VIeTE
Enabling Trust Emergence in Service-oriented Collaborative Environments

Florian Skopik, Hong-Linh Truong and Schahram Dustdar
Distributed Systems Group
Information Systems Institute
Vienna University of Technology, Austria

Keywords: Activity-centric collaboration, Trust, Service-oriented architecture.

Abstract: In activity-centric environments where people from different companies and disciplines work remotely together and where new virtual teams are formed and dissolved continuously, how to find the most suitable collaboration partner for a given task and how well one partner is able to collaborate with another one are challenging research questions. Determining and considering people's professional competencies, collaboration behavior and relationships is a prerequisite to enhance the overall collaboration performance and success, because these factors highly impact on the notion of trust used to select and grade partners. In this paper we analyze these factors and their impact on trust relationships in modern service-oriented collaboration environments. We present VieTE, a framework for trust emergence therein supporting the analysis of trust between partners in various contexts and from different views. In contrast to other approaches, which mostly rely on manual and subjective user feedback, VieTE monitors automatically collaboration efforts and deduces trust between any two partners based on past collaboration, previous successes, and individual competencies.

1 INTRODUCTION

With the rise of information and communication technologies, the way people organize and perform their work has been shifted to a distributed form, where people located at different sites build loosely coupled teams and work together to reach common goals. In such working scenarios, it is difficult for team members to establish personal relationships. However, especially in team oriented environments, one aspect of interpersonal relationships must not be neglected, which is *trust*. High trust between collaboration partners is vital to collaboration processes and thus to their overall success.

A wide range of systems have been proposed for establishing trust, such as those described in (Jøsang et al., 2007), but most of them rely on subjective user feedback, which is time-consuming for the users and error-prone due to social influences or malicious raters. In order to overcome this user feedback dependency and to automate the rating process, we follow a monitoring-based approach by observing and analyzing users' communication and behavior to determine notions of trust during collaborations in ad-hoc, service-oriented collaboration environments. To this end, we have developed the *Vienna Trust Emergence Framework* (VieTE) to support the analysis of trust between any collaboration partners. By determining and providing trust values directly from monitoring collaborations, VieTE improves the support for typical use cases in ad-hoc collaboration scenarios including selecting a partner or service at run-time, permitting user recommendation and ranking, allowing trust-based team formation, supporting trustworthy resource access control or enabling evaluation of team performance; just to mention a few examples.

The remainder of this paper is organized as follows. Section 2 describes our supporting service-oriented collaboration environment. Section 3 presents the related work. In Section 4 we define trust together with its context and views. We present data collection and complex interaction metrics for determining trust in Section 5. Section 6 describes VieTE's architecture and functionalities. We discuss illustrative scenarios in Section 7 and conclude the paper in Section 8.

Skopik F., Truong H. and Dustdar S.
VIeTE - Enabling Trust Emergence in Service-oriented Collaborative Environments.
DOI: 10.5220/0001820904710478
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

2 SERVICE-ORIENTED COLLABORATION ENVIRONMENT

We consider all motivating use cases mentioned in the introduction, and describe a service-oriented collaboration environment which is generic enough to be used in a wide range of real scenarios. In this environment *humans* are organized in *teams* performing *activities* with the support of SOA-based *services*. In this regard the term collaboration means that people work together in various ways to reach a common goal. All tasks they perform are organized in activities, which are structures to help managing and monitoring which humans are jointly performing which tasks by utilizing which services. Figure 1 shows an overview of involved entities and their relationships.

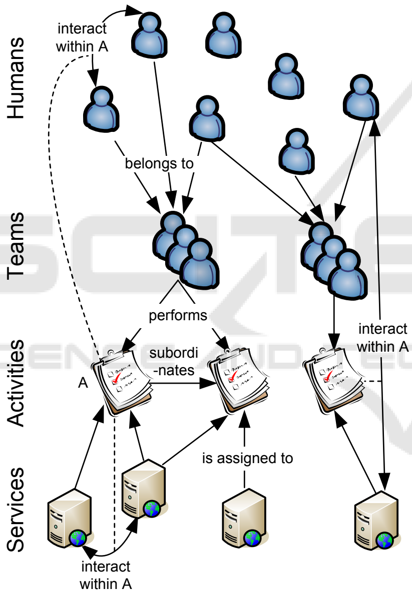

Figure 1: Relationships between entities.

A ***human*** is a single person who *belongs to* one or more ***teams*** at the same time. Every team member has one or more roles, which describe their responsibilities within a team (e.g. leader, contributor etc.), *interacts with* services in context of particular activities and *interacts with* other humans of the same team. Every interaction has a particular purpose, thus we can distinguish between various types including coordination (e.g. agree on a meeting schedule), communication (e.g. send instant message or e-mail) or execution (e.g. execute a service to fulfill a task), dependent on which category of service is utilized. In the described environment we monitor different kinds of interactions, such as SOAP-based (Web) service calls, and e-mail and instant messages.

An ***activity*** is any kind of basic task (e.g. testing a software module) performed during work and is executed by exactly one team, however a team can be assigned to several activities at the same time. Activities are hierarchically structured and can have an arbitrary number of *subactivities*. Furthermore, activities have a particular goal and nature, e.g., in a software testing activity a final goal may be the upload of a test report to a repository. All interactions between humans and/or services take place in context of particular activities.

A ***service*** is a resource *assigned to* an activity providing support for a team during an activity's execution. We are operating in a mixed systems environment, where services may be commonly known software Web services or humans acting as services (e.g., through human provided services (Schall et al., 2008)). Such an environment permits supporting more complex tasks, which cannot be tackled by traditional Web services, but by humans using common Web service technologies and widely adopted infrastructures. If several services are assigned to the same activity they can interact with each other, building service compositions to offer extended functionalities.

3 RELATED WORK

In this paper we present a framework, VieTE, to enable trust emergence in human collaboration environments, which are similar to previous activity-centric approaches, such as IBM's UAM[1] presented for example in (Moody et al., 2006). Several projects in the field of collaboration are currently performed within the EU FP7[2], partly utilizing the concept of trust.

From the comprehensive surveys of trust in computer science, including (Jøsang et al., 2007), (Ruohomaa and Kutvonen, 2005) and (Artz and Gil, 2007), we select and extend the trust definitions in (Mui, 2002) and (Grandison and Sloman, 2000) which fit best to our framework as explained in Section 4.

The overall use of VieTE is related to classic recommender systems (Resnick and Varian, 1997) and collaborative filtering approaches (Herlocker et al., 2004), however, in these fields the opinion of an entire community about a single entity is predominantly recognized while relations between single entities within a group are often neglected. Our work introduces the concept of *views of trust*, which allows to grade an entity from different perspectives, including from an

[1]http://www.research.ibm.com/uam/
[2]http://cordis.europa.eu/fp7

entire communiy's view but also from an individual's view. There are many reputation models from the SOA domain, such as (Maximilien and Singh, 2004) , but they are dedicated to Web services only.

In contrast to mentioned reputation systems, in the domain of social network analysis (Wasserman and Faust, 1994) the relationships between single entities are highly researched. From this area we get valuable input about the composition of typical user communities, such as (Gomez et al., 2008). Experimental case studies, including (Massa and Avesani, 2005), offer insights in human collaboration behavior and enables us to define requirements for our framework and a basic trust model.

The aim of trust models is to abstract the fuzzy notion of trust and to build a mathematical model to enable systematic trust calculation and analysis between any entities. There are several papers dealing with the definition of trust metrics and models in general (Ramchurn et al., 2004), (Huynh et al., 2006), (Theodorakopoulos and Baras, 2006) or focusing particular aspects such as propagation (Guha et al., 2004), (Quercia et al., 2007) or mobility (Shand et al., 2004). For basic trust determination, we utilize the widely adopted concept of a trust graph where its nodes represent acting entities and weighted edges reflect the relationships among entities.

4 CONTEXT AND VIEWS OF TRUST

Before discussing the context and views associated with trust, we define the notion of trust in our framework. Trust has been defined in several different ways depending on the research area, such as in (Grandison and Sloman, 2000) and (Artz and Gil, 2007). A definition of trust from (Mui, 2002) suitable for the introduced collaboration environment states that trust is *"a subjective expectation an agent has about another's future behavior based on history of their encounters."*. The point is that inferred from previous collaboration behavior and experiences a notion of trust is deduced.

Particularly in collaboration scenarios where people from different companies and from a wide range of disciplines work together using services from several vendors, they are often unknown to each other, thus trust cannot rely on personal relationships, but has to be mostly determined by the success of past collaborations and the quality of the outcome only. Hence we argue that one's trust in another one is higher, the more efficiently both performed in the same activities and teams respectively. In the described collaboration environment success and efficiency basically depend on the competencies of the acting entities, in detail humans and services. A human offers competencies such as special skills and capabilities and a service functionalities and features to support particular activities. From this perspective, we adopt the definition from (Grandison and Sloman, 2000), trust is *"the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context"*. Based on that, we combine and extend above definitions of trust to define trust in our collaboration environment as:

> *a subjective opinion based on previous collaboration experiences one entity has about another's competencies to act dependably, securely, and reliably within a specified context, determined by performed activities and involved teams.*

4.1 Context of Trust

It is widely agreed (McKnight and Chervany, 1996), (Marsh, 1994), (Grandison and Sloman, 2000) and an integral part of the above definition that trust is context dependent, which means it is determined for particular entities in particular situations. In contrast to a wide range of reputation systems (Herlocker et al., 2004), which calculate only one kind of globally valid trust value, independent from situation and use case, we derive trust for certain humans and services with respect to their context. Particularly in the introduced collaboration environment context, which reflects a real situation, can be fully described by the notions of team and activity. The team holds structural information about which humans work closely together to reach a common goal and which roles they have, while the activity describes the goal itself to reach. Thus, contextual information of an entity includes all properties available about current and past activities and teams involved.

4.2 Views of Trust

As mentioned in the related work section, trust has been widely defined (i) from an individual view in social networks, where relations between single users are maintained or (ii) from a global view in reputation systems or collaborative tagging systems, which mostly use an aggregation of the individuals' views. For typical collaboration scenarios, where humans are tightly coupled and form teams, we introduce one level in between by taking a team's view of trust into account. This enables us to determine trust of an entire team into another entity, which is a basic demand in collaborative decisions, such as the selection of further team members or services.

In the described collaboration environment we distinguish between the following views of trust, which can be created by combining contextual information from different individuals:

- *individual view:* describes trust of one human in another one or in a service.
- *team view:* describes trust of an entire team in one human or service. For determining team trust previous collaboration encounters from all team members with a particular entity are aggregated.
- *global view:* determines trust in a human or service from a global point of view, similar to global reputation systems, where all available information within a collaboration scenario is taken into account.

Figure 2 shows examples of the three views of trust in a service and lists the influencing factors for trust determination.

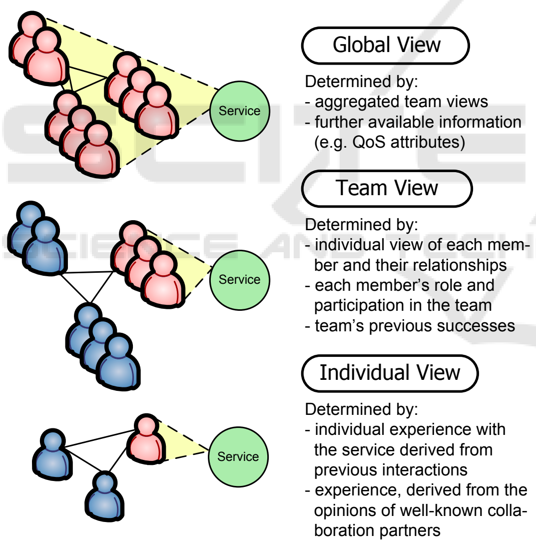

Figure 2: Distinguished views of trust in a service.

The differentiation into diverse *views of trust* combined with the notion of *trust context* is the basis for a comprehensive customization of our framework (as further described in Section 6).

5 COLLECTING DATA FOR DETERMINING TRUST

To enable VieTE to provide notions of trust from different views *and* in different contexts, we need an appropriate data model supporting the described collaboration environment, and utilize a number of data sources to determine trust.

5.1 Data Model

The four main entities of the data model *human*, *team*, *activity* and *service*, and their connections as mentioned in Section 2, are modeled in Figure 3. Every entity is described with further attributes, their so called profile data, serving as input for trust determination. These profiles describe a collaboration scenario including its participating entities, and are available to all entities within the same collaboration scenario. For example all humans within a team share their profile data with each other.

Figure 3 depicts our data model. We distinguish two different kinds of data, which are (i) *profile data*, describing an entity's properties and structural relationships and (ii) *interaction data*, describing dynamic collaboration encounters.

The data model is designed to be generic enough to handle different forms of collaboration scenarios with a wide variety of characteristics. If there is, for instance, no need for a notion of team, all humans can be assigned to a single team or each human may represent an own team respectively. The entity's properties are selected to be applicable in various contexts.

5.2 Data Sources

Besides information about human profiles and team structures, we store data about current and past activities as well as service profiles. Furthermore, in our approach we do not rely on subjective user feedback or reputation, for instance, in form of questionnaires, but we take into account what can be measured automatically. We use a Web services infrastructure which enables us to log low level interaction messages by using Web service handlers and SOAP interceptors. Hence, we are able to capture (i) human-human interactions as long as they take place via observable communication services (ii) human-service interactions, (iii) service-service interactions in service compositions, (iv) predefined service or application events in customized logging scenarios (e.g. e-mail traffic over SMTP or SVN file accesses), and (v) changes of team- and activity structures including adding team members, changing roles or assigning new services to an activity.

For this measurement approach it is necessary to determine the success of interactions. Depending on the type of interactions and participating entities, interaction failures can have manifold reasons,

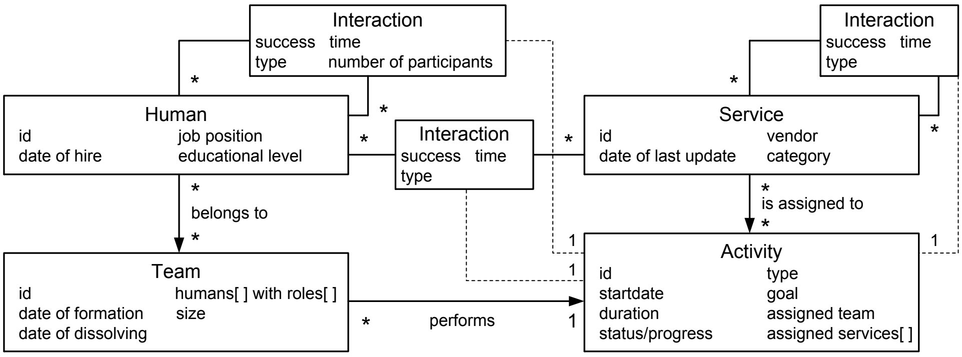

Figure 3: Simplified data model of profiles and interactions.

such as SOAP exceptions if a particular service is down, application specific errors due to missing features or wrong usage of services, not running instant messengers or e-mail clients, unanswered instant messages, missed read notifications of e-mails, interrupted data transfers, invalid communication settings, missed phone calls etc.

5.3 Collaboration Metrics

Data from all mentioned sources is captured and aggregated to more detailed composite metrics and patterns providing meaningful information for a collaboration scenario. The calculation process is fully customizable and can be set up for different views of trust.

Table 1: Exemplary metrics for humans.

View	Metric Description
Indiv.	scores for initiated, accepted, successful and failed human-to-human/service interactions. interaction success with a particular human/service. experience with a particular activity type.
Team	human's average impact on team performance. human's interaction participation within a team.
Global	metrics for team view can be applied too, but aggregated over all teams.

Table 2: Exemplary metrics for teams.

View	Metric Description
Indiv.	only profile data is available.
Team	average number of members' interactions with a particular entity. interaction distribution among members. service access distribution (with respect to a particular or all activities).
Global	amount of interactions compared to other teams. activity participation compared to other teams. team success compared to other teams.

We elaborate some exemplary metrics in Tables 1 to 4 to show the potential of our approach. The exact definitions of these metrics depend on available collaboration data and thus on the environment, and is not in scope of this paper, which focuses a high-level overview of the whole approach. In complex collaboration scenarios more metrics, especially aggregated from simpler ones, are possible. Currently we focus on basic metrics which mostly depend on interaction data, such as interaction scores and success rates.

Table 3: Exemplary metrics for activities.

View	Metric Description
Indiv.	none. Activities are performed by entire teams.
Team	collaboration effort compared to other activities. success of service usage within an activity. success of executive team within an activity. success of artifact outcome. service access distribution within an activity. activity contribution of every team member.
Global	metrics for team view can be applied too, but compared to activities of all teams.

Table 4: Exemplary metrics for services.

View	Metric Description
Indiv.	scores for initiated, accepted, successful and failed service-to-service/human interactions. interaction success with a particular human/service.
Team	average number of invocations. usage factor by a particular team. service access distribution by a particular team.
Global	same metrics as for team view aggregated over all teams.

6 VIeTE ARCHITECTURE

To support trust in the described collaboration environment VieTE consists of (i) tools for managing humans, teams, activities and services, utilizing services in the back-end, (ii) sensors and logging mechanisms to monitor interactions at run-time, and (iii) a trust determination service to deal with activity structures, service information, human and team profiles, and interaction logs, to calculate metrics, and ultimately to determine trust between any interacting entities. Figure 4 depicts the architecture of VieTE.

6.1 Determining Trust

In collaboration scenarios, humans use several tools to manage their activities, to search for suitable partners or services and to formate teams. VieTE provides trust-aware support for these tasks. To this end, we monitor collaborations of humans with other humans and services to collect data for trust determination. We describe the mainly involved components in the following:

Interaction Sensors and Logging. Interactions between and among humans and services are captured by sensors. We have developed mechanisms to intercept SOAP calls to services, e-mail traffic, instant messages over XMPP[3] and SVN document repository accesses, however this can be extended by sensors for a various range of open or proprietary communication protocols, including voice calls or file sharing. The logged entries contain at least the type of interaction, sender-id, receiver-id and a timestamp; furthermore, dependent on interacting entities, the endpoint interface, invoked operation, parameters etc. All communication and application errors are logged as well, e.g., SOAP exceptions or access denials.

Collaboration Metrics Calculation. Based on collected interaction data and information from collaboration management services this component calculates context dependent metrics for humans, teams, services and activities, as described before.

Trust Model. This component implements a directed graph, where the nodes represent humans and services and the links reflect their trust relationships. These relationships depend on the metrics calculated in discrete time steps according to the context described by activities and teams. The realization of these functionalities depends on the utilized trust model, but currently we build trust using weighted averages of preselected metrics. The model is customized by user specified policies, which control the metrics to be used and how these metrics are combined, i.e. the weighting factors, to deduce a notion of trust.

Trust Provider and API. The provider extracts data from the trust model and creates context dependent views according to API parameters. Collaboration Management Tools can access the trust model data through the Trust API. The following excerpt of this interface shows the signature of the methods provided to obtain the trust relationship from an entity `trustor` to an entity `trustee` restricted to a context `ctx`. The view of trust is derived from the type of the entity trustor, which may be either individual, team or global.

```
int getTrust(Entity trustor, Entity trustee, Ctx ctx);
List<Entity> getTrustors(Entity trustee, Ctx ctx);
List<Entity> getTrustees(Entity trustor, Ctx ctx);
...
```

[3] http://www.xmpp.org/

6.2 Prototype Implementation

The **VieTE portal** consists of a Liferay Enterprise Open Source Portal[4] in which tools run as JSR-168[5] compatible portlets. We use portlets for visualizing and processing user inputs only, while the main part of the business logic is encapsulated in Web services.

In the **back-end**, we use a Tomcat server with a deployed Axis2 for hosting Web services. We utilize existing, earlier developed services for registering and managing humans, teams, activities and services, and for interaction logging; and develop new services for all tasks concerning trust determination and management. Communication between portlets and services is realized with a SOAP based Web service stack. All relevant data is stored in a IBM DB2.

For the implementation of the trust model and applying basic graph algorithms we utilize the Java Universal Network/Graph Framework[6].

7 ILLUSTRATIVE SCENARIOS

We set up a collaboration scenario in the field of software development which consists of two teams as shown in Figure 5, where trust values $\in [0,1]$, ranging from no to full trust, are calculated between any two entities, humans or services which directly interact. These values are derived from the types and amount of successful interactions compared to the total amount of interactions between two particular entities.

For the sake of simplicity, we assume each team is predominantly involved in only one type of activity, which is software implementation for $team_A = \{H_1, H_2, H_3, H_8\}$ and software testing for $team_B = \{H_4, H_5, H_6, H_7, H_8\}$. Each human has a particular role (developer, assistant, trainee) in a team. H_8 is member of both teams, thus has two roles. For the

[4] http://www.liferay.com
[5] http://jcp.org/aboutJava/communityprocess/final/jsr168/
[6] http://jung.sourceforge.net/

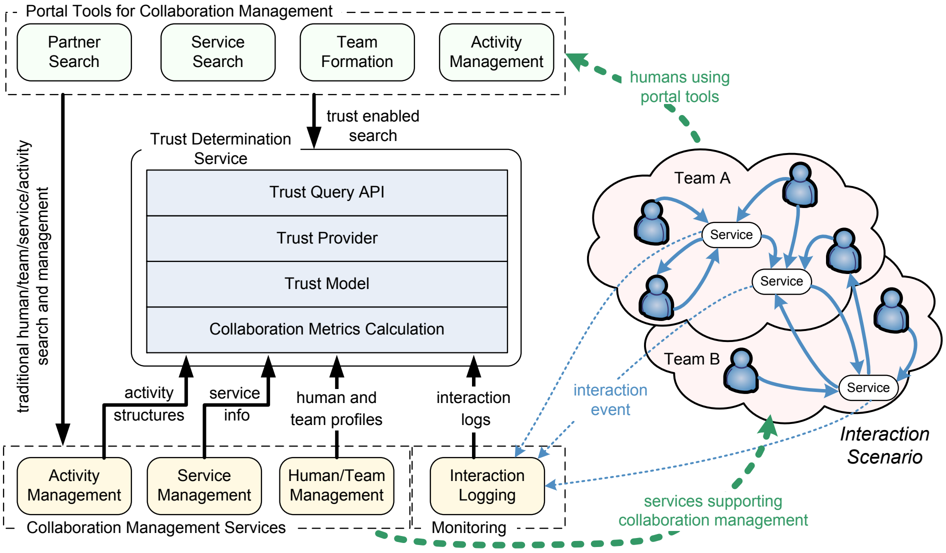

Figure 4: Architectural overview of the framework.

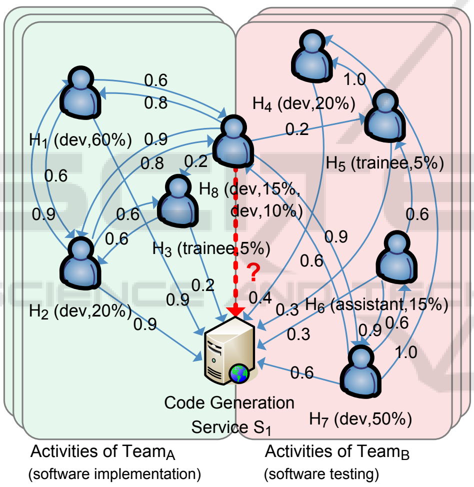

Figure 5: Experimental setup with humans having different roles and interaction participation within teams (in %).

creation of global and team views we built the average of the involved individuals' trust values weighted by their interaction participation. Interaction participation refers to the amount of successful interactions performed by one human expressed in percent of all successful interactions within a team.

Scenario 1: Trust Determination. H_8 wants to know if it is worth using service S_1, however, because H_8 never used it in the past, H_8 has to rely on others' notions of trust. Table 5 shows the calculated trust values of different trustors according to their contexts. Which one is best applicable depends on the purpose for which H_8 intends to use S_1. If H_8 wants to utilize S_1 for software implementation, relying more on $team_A$'s view, especially the view of its developers is wiser. For software testing the situation is different and S_1 is less trusted than for implementation activities. Furthermore, it is obvious that trainees trust S_1 less than developers, because high experience is required to operate S_1 properly and thus S_1 has not been used with high success by unexperienced trainees in the past.

Table 5: Different trustors' trust values in service S1 depending on view and context.

Trustor	View	Contextual Restrictions	Value
all	global		0.667
all	global	activity.type=swimpl.	0.859
all	global	activity.type=swtest	0.489
all devs	global	human.role=dev	0.733
all trainees	global	human.role=trainee	0.25
teamA	team		0.859
teamA's devs	team	human.role=dev	0.9
H_8	indiv.		unknown

Scenario 2: Partner Recommendation. In this scenario we use VieTE to find suitable collaboration partners for H_8. For the global and team views the trust relationships of all entities and all members within a team respectively are considered, as in common reputation systems, and the results are not dedicated to H_8 only. However, for the individual view only direct relationships from H_8 to others are considered.

Table 6 shows a list of trustees, ordered descending by trust values calculated for particular contexts. From the global view result generally well trusted partners, however from H_8's individual view VieTE suggests only partners, with which H_8 personally in-

Table 6: H_8's trustees depending on view and context.

View	Contextual Restriction	Trustees
global		H4,H7,H1,H2,H6,H3,H5
global	activity.type=swimpl.	H1,H2,H3
global	human.role=dev	H4,H7,H1,H2
team	human.team=team$_A$	H1,H2,H3
indiv.	human.team=team$_A$	H2,H1,H3

teracted in the past, while others are out of scope. Note that the order of trustees for H_8's individual view and $team_A$'s team view are different, though contextual restrictions are set to take into account only humans from $team_A$ in both cases. This is due to the fact that H_8's trust in H_2 is quite high (0.9), while the trust of other team members in H_2 is only medium (0.6 on average), thus on $team_A$'s view H_2 is ranked lower than H_1.

8 CONCLUSIONS AND FUTURE WORK

In this paper we have discussed the role of trust and related concepts in service-oriented collaboration environments. We defined a collaboration model comprising of humans and services, and proposed an approach and a framework to automate trust determination based on monitoring interactions and utilizing profile information.

Currently we focus on trust metrics and models and extend the existing prototype to make it feasible for supporting real world scenarios in the area of networked enterprises. The challenge is the definition of suitable metrics which are able to reflect real trust relationships. After that, the next step will be to perform an empirical evaluation and to prove that the selected metrics and models appropriately address the challenges in automatic trust inference.

ACKNOWLEDGEMENTS

This work is mainly supported by the European Union through the IP project COIN (FP7-216256).

REFERENCES

Artz, D. and Gil, Y. (2007). A survey of trust in computer science and the semantic web. *J. Web Sem.*, 5(2):58–71.

Gomez, V., Kaltenbrunner, A., and Lopez, V. (2008). Statistical analysis of the social network and discussion threads in slashdot. In *WWW*, pages 645–654. ACM.

Grandison, T. and Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4).

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *WWW*, pages 403–412, New York, NY, USA. ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.

Huynh, T. D., Jennings, N. R., and Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154.

Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644.

Marsh, S. P. (1994). *Formalising trust as a computational concept*. PhD thesis, University of Stirling.

Massa, P. and Avesani, P. (2005). Controversial users demand local trust metrics: An experimental study on epinions.com community. In *AAAI*, pages 121–126.

Maximilien, E. M. and Singh, M. P. (2004). Toward autonomic web services trust and selection. In *ICSOC*, pages 212–221. ACM.

McKnight, D. H. and Chervany, N. L. (1996). The meanings of trust. Technical report.

Moody, P., Gruen, D., Muller, M. J., Tang, J. C., and Moran, T. P. (2006). Business activity patterns: A new model for collaborative business applications. *IBM Systems Journal*, 45(4):683–694.

Mui, L. (2002). *Computational models of trust and reputation: Agents, evolutionary games, and social networks*. PhD thesis, Massachusetts Institute of Technology.

Quercia, D., Hailes, S., and Capra, L. (2007). Lightweight distributed trust propagation. In *ICDM*, pages 282–291. IEEE Computer Society.

Ramchurn, S. D., Jennings, N. R., Sierra, C., and Godo, L. (2004). Devising a trust model for multi-agent interactions using confidence and reputation. *Applied Artificial Intelligence*, 18(9-10):833–852.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Ruohomaa, S. and Kutvonen, L. (2005). Trust management survey. In *iTrust*, volume 3477 of *Lecture Notes in Computer Science*, pages 77–92. Springer.

Schall, D., Truong, H.-L., and Dustdar, S. (2008). Unifying human and software services in web-scale collaborations. *IEEE Internet Computing*, 12(3):62–68.

Shand, B., Dimmock, N., and Bacon, J. (2004). Trust for ubiquitous, transparent collaboration. *Wireless Networks*, 10(6):711–721.

Theodorakopoulos, G. and Baras, J. S. (2006). On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge University Press.