Social Formation and Interactions in Evolving Service-oriented Communities

Florian Skopik, Daniel Schall, Harald Psaier, Schahram Dustdar
Distributed Systems Group
Vienna University of Technology
Argentinierstraße 8/184-1, A-1040 Vienna, Austria
{skopik|schall|psaier|dustdar}@infosys.tuwien.ac.at

***Abstract*—The global scale and distribution of companies have changed the economy and dynamics of businesses. Web-based collaborations and cross-organizational processes typically require dynamic and context-based interactions between people and services. However, finding the right partner to work on joint tasks or to solve emerging problems in such scenarios is challenging due to scale (number of involved people and services) and the temporary nature of collaborations. Furthermore, actor skills and competencies evolve over time requiring dynamic approaches for the management of actor properties. Web services and SOA are the ideal technical framework to automate interactions spanning people and services. In this paper, we present a novel discovery mechanism based on social trust to support formation and dynamic interactions in service-oriented collaboration networks. We argue that trust between members is essential for successful collaborations. Here we discuss profile similarity-based link establishment to connect disparate network segments.**

***Keywords*-interaction monitoring, trust inference, group formation, privacy issues, service-centric collaborations**

I. Introduction

Small and medium-sized organizations create alliances to compete with global players, to cope with the dynamics of economy and business, and to harvest business opportunities that a single partner cannot take. In such networks where companies, communities, and individuals form virtual organizations, collaboration support is a major research track. In this paper, we focus on using SOA to support the creation and operation of professional virtual communities (PVCs). This kind of communities – also referred to as a special case of a virtual organization – is created by individuals to facilitate the collaboration of professionals. For instance, the members of PVCs may work on new technology standards, discuss current research problems, or offer support to the economy.

Individuals and companies that are interested in collaborations register at dedicated portals, where they can flexibly discover partners to form temporal alliances [1]. The collaborations in such networks usually span numerous individuals distributed over various organizations and locations. Due to the scale of these networks it is impossible for the individuals to keep track of the dynamics in such networks. However, the recent adoption of service-oriented concepts permits the (semi-)automatic management of member profiles and network structures. In particular, SOA provides the functional means to allow loose coupling of entities through predefined interfaces and well-formed interaction messages. Upon SOA, monitoring of interactions enables the inference of social relations and expertise/interest profiles through mining logs. Hence, we use SOA to support and guide human interactions in collaborations by utilizing social relations. The automatic inference and adaptation of relations between network members [2], [3] has several advantages. Negative influences, such as using outdated information for partner discovery, do not exist compared to manually declared relations. Moreover, monitoring of interaction behavior allows timely adaptations in ongoing collaborations, for instance, updates of member profiles based on successes in recent collaborations and collected experiences, without major user intervention. This paper deals with the following contributions:

- *Social Composition Model.* We introduce concepts to enable the seamless integration of human capabilities in SOA, and the concept of social trust to support the discovery of human-provided services and their interactions.
- *Trust-based Link Establishment in Collaborative SOA.* We study the application of introduced concepts to support group formations in state-of-the-art SOA with the human user in the loop.
- *Prototype and Evaluation.* We discuss the Social SOA formation tool – a prototype implementation on top of well adopted standards, including WSDL, SOAP and FOAF (Friend-Of-A-Friend) [4], and evaluate its applicability with a SOA testbed.

The remainder of this paper is organized as follows. Sect. II motivates the need for socially enhanced SOA. We deal with an example scenario in Sect. III and introduce an interaction model for SOA communities in Sect. IV. Sect. V highlights trust mechanisms to support discovery and formation. We discuss the effectiveness of our link establishment approach in Sect. VI, and introduce a prototype implementation from the end-user's perspective in Sect. VII. Related work is outlined in Sect. VIII and the paper concluded in Sect. IX.

II. Socially Enhanced SOA Systems

While the traditional SOA concepts deem to be sufficient from a pure technological point of view, the situation changes with the human user in the loop. Considering service-oriented collaboration scenarios on the Web, here we discuss various views on socially-enhanced SOA. In Fig. 1, three main building blocks are identified that are based on traditional SOA concepts (services, discovery, and interactions). We argue that the role of humans in SOA should be extended so that people are able to shape the availability of services. Furthermore, not only software-based services are part of such systems, but also services provided by human actors.

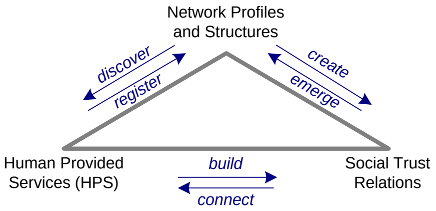


Figure 1. Social Compositions in SOA.

Human-Provided Services (HPS). HPS act as interaction interfaces toward humans [5], letting users define various HPSs for different collaborative activities indicating their ability (and willingness) to participate in ad-hoc as well as process-centric collaborations. Users can manage interactions, which might span various platforms and services. The very idea of HPS is to support humans in offering their skills and capabilities as services (e.g., a 'document review service' provided by one or more human actors). For example, human activities can be defined by the end-user and are mapped onto Web service interfaces.

Social Trust. In this paper, we focus on supporting formations and interactions in service-oriented collaboration environments by accounting for the individuals' social relations, especially *social trust*. In contrast to a common security perspective on trust, the notion of social trust refers to the interpretation of previous collaboration behavior [2], [3] and the similarity of dynamically adapting interests [6], [3]. Especially in collaborative environments, where users are exposed to higher risks than in common social network scenarios, and where business is at stake, considering social trust is essential to effectively guide human interactions.

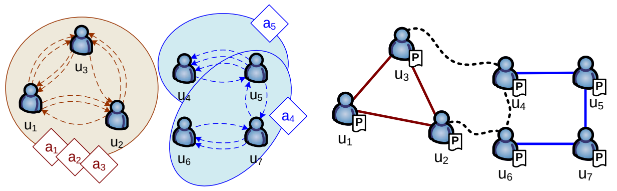
Trust-based Network Profiles. We argue that in a socially enhanced SOA, network profiles replace traditional service registries. Queries for services of collaboration partners are not only based on sole service capabilities and QoS, but increasingly personal relations are of paramount importance. Especially in social environments, provided services of personally known partners are highly favored compared to unknown third party services. Thus, we adopt common standards of the social network domain to reflect personal relations and partner properties; in particular FOAF [4]. However, we employ a system that dynamically creates and adapts FOAF structures upon inferred trust relations; thus keeping track of the dynamics in collaboration networks automatically. Network Profiles support the (i) discovery of potential collaboration partners (direct relations *and* recommendations of yet unknown actors through well known actors); (ii) routing of requests and messages in the network; (iii) creation of human-service compositions and (interest) group formation within larger communities.

III. Emerging Relations in PVCs

We depict a professional virtual community (PVC) environment to familiarize with our concepts, and to demonstrate the dynamic emergence of social relations. A PVC is a virtual community that consists of experts belonging to different physical organizations, and who interact and collaborate by the means of information and communication technologies to perform their work. Nowadays, service-oriented technologies are increasingly used to realize PVCs. The support of loose coupling, convenient discovery, dynamic binding and composition mechanisms makes SOA the ideal grounding for Web-enabled PVCs.

Fig. 2(a) depicts various member groups that collaborate in context of five different activities. The color of the activity context determines the expertise area an activity is related to. Such activities are, for instance, the specification of new technology standards or scientific dissemination. Activities are a concept to structure information in ad-hoc collaboration environments, including the goal of the ongoing tasks, involved actors, and utilized resources. They are either assigned from outside the community, e.g. belonging to a higher-level process, or emerge by identifying collaboration opportunities. In order to achieve their goals, the members of the PVC interact in context of the currently performed activities. In this paper we focus on a special type of interaction: *requests for support (RFSs)*. PVC members interact using SOA technology. In our scenario we make use of the HPS framework to allow human participation in a service oriented manner, i.e., humans can provide their capabilities as services, and enable human interactions through SOAP. All SOAP messages are logged for later analysis.

Social relations, e.g., reflected in FOAF profiles [4], emerge from interactions (Fig. 2(b)), and are bound to particular scopes (here: expertise areas). As shown later in this work, we model the interaction context with tags and keywords in order to create communities with actors in similar activities. Through analyzing interaction contexts (i.e., tags from exchanged messages that are collected in activities), we determine a community's predominant activity focus and single members' centers of interests. Frequently used keywords are stored in the actors' profiles (see symbol *P*) and later used to determine interest and expertise similarities. In the given scenario, this similarity measurement is used to support the emergence of trust between PVC members regarding help and support in



(a) Member interactions.

(b) Emerging relations.

Figure 2. Collaboration model for service-oriented PVCs: (a) interactions in context of activities; (b) emergence of relations and profiles.

different expertise areas. We manage trust relations in a directed graph model $G = (N, E)$, where nodes N represent the network members and directed edges E reflect trust relations annotated by their scope. While some kind of social relations already exist in a community, e.g., expressed through FOAF profiles, supporting the emergence of new relations becomes a paramount undertaking to form larger expert group and support the continuous growth of communities.

Trust-based Link Establishment. Consider a scenario in the given PVC in Fig. 2(b) where collaboration between one community (u_1, u_2, u_3) and another one (u_4, u_5, u_6, u_7) should be facilitated. In that case, actors from both communities should be 'connected', i.e., introduced to each other. However, not just random actors should be picked, but actors having similar interests and therefore, a common basis for future interactions (see dashed lines). We argue that establishing personal contacts in socially oriented environments is of high importance compared to the traditional SOA domain, where services are mostly composed based on their sole properties (e.g., features and QoS) only.

Let us assume we are able to infer meaningful social relations between interacting network members (as detailed in [3] and partly shown later in this paper). These relations have major impact on future collaborations in different manners:

- *Supporting the Formation of Expert Groups.* Successful previous compositions of actors should not be dissolved but actively facilitated for future collaborations. Thus, tight trust relations are dynamically converted to FOAF relations.
- *Controlling Interactions and Delegations.* Interactions and delegations of tasks between members can be guided upon FOAF relations. We argue that people tend to favor well-known members over any third parties.
- *Establishing new Social Relations.* The emergence of new personal relations is actively facilitated by establishing links. Connecting actors with similar interests (see dashed edges in Fig. 2(b)) supports the emergence of future trustworthy compositions.

IV. Human Interactions in SOA

Community members interact to reach a predefined goal. For instance, they request support, exchange information, delegate tasks, and coordinate actions to perform certain activities. Therefore, interactions always take place within certain contexts. Traditional service-oriented architectures focus on modeling and implementing interactions between distributed software-based services using Web services technology. A central part of SOA are standards such as descriptive service interfaces (WSDL) and the exchange of XML-based messages following a standardized format (SOAP). These mechanisms enable the dynamic discovery and invocation of services.

Also human interactions may rely on these SOA principles as discussed in the following. This fact enables the adoption of various available monitoring and logging tools in service-oriented collaboration systems. The XML-based structure of SOAP messages is well-suited for message header extensions, such as addressing and routing information, and annotation with contextual elements (e.g., activity identifier).

A. Human-Provided Services

As an example, an excerpt of a generic request for support (RFS) schema definition is shown in Listing 1. A user may send such a message (instance of the schema) to a HPS in case s/he needs assistance in ongoing collaborations. For that purpose the user defines the `Request`, including a `subject` and the detailed problem (`requ`), links to important `resources`, and `keywords` to categorize the message (such as the expertise area).

```
<xsd:schema tns="http://myhps.org/rfs">
 <xsd:complexType name="GenericResource">
  <xsd:sequence>
   <xsd:element name="Location" type="xsd:anyURI"/>
   <xsd:element name="Expires" type="xsd:dateTime"/>
  <xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="Request">
  <xsd:sequence>
   <xsd:element name="subject" type="xsd:string"/>
   <xsd:element name="requ" type="xsd:string"/>
   <xsd:element name="resource" type="GenericResource"/>
   <xsd:element name="keywords" type="xsd:string"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:element name="SupportRequest" type="Request"/>
 <xsd:element name="AckSupportRequ" type="xsd:string"/>
 <xsd:element name="GetSupportReply" type="xsd:string"/>
 <!-- reply details omitted -->
 <xsd:element name="SupportReply" type="Reply"/>
</xsd:schema>
```

Listing 1. RFS schema definition.

The `GenericResource` defines common attributes and metadata associated with resources such as documents or policies. A `GenericResource` can encapsulate remote resources that are hosted by a collaboration infrastructure (e.g., document management). An *interaction policy* is a special type of resource and plays an important role for controlling interaction flows, e.g., time constraints, delegation behavior including decisions whether to respond to a requester directly or to a 'social broker', and so on. `Request` defines the structure of an RFS (here we show a simplified example). A `Reply` is the corresponding RFS response (we omitted the actual XML defintion).

Listing 2 shows the binding of the HPS WSDL to the (HPS) infrastructure. The protocol (at the technical HPS middleware level) is asynchronous allowing RFSs to be stored, retrieved, and processed. For that purpose we implemented a middleware service (HPS Access Layer - HAL) which dispatches and routes RFSs.

```
<wsdl:portType name="HPSSupportPortType">
 <wsdl:operation name="GetSupport">
  <wsdl:input xmlns="http://.../addressing/wsdl"
   message="GetSupport" wsaw:Action="urn:GetSupport">
  </wsdl:input>
  <wsdl:output message="AckSupportRequ"/>
 </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="..." type="HPSSupportPortType">
 <soap:binding style="document"
  transport="http://xmlsoap.org/soap/http"/>
</wsdl:binding>
```

Listing 2. HPS WSDL binding excerpt.

GetSupport depicts a message corresponding to the RFS SupportRequest. Upon receiving such a request, HAL generates a session identifier contained in the output message AckSupportRequ. A notification is sent to the requester (assuming a callback destination or notification endpoint has been provided) to deliver RFS status updates for example; processed RFSs can be retrieved via GetSupportReply. More information about this notification mechanism can be found in [5].

B. Activity-based Interaction Context Model

Fig. 3 depicts the applied context model (simplified for brevity), where actors, described by their profiles, perform activities.

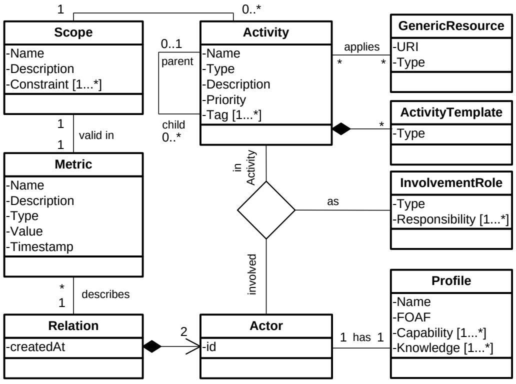


Figure 3. Context model: Linked Actors perform activities in scopes.

Activities reside in more abstract scopes, e.g., all activities of a specific type (activity scope), or all activities belonging to a certain project (project scope). For instance, supporting the creation of white box test cases resides in a software development scope. Furthermore, actors are linked to collaboration partners in the network. These relations are reflected by FOAF profiles, are bound to scopes, and are characterized by various metrics that rely on previous interactions.

V. Social Trust in Collaborative SOA

Collaborative networks as outlined in the previous sections are subject to our trust studies. Unlike a security view, we focus on the notion of dynamic trust from a social perspective [7]. We argue that trust between community members is inevitable for successful collaborations. The notion of social trust considers the similarity of dynamically adapting skills and interests [6], [8]. In this paper, we particularly focus on the establishment of trust through measuring interest similarities [3], [6], [7]:

- *Trust Mirroring* implies that actors with similar profiles (interests, skills, community membership) tend to trust each other more than completely unknown actors.
- *Trust Teleportation* rests on the similarity of human or service capabilities, and describes that trust in a member of a certain community can be teleported to other members. For instance, if an actor, belonging to a certain expert group, is trusted because of his distinguished knowledge, other members of the same group may benefit from this trust relation as well.

A. Profile Similarity Measurement

In contrast to common top-down approaches that apply taxonomies and ontologies to define certain skill profiles and expertise areas, we follow a mining approach that addresses inherent dynamics of flexible collaboration environments. In particular, skills, expertise and interests change over time, but are rarely updated if they are managed manually in registries. Hence, we determine and update them automatically through interaction mining. As discussed before, interactions, such as task delegations and support requests are tagged with keywords. These keywords contribute to the description of activities, i.e., describe their focus. As actors process or discard received messages, our system is able to learn their expertise and centers of interests. We use task keywords to create dynamically adapting interest profiles based on tags and manage them in a vector space model [9].

We assume that users pick keywords from a globally available taxonomy (such as the ACM taxonomy[1]) instead of adding arbitrary tags. The advantage is that we avoid (i) the use of synonyms, thus leading to inaccurate interest profiles (notebook v.s. laptop both meaning the same), (ii) equally meant but differently written tags (and their singular/plural forms), e.g., social network v.s. social-networks). An approach to similarity measurement that compensates such influences has been discussed in [10].

The profile vector $\mathbf{p_{u_i}}$ of actor u_i in Eq. (1) describes the frequencies f the tags $T = \{t_1, t_2, t_3 \dots\}$ are used in requests and delegated tasks accepted by actor u_i.

$$\mathbf{p_{u_i}} = \langle f(t_1), f(t_2), f(t_3) \dots \rangle \qquad (1)$$

The tag frequency matrix $\mathfrak{T}$ (2) in Eq. 2, built from profile vectors, describes the frequencies of used tags $T = \{t_1, t_2, t_3 \dots\}$ by all actors $N = \{u_1, u_2, u_3 \dots\}$.

$$\mathfrak{T} = \langle \mathbf{p_{u_1}}, \mathbf{p_{u_2}}, \mathbf{p_{u_3}} \dots \rangle_{|T| \times |N|} \qquad (2)$$

The popular $tf^{*}idf$ model [9] introduces tag weighting based on the relative distinctiveness of tags; see Eq. (3). Each entry in $\mathfrak{T}$ is weighted by the log of the total number of actors $|N|$, divided by the amount $n_t = |\{u_i \in N \mid tf(t, u_i) > 0\}|$ of actors who used tag t.

$$tf^{*}idf(t, u_i) = tf(t, u_i) \cdot \log \frac{|N|}{n_t} \qquad (3)$$

Finally, the cosine similarity, a popular measure to determine the similarity of two vectors in a vector space model, is applied to determine the similarity of two actor profiles $\mathbf{p_{u_i}}$ and $\mathbf{p_{u_j}}$; see Eq. (4). The result is a real value $sim_p \in [0, 1]$, whereas 0 denotes no overlap between used tags and 1 reflects identically used keywords.

$$sim_p(\mathbf{p_{u_i}}, \mathbf{p_{u_j}}) = \cos(\mathbf{p_{u_i}}, \mathbf{p_{u_j}}) = \frac{\mathbf{p_{u_i}} \cdot \mathbf{p_{u_j}}}{||\mathbf{p_{u_i}}||\;||\mathbf{p_{u_j}}||} \qquad (4)$$

[1] http://www.acm.org/about/class/1998/

B. The Interplay of Interest Similarity and Trust

In our model, trust $\tau(u_i, u_j) \in [0,1]$ mainly relies on the interest and expertise similarities of actors (see [6] for details on that assumption). We two major concepts to facilitate the emergence of trust among network members.

Trust Mirroring. Trust τ_{mir} is typically applied in environments where actors have the same roles (e.g., online social platforms). Depending on the environment, interest and competency similarities of people can be interpreted directly as an indicator for future trust (Eq. 5). There is strong evidence that actors 'similar minded' tend to trust each other more than any random actors [7], [8]; e.g., movie recommendations of people with same interests are usually more trustworthy than the opinions of unknown persons.

$$\tau_{mir}(u_i, u_j) = sim_p(\mathbf{p_{u_i}}, \mathbf{p_{u_j}}) \tag{5}$$

Trust Teleportation. Trust τ_{tele} is applied in sparse trust networks. We assume that u_i has established a trust relationship to u_j in the past, for example, relying on trust mirroring (applied in the following experiments) or based on u_j's capabilities to assist u_i in work activities (see for instance [3]). Therefore, others having interests and capabilities similar to u_j may become similarly trusted by u_i in the future. In contrast to mirroring, trust teleportation may also be applied in environments comprising actors with different roles. For example, a manager might trust a software developer belonging to a certain group. Other members in the same group may benefit from the existing trust relationship by being recommended as trustworthy as well. We attempt to predict the amount of future trust from u_i to a third party u_k by attenuating $\tau(u_i, u_j)$ considering the profile similarity of the trustee u_j and the still unknown actor u_k. Since there may be multiple recommendations, in Eq. 6 the degree of teleported trust is additionally weighted by the profile similarities (sim_p) of u_i and each actor in the set of recommenders M'.

$$\tau_{tele}(u_i, u_k) = \frac{\sum_{u_j \in M'} \tau(u_i, u_j) \cdot (sim_p(\mathbf{p_{u_j}}, \mathbf{p_{u_k}}))^2}{\sum_{u_j \in M'} sim_p(\mathbf{p_{u_j}}, \mathbf{p_{u_k}})} \tag{6}$$

Eq. 6 deals with a generalized case where several trust relations from u_i to members of a group M' are teleported to a still untrusted actor u_k. Teleported relations are weighted and attenuated by the similarity measurement results of actor profiles.

C. Establishment of Social Relations

Based on a pre-configured profile similarity threshold $\vartheta_T \in [0,1]$ the system can recommend new links. These links reflect potentially beneficial relations due to actors' interest similarities. Setting $\vartheta_T = 0$ means that all actors will be connected, thus resulting in a fully meshed network; setting $\vartheta_T = 1$ means that virtually no new relations will be introduced (except entire identical tagging profiles). Appropriate top and bottom limits are determined in the evaluation in Sect. VI. Practically, there should be enough links introduced to connect yet unconnected subcommunities, however, still considering their differing interests.

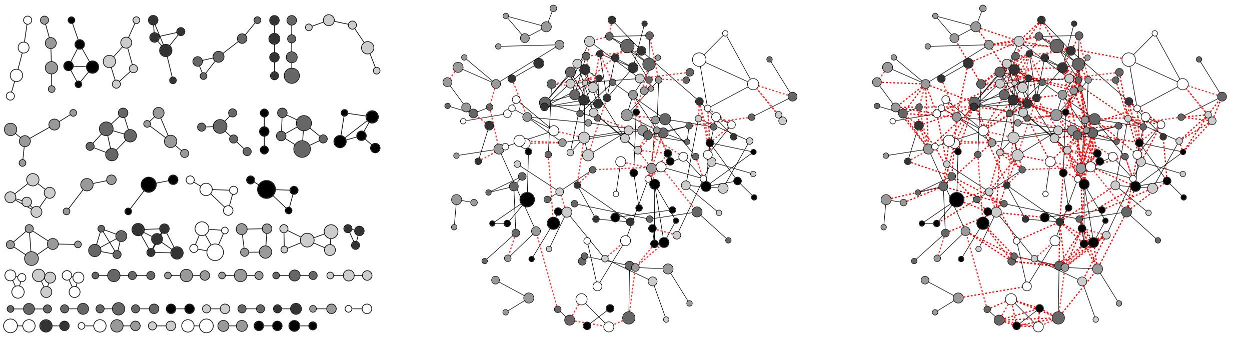
VI. Evaluation and Discussion

We use a Web service testbed to simulate the interaction behavior in a SOA-based PVC. The purpose of the Genesis2 framework [11] (in short, G2) is to support software engineers in setting up testbeds for runtime evaluation of SOA-based concepts and implementations. It allows to establish environments consisting of services, clients, registries, and other SOA components, to program the structure and behavior of the whole testbed, and to steer the execution of test cases on-the-fly. G2's most distinct feature is its ability to generate real testbed instances (instead of just performing simulations) which allows engineers to integrate these testbeds into existing SOA environments and, based on these infrastructures, to perform realistic tests at runtime.

Experiment Setup. The created environment consists of 200 services that interact in small groups of 2 to 5 members; thus 58 groups are built. Typically, groups of that size perform certain activities. During collaboration services interact by delegating tasks and requesting support; thus, in our simulation we let random services interact in fixed time intervals. Each interaction is tagged with a maximum of 3 keywords. We run different tests and vary the number of globally known tags, as well as the amount of occurring interactions. The results of these experiments help to determine appropriate similarity thresholds to introduce new (trust) edges in the collaboration graph for recommending and facilitating future collaborations.

Results. Fig. 4 demonstrates the effects on the graph structure when new links are introduced (red dashed lines). The size of the nodes denote their involvement in activities, i.e., the number of received interactions. Additionally the single groups are colored for better visibility. In the beginning (Fig. 4(a)) various small components exist but are not interconnected. These components represent small groups of actors that interact in context of their activities. Links reflect interaction paths that may lead to trust over time (see [3]). After finishing the simulation, we gradually introduce new links using our concepts of *trust mirroring* and *trust teleportation*. The threshold ϑ_T denotes the lower boundary of tag usage similarity, i.e., all pairs of actors that have higher profile similarity than ϑ_T are connected. Thus, higher ϑ_T leads to less connections. The optimal number of introduced edges in the graph depends on several properties. On the one side, independent components should be connected, so that previously unknown actors get introduced to each other. On the other side, simply connecting all actors with each other is obviously not beneficial. An optimal connection is hard to determine, but various graph metrics [12] are appropriate indicators, such as *number of connected components* nc, *average number of neighbors* nn, or *network density* nd.

In the following experiments, we determine the number of added edges depending on the configured threshold



(a) No added edges (initial) (metrics: nc=55, nn=1.8, nd=0.010).

(b) Added edges ($\vartheta_T = 0.8$) (metrics: nc=8, nn=2.62, nd=0.014).

(c) Added edges ($\vartheta_T = 0.6$) (metrics: nc=1, nn=5.56, nd=0.03).

Figure 4. Gradually interconnecting trust network based on evolving interest similarities.

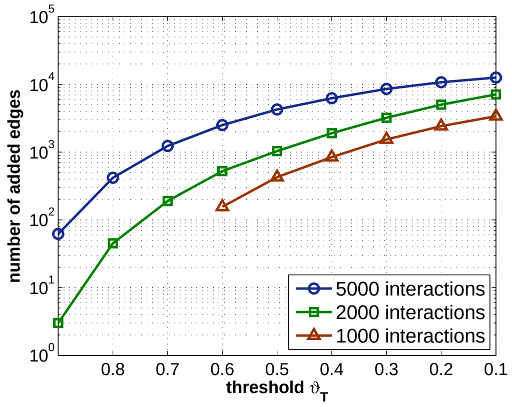


Figure 5. Impact of ϑ_T on number of added edges for varying number of interactions (20 tags).

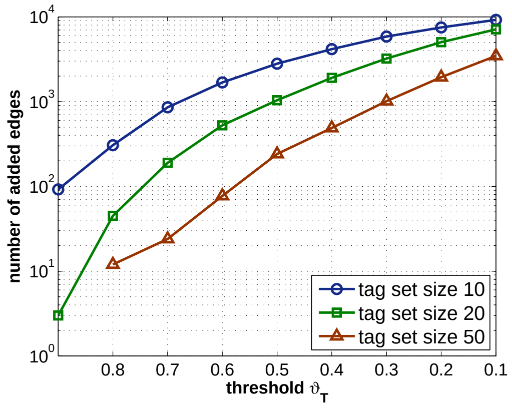


Figure 6. Impact of ϑ_T on number of added edges for varying tag sets (2000 interactions).

ϑ_T. The first set of experiments investigates the impact of varying numbers of interactions in our scenario (see results in Fig. 5). Actors pick up to three tags from a globally available tag set of size 20 to annotate their interactions. Obviously more interactions lead to bigger profiles as more tags are collected. Therefore, after longer collaboration (e.g., 5000 interactions in the whole scenario) more similar actors can be determined than after a lower amount of interactions (e.g., 2000). For only 1000 interactions a threshold of 0.6 is not exceeded. Note, numbers on the x-axis are in the reverse order. Normally, one would start with introducing links between actors with identical profiles ($\vartheta_T = 1$) and than gradually degrade that value until a satisfying degree of connection has been reached. Also note that the y-axis uses a logarithmic scale. In the second set of experiments, 2000 interactions are performed, however, the number of globally available tags is changed. This means that actors can choose from 10, 20 or 50 different keywords to annotate their interactions. As expected, for smaller tag sets higher profile similarity is achieved (see results in Fig. 6).

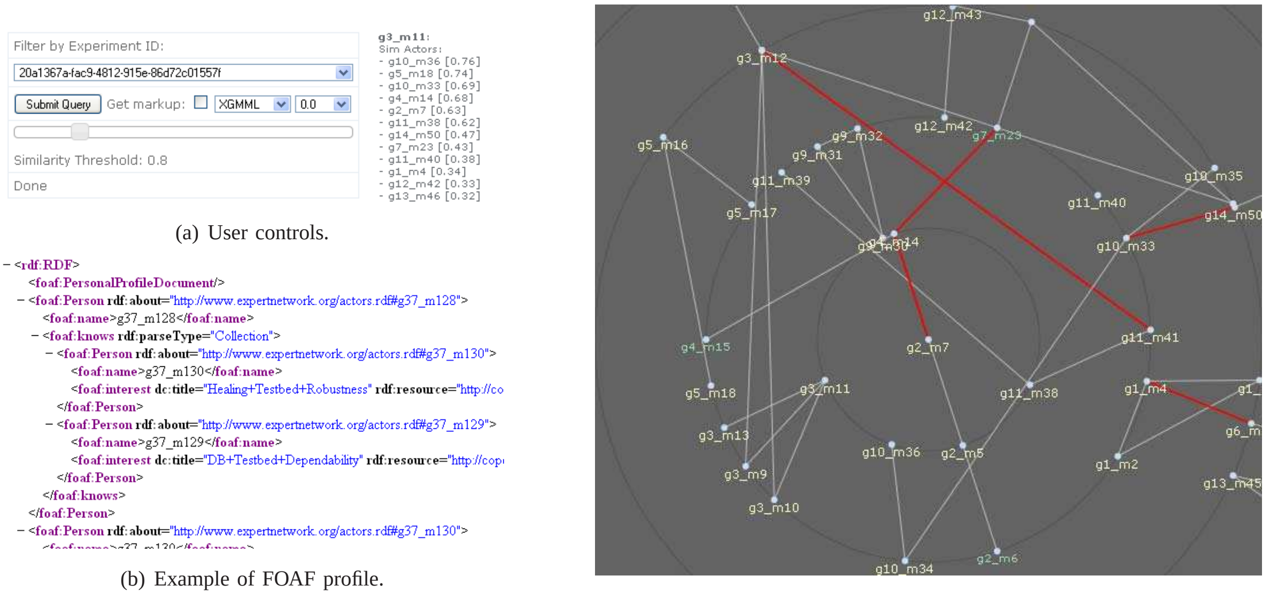
VII. Prototype and Implementation

A. Interaction Logging

The previously presented results are based on G2's testbed generation capabilities and a framework for monitoring and logging interactions between services. Interactions are captured through (SOAP) message interceptors deployed within the service runtime environment. Logged messages are persistently saved in a database for analysis. An example interaction log is shown by Listing 3, which includes various SOAP header extensions for message correlation and context-aware interaction analysis.

```
<soap:Envelope
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:vietypes="http://viete.infosys.tuwien.ac.at/Type"
 xmlns:hps="http://www.infosys.tuwien.ac.at/hps/"
 xmlns:rfs="http://.../socialsoa/rfs">
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:wsa="http://schemas.xmlsoap.org/ws/.../addressing"
 <soap:Header>
  <vietypes:timestamp value="2010-05-25T17:24:18"/>
  <vietypes:msgflags priority="urgent"/>
  <vietypes:activity url="http://.../Activity#42"/>
  <wsa:MessageID>uuid</wsa:MessageID>
  <wsa:ReplyTo>http://.../Actor#Harald</wsa:ReplyTo>
  <wsa:From>http://.../Actor#Florian</wsa:From>
  <wsa:To>http://.../Actor#Daniel</wsa:To>
  <wsa:Action>http://.../Type/RFS</wsa:Action>
 </soap:Header>
 <soap:Body>
  <hps:Request>
   <rfs:subject>ACM taxonomy for my paper?</rfs:subject>
   <rfs:requ>What ACM categories fit best
             to my paper?</rfs:requ>
   <rfs:resource>
    <vietypes:resource type="doc" uri="http://..."/>
   </rfs:resource>
   <rfs:keywords>document, categorization</rfs:keywords>
  </hps:Request>
 </soap:Body>
</soap:Envelope>
```

Listing 3. Simplified RFS via SOAP example.



Filter by Experiment ID:

20a1367a-fac9-4812-915e-86d72c01557f

Submit Query Get markup: □ XGMML 0.0

Similarity Threshold: 0.8

Done

```
g3_m11:
Sim Actors:
- g10_m36 [0.76]
- g5_m18 [0.74]
- g10_m33 [0.69]
- g4_m14 [0.68]
- g2_m7 [0.63]
- g11_m38 [0.62]
- g14_m50 [0.47]
- g7_m23 [0.43]
- g11_m40 [0.38]
- g1_m4 [0.34]
- g12_m42 [0.33]
- g13_m46 [0.32]
```

(a) User controls.

```
– <rdf:RDF>
    <foaf:PersonalProfileDocument/>
  – <foaf:Person rdf:about="http://www.expertnetwork.org/actors.rdf#g37_m128">
      <foaf:name>g37_m128</foaf:name>
    – <foaf:knows rdf:parseType="Collection">
      – <foaf:Person rdf:about="http://www.expertnetwork.org/actors.rdf#g37_m130">
          <foaf:name>g37_m130</foaf:name>
          <foaf:interest dc:title="Healing+Testbed+Robustness" rdf:resource="http://co
        </foaf:Person>
      – <foaf:Person rdf:about="http://www.expertnetwork.org/actors.rdf#g37_m129">
          <foaf:name>g37_m129</foaf:name>
          <foaf:interest dc:title="DB+Testbed+Dependability" rdf:resource="http://cop
        </foaf:Person>
      </foaf:knows>
    </foaf:Person>
  – <foaf:Person rdf:about="http://www.expertnetwork.org/actors.rdf#g37_m130">
      <foaf:name>g37_m130</foaf:name>
```

(b) Example of FOAF profile.

(c) Network visualization view.

Figure 7. Web-based formation tool and network visualization.

The most important extensions are (see [3] for details on the implementation):

- `Timestamp` captures the actual creation of the message and is used to calculate temporal interaction metrics, such as average response times.
- `Message flags`, including priority of messages.
- `Activity uri` describes the context of interactions (see Fig. 3 for the model).
- `MessageID` enables message correlation, i.e., to properly match requests and responses.
- `WS-Addressing` extensions, besides `MessageID`, are used to route requests through the network.

The SOAP body transports the actually exchanged message. In this example a request for support (RFS) [3] shows how one actor requests some help from another one in the motivating collaboration scenario. Note, interactions are only captured to collect keywords and support the creation of user profiles. Logged data can be purged immediately after keyword extraction. Thus, our approach of interaction observation is less intrusive compared to others (e.g., semantic analysis of captured messages). We understand today's privacy concerns as a big issue of most systems that log user data for adaptation purposes (such as establishing network links).

B. User Tools

The implemented prototype includes a Web-based formation tool assisting users in analyzing various thresholds for trust-based link establishment between independent networks. Fig. 7 shows screenshots of the tool and an example FOAF profile that can be retrieved from the Web application. All user interfaces have been implemented using state-of-the-art Web technologies such as ASP.NET MVC hosted by a .NET 3.5 runtime. Our implementation comprises a network visualization view built on top of a JavaScript library[2]. The network view is obtained by mapping raw SOAP-interactions into a graph representation composed of nodes (services) and edges (interaction links). Each link holds additional data such as the number of exchanged messages between services. Nodes are associated with profiles and also groups indicated by a prefix in the view in Fig. 7(c) representing the initial disconnected components of the interaction network.

As a first step, the user accesses information captured from the service-oriented collaboration environment (Fig. 7(a)). In our implementation, this is performed by selecting a particular set of logs which are associated with an `Experiment ID`. After issuing the corresponding query, a graph is visualized typically consisting of several disconnected components. The tool queries a `Similarity Service` to obtain a set of similar actors for each node in the network (see list on the right side in Fig. 7(a)). The presented list shows actor name and degree of similarity. By default, the collaboration network is visualized in a graph view as depicted in Fig. 7(b). The user is able to select a similarity threshold by moving a slider bar. A reduced (demanded) similarity threshold results in trust edges being added to the visualization (*color online*: depicted as red colored edges between nodes). Alternatively, interactions can be retrieved as FOAF profiles (see Fig. 7(c)) that include `foaf:interest` tags. This mechanism can be used to retrieve and aggregate captured profiles from distributed environments (e.g., from multiple instances of the logging service).

VIII. Related Work

The concept of virtual organizations and professional communities that are supported by ICT is widely studied, e.g., see [1]. While others discuss such environments from a business or workflow perspective [13], we apply

[2] Graph visualizations for the Web: `http://thejit.org`.

concepts from SOA and the social network domain. For example, major software vendors have been working on standardizing human interactions in business-centric applications (e.g., see WS-HumanTask [14]). Instead, here we focus on dynamic interaction scenarios based on social concepts and dynamic trust in collaborative service-oriented systems. In particular, we adopt the concept of *Human Provided Services* [5], [15] to support flexible service-oriented collaboration across multiple organizations and domains. A similar view is shared by [16] who defines *emergent collectives* which are networks of interlinked valued nodes (services). Furthermore, we adopt the well-known FOAF standard for managing collaboration links. Work by [17], [18] discusses link prediction based on similarity, focusing on structural graph properties such as number of neighbors and number of in/out links. However, in our model, these links reflect social trust relations. Until now, a wide range of computational trust models have been proposed [2], [3], [19], [20]. In particular, we focus on social trust [3], [6], [7], [8] that relies on the similarity of user profiles that express capabilities and interests. Especially [6] proofed with data from real systems that trust between users emerges based on interest similarities. We adopt this finding to justify our approach of link establishment. In contrast to many common top-down approaches that model user profiles at least partly by the means of ontologies [21], [22], we create interest profiles fully dynamically through mining tagged interactions. Tagging and its meaning has been studied by [23]. While others create tagging profiles with hierarchical clustering models, we apply a more lightweight approach using various analytical models from the domain of information retrieval, including term-frequency and inverse document frequency metrics [9].

IX. Conclusion and Future Work

In this paper, we discussed concepts and mechanisms to enable service-oriented virtual communities. These communities rely on SOA technology and enable humans to collaborate in a service-oriented manner. Existing approaches in service-oriented systems typically aim at devising a predefined interaction model between people and services. The presented work attempts to align SOA concepts and service-oriented collaborations driven by dynamics such as evolving skills and preferences. We believe that the automated management of social aspects including trust are key issues. Since personal relations are of paramount importance in these social environments, we introduced the concept of social trust to establish links between community members. Besides interest similarities, our future work considers more diverse collaboration metrics to capture and predict interaction behavior and attitude.

Acknowledgements

This work was supported by the European Union FP7 projects COMPAS (215175), COIN (216256) and SCube (215483).

References

[1] L. M. Camarinha-Matos and H. Afsarmanesh, “Collaborative networks,” in *PROLAMAT*, 2006, pp. 26–40.

[2] L. Mui, M. Mohtashemi, and A. Halberstadt, “A computational model of trust and reputation for e-businesses,” in *HICSS*, 2002, p. 188.

[3] F. Skopik, D. Schall, and S. Dustdar, “Modeling and mining of dynamic trust in complex service-oriented systems,” *Information Systems*, vol. 35, Nov 2010.

[4] D. Brickley and L. Miller, “Foaf vocabulary specification 0.97,” 2010.

[5] D. Schall, H.-L. Truong, and S. Dustdar, “Unifying human and software services in web-scale collaborations,” *IEEE Internet Computing*, vol. 12, no. 3, pp. 62–68, 2008.

[6] J. Golbeck, “Trust and nuanced profile similarity in online social networks,” *ACM Transactions on the Web*, vol. 3, no. 4, pp. 1–33, 2009.

[7] C.-N. Ziegler and J. Golbeck, “Investigating interactions of trust and interest similarity.” *Decision Support Systems*, vol. 43, no. 2, pp. 460–475, 2007.

[8] Y. Matsuo and H. Yamamoto, “Community gravity: Measuring bidirectional effects by trust and rating on online social networks,” in *World Wide Web Conference*, 2009, pp. 751–760.

[9] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Proc. and Mgmt.*, vol. 24, no. 5, pp. 513–523, 1988.

[10] F. Skopik, D. Schall, and S. Dustdar, “Start trusting strangers? bootstrapping and prediction of trust,” in *International Conference on Web Information Systems Engineering*, 2009, pp. 275–289.

[11] L. Juszczyk and S. Dustdar, “Script-based generation of dynamic testbeds for soa,” in *International Conference on Web Services*, 2010.

[12] H. C. Romesburg, *Cluster Analysis for Researchers*. Krieger Pub. Co., 2004.

[13] A. Marconi, M. Pistore, A. Sirbu, H. Eberle, F. Leymann, and T. Unger, “Enabling adaptation of pervasive flows: Built-in contextual adaptation,” in *International Conference on Service-oriented Computing*, 2009, pp. 445–454.

[14] M. Amend et al., “Web Services Human Task (WS-HumanTask), Version 1.0.” 2007.

[15] D. Schall, S. Dustdar, and M. B. Blake, “Programming human and software-based web services,” *Computer*, vol. 43, pp. 82–85, 2010.

[16] C. Petrie, “Plenty of room outside the firm,” *IEEE Internet Computing*, vol. 14, 2010.

[17] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social networks,” in *Int’l Conf. on Information and Knowledge Management*. ACM, 2003, pp. 556–559.

[18] C.-W. Hang and M. P. Singh, “Trust-based recommendation based on graph similarity,” in *AAMAS Workshop on Trust in Agent Societies (Trust)*, 2010.

[19] D. Artz and Y. Gil, “A survey of trust in computer science and the semantic web.” *J. Web Sem.*, vol. 5, no. 2, pp. 58–71, 2007.

[20] L. Li, Y. Wang, and E.-P. Lim, “Trust-oriented composite service selection and discovery,” in *International Conference on Service-oriented Computing*, 2009, pp. 50–67.

[21] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt, “Capturing knowledge of user preferences: ontologies in recommender systems,” in *International Conference on Knowledge Capture*. ACM, 2001, pp. 100–107.

[22] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, “Ontological user profiling in recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 54–88, 2004.

[23] S. A. Golder and B. A. Huberman, “The structure of collaborative tagging systems,” *The Journal of Information Science*, 2006.