relation toward a trustee by providing context and personalization parameters and receiving a trustee's reputation value.

Collection and preparation In the collection and preparation phase, the reputation system gleans information about the past behavior of a trustee and prepares it for subsequent computing. Although personal experience is the most reliable, it is often not sufficiently available or nonexistent. Therefore, data from other sources needs to be collected. These can be various, ranging from public or personal collections of data centrally stored to data requested from different

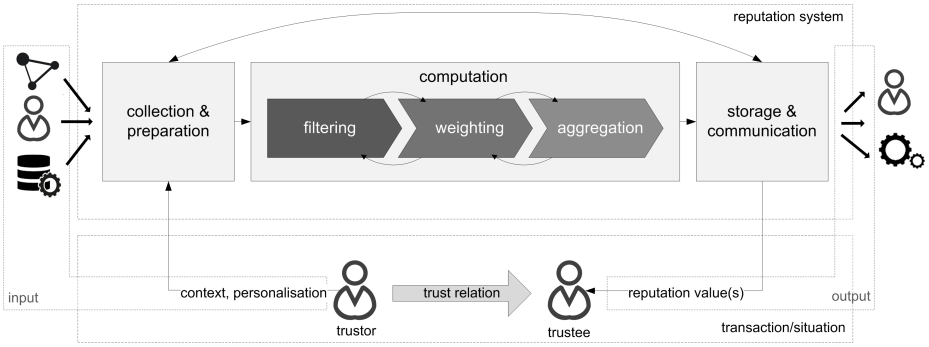

Fig. 1. Generic process of a reputation system, inspired by [10]

peers in a distributed network. After all available data is gathered, it is prepared for further use. Preparation techniques include, for instance, a normalization. Once the preparation is completed, the reputation data serves as input for the computation phase.

Computation The computation phase is the central part of every reputation system which takes the reputation information collected as input and generates a trust/reputation value as output. This phase can be divided into the three generic process-steps *filtering, weighting* and *aggregation*. Depending on the computation engine, not all steps have to be implemented. The first two steps, filtering and weighting preprocess the data for the subsequent aggregation. The need for these steps is obvious: The first question to be answered is *which* information is useful for further processing (filtering). The second process-step concerns the question of *how relevant* the information is for the specific situation (weighting). In line with this, Zhang et al. [12] pointed out that current trust models can be classified into the two broad categories *filtering-based* and *discounting-based*. The difference between filtering and weighting is that the filtering process reduces the information amount while it is enriched by weight factors in the second case. Filtering can, therefore, be seen as *hard selection* while weighting is more like a *soft selection*. Finally, the reputation values are aggregated to calculate one or several reputation scores. Depending on the algorithm, the whole computation process or single process steps can be run through for multiple times.

Storage and communication After reputation scores are calculated, they are either stored locally, in a public storage or both depending on the structure (central/decentralized/hybrid) of the reputation system. Common reputation systems not only provide the reputation scores but also offer extra information to help the end-users understand the meaning of a score-value. They should furthermore reveal the computation process to accomplish transparency.

In this work, we focus on the computation phase, since the first phase (collection & preparation) and the last phase (storage & communication) strongly depend on the structure of the reputation system (central or decentralized). The computation phase, however, is independent of the structure and can look alike for systems implemented in both central and decentralized environments. It, therefore, works well for design with reuse.

3.2 Hierarchical component taxonomy

In this section, the computation process is examined in detail. We will introduce a novel hierarchical component taxonomy that is based on the functional blocks of common reputation systems identified in this work. Thereto, we clarify the objectives of the identified classes (functions) and name common examples. Our analysis and selection of reputation systems is based on different surveys [8, 13, 6, 2, 1]. Figure 3.2 gives an overview of the primary and secondary classes identified.

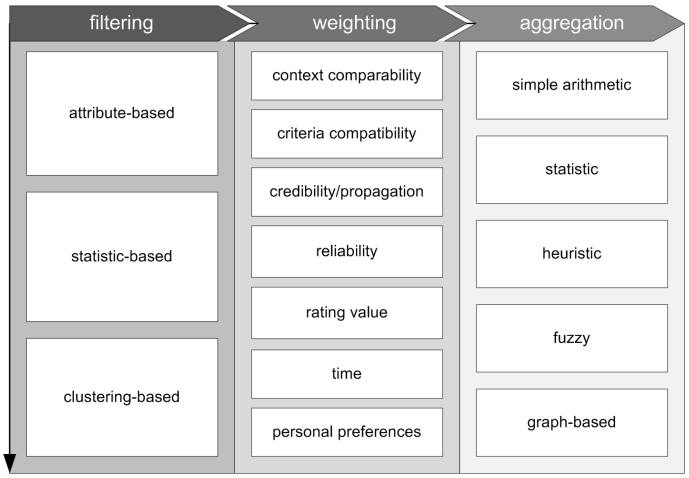

Fig. 2. Classes of filtering-, weighting- and aggregation-techniques

Beginning with the filtering phase, the three broad classes *attribute-based*, *statistic-based* and *clustering-based* filtering could be identified:

1. **Attribute-based filtering**: In several trust models, input data is filtered based on a constraint-factor defined for the value of single attributes. Attribute-based filters mostly implement a very simple logic, in which an attribute is usually compared to a reference value. Due to their lightweight, they are proper for reducing huge amounts of input data to the part necessary for the reputation calculation. Besides the initial filtering of input data, it is often applied after the weighting phase in order to filter referrals that have

been strongly discounted. An example of an attribute often constrained, is time, since it is desirable to disregard very old ratings. eBay's reputation system, for instance, only considers transactions having occurred in the last 12 months for their overview of positive, neutral and negative ratings. Other models such as Sporas [14] ignore every referral but the latest, if one party rated another party more than once. In this way, simple ballot stuffing attacks can be prevented. In ballot stuffing attacks, parties improve their reputation by means of positive ratings after fake transactions.

2. **Statistic-based filtering**: Further techniques that are used to enhance the robustness of trust models against the spread of false rumors apply statistical patterns. Whitby et al. [15], for example, proposed a statistical filter technique to filter out unfair ratings in Bayesian reputation systems applying the majority rule. The majority rule considers feedback that is far away from the majority's referrals as dishonest. In this way, dishonest or false feedback can easily be detected and filtered.
3. **Clustering-based filtering**: Clustering-based filter use cluster analysis approaches to identify unfair ratings. These approaches are comparatively expensive and therefore rarely used as filtering techniques. An exemplary procedure is to analyze an advisors' history. Since a rater never lies to himself, an obvious way to detect false ratings is to compare own experience with advisors' referrals. Thus, both fair and unfair ratings can be identified. iCLUB [16], for example, calculates clusters of advisors whose evaluations against other parties are alike. Then, the cluster being most similar to the own opinion is chosen as fair ratings. If there is no common experience (e.g. bootstrapping) the majority rule will be applied. Another example for an approach using cluster filtering was proposed by Dellorcas [17].

Once all available information is reduced to those suitable for measuring trust and reputation in the current situation, it becomes clear that various data differ in their characteristics (e.g. context, reliability). Hence, the referrals are weighted in the second process-step based on different factors. In contrast to the filtering, applied techniques strongly differ. For that reason, our classification of weighting techniques is based on the properties of referrals that are analyzed for the discounting. We identified the following classes:

1. **Context comparability**: Reputation data are always bound to the specific context in which it was created. Ratings that were generated in one application area might not be automatically applicable in another application area. In eCommerce, for instance, transactions are accomplished involving different prices, product types, payment methods, quality or time. The non-consideration of this context leads to the value imbalance problem where a malicious seller can build a high reputation by selling cheap products while cheating on expensive ones. To increase comparability and avoid such situations, context has become a crucial attribute for many current approaches like [18] or [9].
2. **Criteria comparability**: Besides the context in which feedback was created, the criteria that underlie the evaluation are important. Particularly,

if referrals from different application areas or communities are integrated, criteria comparability can be crucial. In file-sharing networks, for instance, a positive rating is often granted with a successful transaction independent of the quality of service. On eCommerce platforms, in contrast, quality may be a critical factor for customer satisfaction. Other distinctions could be the costs of reviews, the level of anonymity or the number of peers in different communities or application areas. Weighting based on criteria comparability can compensate these differences.

3. **Credibility/propagation**: In network structures such as in the web-of-trust, trust can be established along a recommendation or trust chain. Obviously, referrals that have first-hand information about the trustworthiness of an agent are more credible than referrals received at second-hand (with propagation degree of two) or higher. Several models, therefore, apply a propagation (transitivity) rate to discount referrals based on their distance. The biometric identity trust model [19], for instance, derives the reputation-factor from the distance of nodes in a web-of-trust.
4. **Reliability**: Reliability or honesty of referrals can strongly affect the weight of reviews. The concept of feedback reputation that measures the agents' reliability in terms of providing honest feedback is often applied. As a consequence, referrals created by agents having a low feedback reputation will have a low impact on the aggregated reputation. The bases for this calculation can be various. Google's PageRank [20], for instance, involves the position of every website connected to the trustee in the web graph in their recursive algorithm. Epinions[2], on the other hand, allows users to directly rate reviews and reviewers. In this way, the effects of unfair ratings are diminished.
5. **Rating value**: Trust is event sensitive. For stronger punishment of bad behavior, the weight of positive ratings compared to negative ratings can be calculated asymmetrically. An example for a model using an "adaptive forgetting scheme" was proposed by Sun et al. [21], in which good reputation can be built slowly through good behavior but easily be ruined through bad behavior.
6. **Time**: Due to the dynamic nature of trust, it has been widely recognized that time is one important factor for the weighting of referrals. Old feedback might not be relevant for reputation scoring as new referrals. An example measure for time-based weighting is the "forgetting factor" proposed by Jøsang [22].
7. **Personal preferences**: Reputation systems are used by various end-users (e.g. human decision makers, services). A reputation system must, therefore, allow the adaptation of its techniques to subjective personal preferences. Different actors might, for example, have different perceptions regarding the importance of direct experience and referrals, the significance of distinct information sources or the rating of newcomers.

The tuple of reputation data and weight-factor(s) serve as input for the third step of the computation process - the aggregation. In this phase, one or several

[2] http://www.epinions.com/

trust/reputation values are calculated by composing the available information. In some cases, the weighting and the aggregation process are run through repetitively in an iterative manner. However, the single steps can still be logically separated. The list of proposed algorithms to aggregate trust and reputation values has become very long during the last decade. Here, we summarize the most common aggregation techniques and classify them into the five blocks *simple arithmetic, statistic, heuristic, fuzzy and graph-based models*:

1. **Simple arithmetic**: The first class includes simple aggregation techniques like ranking, summation or average. Ranking is a very basic way to measure trustworthiness. In ranking algorithms, ratings are counted and organized in a descending order based on that value. This measure has no exact reputation score, however, it is frequently used as a proxy for the relative importance/trustworthiness. Examples for systems using ranking algorithms are message boards like Slashdot[3] or citation counts used to calculate the impact factor in academic literature. Other aggregation techniques that are well known due to the implementation on eBay or Amazon[4] are the summation (adding up positive and negative ratings) or the average of ratings. Summation, though, can easily be misleading, since a value of 90 does not reveal the composition of positive and negative ratings (e.g. +100,-10 or +90,0). The average, on the other hand, is a very intuitive and easy to understand algorithm.
2. **Statistic**: Many of the prominent trust models proposed in the last years use a statistical approach to provide a solid mathematical basis for trust management. Applied techniques range from *Bayesian probability* over *belief models* to *Hidden Markov Models*. All models based on the beta probability density function (beta PDF) are examples for models simply using Bayesian probability. The beta PDF represents the probability distributions of binary events. The *a priori* reputation score is thereby gradually updated by new ratings. Result is a reputation score that is described in a beta PDF function parameter tuple (α, β), where α represents positive and β represents negative ratings. A well known model using the beta PDF is the the Beta Reputation system [22]. A weakness of Bayesian probabilistic models, however, is that they cannot handle uncertainty. Therefore, belief models extend the probabilistic approach by DempsterShafer theory (DST) or subjective logic to include the notion of uncertainty. Trust and reputation models involving a belief model have been proposed by Jøsang [23] or Yu and Singh [24]. More complex solutions that are based on machine learning, use the Hidden Markov Model, a generalization of the beta model, to better cope with the dynamic behavior. An example was introduced by Malik et al. [25].
3. **Heuristic**: Since statistical approaches are very complex, a shift towards heuristic-based trust modeling has become visible in scientific literature. Heuristic approaches try to provide custom-designed practical and easy to

[3] http://www.slashdot.org/
[4] http://www.amazon.com/

understand and implement solutions. Thereby, the filtering and weighting phases are of high importance as the aggregation is mostly based on a combination of rating and weights. Exemplary models were proposed by Xiong and Liu [26] or Zhang and Wang [18].

4. **Fuzzy**: Aggregation techniques classified as fuzzy models use fuzzy logic to calculate a reputation value. In contrast to classical logic, fuzzy logic allows to model truth or falsity within an interval of [0,1]. Thus, it can describe the degree to what an agent/resource is trustworthy or not trustworthy. Fuzzy logic has been proven to deal well with uncertainty and mimic the human decision-making process [27]. Thereby, a linguistic approach is often applied. REGRET [28] is one prominent example of a trust model making use of fuzzy logic.
5. **Graph-based**: A variety of trust models employ a graph-based approach. They rely on different measures describing the position of nodes in a network involving the flow of transitive trust along trust chains in network structures. As online social networks have become popular as a medium for disseminating information and connecting people, many models regarding trust in social networks have lately been proposed. Graph-based approaches use measures from the field of graph theory such as centrality (e.g. Eigenvector, betweenness), distance or node-degree. Reputation values, for instance, grow with the number of incoming edges (in-degree) and in- or decrease with the number of outgoing edges (out-degree). The impact of one edge on the overall reputation can depend on several factors like the reputation of the node an edge comes from or the distance of two nodes. Popular algorithms using graph-based flow model are Google's PageRank [20] as well as the Eigentrust Algorithm [29]. Other examples are the web-of-trust or trust models particularly designed for social networks as described in [6]. As mentioned above, due to the incremental nature of these algorithms, the weighting and aggregation phases are incrementally run through for several times.

The classification of the computation engine's components used in different trust models in this taxonomy is not limited to one component of each primary class. Depending on the computation process, several filtering, weighting and aggregation techniques can be combined and run through more than once. Malik et al. [25], for instance, introduced a hybrid model combining heuristic and statistical approaches. However, our taxonomy can reveal the single logical components, a computation engine is built on. It, moreover, serves as an overview of existing approaches. Since every currently known reputation system can find its position, to the best of our knowledge, this taxonomy can be seen as complete. Though, an extension by new classes driven by novel models and ideas is possible. Our hierarchical component taxonomy currently contains 3 primary component classes, 15 secondary component classes, 26 component terms and 36 subsets. Table 1 shows an excerpt of the hierarchical component taxonomy with building blocks of the primary class "weighting".

Table 1. Excerpt of the hierarchical component taxonomy

Primary component class	**Secondary component class**	**Component term**	**Subset**	**Description**
weighting	credibility/ propagation	propagation discount		Discount referrals along trust chains
	reliability	subjective reliability	property similarity	Discount based on similarity of personal properties
			rating similarity	Discount referrals based on similarity of ratings toward other agents
		objective reliability	Explicit	Discount based on explicit reputation information like referrals or certificates
			Implicit	Discount based on implicit reputation information like profile age, number of referrals or position
...	...	...	...	...

4 The component taxonomy as a framework for design with reuse

The hierarchical component taxonomy introduced in the former section, serves as a natural framework for the design of reputation systems with reuse. To support this process, we set up a component repository combining a knowledge and a service repository. Thus, it does not only contain information about software components on implementation level but also provides extensive descriptions of the ideas applied on a conceptual level. This comprehensive set of fundamental component concepts and ideas combined with the related implementation allows the reuse of both ideas and already implemented components.

In this section, we first describe the conceptual design of our component repository in detail. Then, we show how we implemented a web application using our thorough repository to provide design knowledge for reuse on a conceptual and an implementation level.

4.1 Conceptual design of the component repository

Reuse-based software engineering can be implemented on different levels of abstraction, ranging from the reuse of ideas to the reuse of already implemented software components for a very specific application area. In this work, we want to apply our taxonomy for reuse on two levels - a conceptual level and an implementation level. The developed repository, therefore, provides design knowledge for reuse on two logical layers, as depicted in Figure 3.

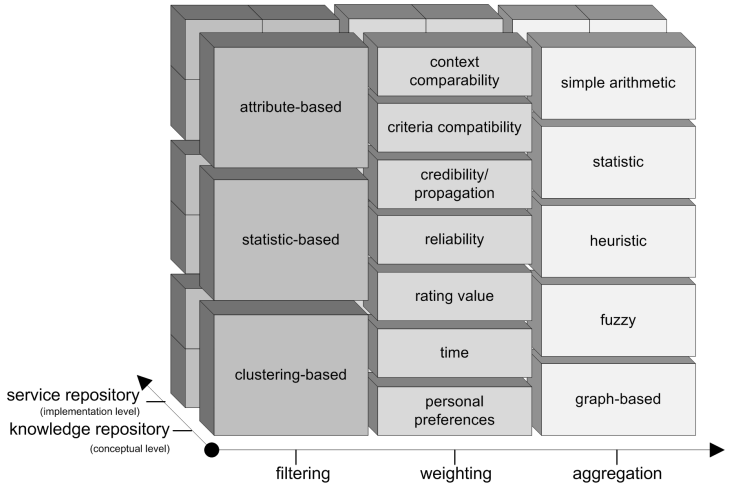

Fig. 3. Logical layers of the component repository for design with reuse

Reuse on conceptual level When reusing an implemented component one is unavoidably constrained by design decisions that have been made by the developer. A way to prevent this, is to conceive more abstract designs that do not specify the implementation. Thus, we provide an abstract solution to a problem by means of design pattern-like concepts. Design patterns are descriptions of commonly occurring problems and a generic solution to the problems that can be used in different settings [30]. Our design pattern-like concepts consist of essential elements that are exemplary depicted on Table 2.

Reuse on implementation level On implementation level we provide fully implemented reusable components by means of web-services in a service-orientated architecture. These services encapsulate the concepts' logic and functionality in independent and interchangeable modules to achieve the separation of concerns. The web-services are incorporated via well-defined interfaces. All service provided are registered as artifacts in the service repository. An artifact contains essential information about one live reachable service such as ID, type (REST or ws), URL, description, parameters, example calls, example output and the design pattern that is implemented by the service.

4.2 Implementation of the repository

We prototypically implemented our repository as a web-based application in a three-tier client-server-architecture[5]. On client-side (presentation layer) we employed the current web standards HTML5, JavaScript and CSS (Bootstrap). On server-side the logic was implemented in PHP on an Apache server (logic layer)

[5] http://trust.bayforsec.de

Table 2. Design pattern on the conceptual level (example)

Component term	Context similarity
Subset	Absolute congruence
Description	This component uses an absolute congruence metric as similarity measure to identify context similarity.
Problem description	Reputation data is always bound to the specific context in which it was created. Ratings that were generated in one application area might not be automatically applicable in another application area which can result in the value imbalance problem.
Solution description	Apply similarity measurement between context c_i (reference context) and context c_j of referrals in the referral set to deliver a weight-factor for each item of the referral set using the following formula: $$w(c_1, c_2) := \frac{k(c_i) \cap k(c_j)}{k(c_i) \cup k(c_j)}$$ $k(c_i)$ denotes the total number of keywords describing context c_i.
Applicability	Set of nominal context attributes.

Code example (php)	`function calculate_values($reference, $context_sets) {`
	`  $reference_context = $reference['context_attributes'];`
	`  $return_values = array();`
	`  while(!empty($context_sets)) {`
	`  ...shortened...`
	`  }`
	`  return $return_values;`
	`}`

Implementation	Context similarity-based weighting service (absolute congruence)
Literature	– Mohammad Gias Uddin, Mohammad Zulkernine, and Sheikh Iqbal Ahamed. 2008. CAT: a context-aware trust model for open and dynamic systems. In Proceedings of the 2008 ACM symposium on Applied computing (SAC '08). ACM, New York, NY, USA, 2024-2029.
Tags	weighting, context, similarity, congruence

connecting to a MySQL-Database (persistent layer). All webservices were created in PHP an registered as artifacts in our service repository. This web-application is planned to become a platform for researchers to make their concepts and implementations publicly available.

5 Contribution and Future work

Many surveys of trust and reputation systems give an overview of existing trust and reputation systems by means of a classification of existing models and approaches. In contrast to that, we provide a collection of ideas and concepts classified by their functions. Furthermore, these ideas are not only named but also clearly described in well-structured design pattern-like artifacts which can easily be adapted to a specific situation. Therewith, we reorganized the design knowledge for computation techniques in reputation systems and translated the most common ideas to a uniform format. To directly make use of novel components, the webservices created on implementation level can instantly be reused

and integrated in existing reputation systems to extend their capabilities. This approach, to publicly provide implemented computation components as webservices may help to better spread innovative ideas in trust and reputation systems and to give system builders a better choice allowing to experiment with different computation techniques. We, moreover, encourage researchers to focus on the design of single components by providing a platform, where concepts and their prototypical implementation can be made publicly available.

However, there are still many unexplored areas regarding the design with reuse in trust and reputation systems. The following list gives an overview of those issues that will be topic for our future research:

1. Reusability in collection & preparation and storage & communication: The work in hand considers the design of computation techniques with reuse. Reusability could also play a role in other process steps run through in a reputation system. To clarify the opportunities, further research is necessary in this area.
2. Additional views on the component repository: Currently, our hierarchical taxonomy provides a functional view on the identified components. However, a developer could also benefit from additional views, like an attack view, in which the components are classified as possible solutions in a taxonomy of attacks on reputation systems.
3. Generic testbed and evaluation criteria: To measure the quality of a component, a testbed for comparison and a set of sound evaluation criteria such as robustness, efficiency or complexity is needed.
4. Software-supported selection of components: The selection and interpretation of adequate components for new reputation systems in a specific application area requires time, effort and to some extend knowledge of this research area. To increase usability and simplicity, a software application is needed to support a user in this development process.
5. Advanced meta information and machine readability: To take one step further, the most qualified composition could be automatically found and assembled by a software program based on the reputation data (input) provided. The research involves the development of sound principles for automated component composition.

6 Conclusion

The research in trust and reputation systems is still growing. In this paper, we presented concepts to foster reuse of existing approaches. We provided a hierarchical taxonomy of computation components from a functional view and described the implementation of a component repository that serves as both a knowledge base and a service repository. In this way, we communicate design knowledge for reuse, support the development of new reputation systems and encourage researchers to focus on the development of single components that can be integrated in various reputation systems to easily extend their capabilities by new features.

Acknowledgments The research leading to these results was supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association.

References

1. Yao, Y., Ruohomaa, S., Xu, F.: Addressing Common Vulnerabilities of Reputation Systems for Electronic Commerce. Journal of theoretical and applied electronic commerce research **7**(1) (2012) 1–20
2. Tavakolifard, M., Almeroth, K.C.: A Taxonomy to Express Open Challenges in Trust and Reputation Systems. Journal of Communications **7**(7) (2012) 538–551
3. Alan R. Hevner, Salvatore T. March, Jinsoo Park, Sudha Ram: Design Science in Information Systems Research. MIS Quarterly **28**(1) (2004) 75–105
4. McKnight, D.H., Chervany, N.L.: The meanings of trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Reseach Center (1996)
5. Gambetta, D.: Can We Trust Trust? In Gambetta, D., ed.: Trust: Making and Breaking Cooperative Relations. Basil Blackwell, Oxford (1988) 213–237
6. Sherchan, W., Nepal, S., Paris, C.: A survey of trust in social networks. ACM Computing Surveys **45**(4) (2013) 1–33
7. Artz, D., Gil, Y.: A survey of trust in computer science and the Semantic Web. Web Semantics **5**(2) (2007) 58–71
8. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems **43**(2) (2007) 618–644
9. Rehak, M., Gregor, M., Pechoucek, M., Bradshaw, J.: Representing Context for Multiagent Trust Modeling. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IAT '06, Hongkong, China (2006) 737–746
10. Swamynathan, G., Almeroth, K.C., Zhao, B.Y.: The design of a reliable reputation system. Electronic Commerce Research **10**(3-4) (2010) 239–270
11. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Communications of the ACM **43**(12) (2000) 45–48
12. Zhang, L., Jiang, S., Zhang, J., Ng, W.K.: Robustness of Trust Models and Combinations for Handling Unfair Ratings. In Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H., eds.: Trust Management VI. Volume 374 of IFIP advances in information and communication technology. Springer, Berlin, Heidelberg (2012) 36–51
13. Noorian, Z., Ulieru, M.: The State of the Art in Trust and Reputation Systems: A Framework for Comparison. Journal of theoretical and applied electronic commerce research **5**(2) (2010) 97–117
14. Zacharia, G., Moukas, A., Maes, P.: Collaborative reputation mechanisms for electronic marketplaces. Decision Support Systems **29**(4) (2000) 371–388
15. Andrew Whitby, Audun Jøsang, Jadwiga Indulska: Filtering out unfair ratings in bayesian reputation systems. In R. Falcone, S. Barber, J. Sabater and M. Singh, ed.: Proceedings of the 7th International Workshop on Trust in Agent Societies, Rome, Italy (2004) 106–117
16. Liu, S., Zhang, J., Miao, C., Theng, Y.L., Kot, A.C.: iCLUB: An Integrated Clustering-based Approach to Improve the Robustness of Reputation Systems. In: The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3. AAMAS '11, Taipei, Taiwan (2011) 1151–1152

17. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: Proceedings of the 2nd ACM conference on Electronic commerce. ACM, New York (2000) 150–157
18. Zhang, H., Wang, Y., Zhang, X.: A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments. In: Proceedings of the 5th IEEE International Conference on Service-Oriented Computing and Applications. Volume SOCA., Taipei, Taiwan (2012) 1–8
19. Obergrusberger, F., Baloglu, B., Sänger, J., Senk, C.: Biometric Identity Trust: Toward Secure Biometric Enrollment in Web Environments. In Akan, O., Bellavista, P., Cao, J., Dressler, F., Ferrari, D., Gerla, M., Kobayashi, H., Palazzo, S., Sahni, S., Shen, X., Stan, M., Xiaohua, J., Zomaya, A., Coulson, G., Yousif, M., Schubert, L., eds.: Cloud Computing. Volume 112 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, Cham (2013) 124–133
20. Sergey Brin and Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the 7th International Conference on World Wide Web. Volume 1-7., Brisbane and Australia 107–117
21. Sun, Y., Han, Z., Yu, W., Ray Liu, K.: Attacks on Trust Evaluation in Distributed Networks. In: Proceedings of th 40th Annual Conference on Information Sciences and Systems. CISS, Princeton, USA (2006) 1461–1466
22. Audun Jøsang, Roslan Ismail: The Beta Reputation System. In: In Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia (2002)
23. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **09**(03) (2001) 279–311
24. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: Proceedings of the 1st international joint conference on Autonomous agents and multiagent systems. AAMAS '02, Bologna, Italy (2002) 294–301
25. Malik, Z., Akbar, I., Bouguettaya, A.: Web Services Reputation Assessment Using a Hidden Markov Model. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Baresi, L., Chi, C.H., Suzuki, J., eds.: Service-Oriented Computing. Volume 5900 of Lecture notes in computer science. Springer Berlin Heidelberg, Berlin, Heidelberg (2009) 576–591
26. Li Xiong, Ling Liu: A reputation-based trust model for peer-to-peer e-commerce communities. In: Proceedings of the IEEE International Conference on E-Commerce. CEC '03, Newport Beach, USA (2003) 275–284
27. Shanshan Song, Kai Hwang, Runfang Zhou, Yu-Kwong Kwok: Trusted P2P Transactions with Fuzzy Reputation Aggregation. In: IEEE Internet Computing. Volume 9. IEEE Educational Activities Department, Piscataway (2005) 24–34
28. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems. AAMAS '02, Bologna, Italy (2002) 475–482
29. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th international conference on World Wide Web. WWW '03, Budapest, Hungary (2003) 640–651
30. Gamma, E.: Design patterns: Elements of reusable object-oriented software. Addison-Wesley professional computing series. Addison-Wesley, Reading (1995)