Managing Social Overlay Networks in Semantic Open Enterprise Systems

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

ABSTRACT

Cross-enterprise collaboration has emerged as a key survival factor in today's global markets. Semantic Web technologies are the basis to establish enterprise interoperability including data mediation support and automatic composition of services. Capabilities of services are semantically described and reasoning techniques support the discovery and selection of services at run-time. These technologies are commonly based on precisely defined enterprise ontologies. In contrast to Semantic Web technologies that cover interactions between (technical) services, human collaborations emerge based on *social preferences*. Social networks have become a mass phenomenon. The fundamental aspects of these networks are to manage personal contacts and to share profile information with friends. These principles are increasingly harnessed in businesses and professional environments. In a manner similar to service-oriented systems, they enable flexible discovery and dynamic collaborations between participants. In this paper, we discuss the concept of social overlays for Web service based collaboration infrastructures. This mechanism enables information flows between actors in order to allow for flexible group formations in highly dynamic large-scale networks.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based Services; H.4 [**Information Systems Applications**]: Miscellaneous; I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination

General Terms

Human Factors, Management, Measurement

Keywords

Semantic Service-Oriented Collaboration, Social Networks, Formation

WIMS '11, May 25-27, 2011, Sogndal, Norway

1. INTRODUCTION

The rapid advancement of ICT-enabled infrastructure has fundamentally changed how businesses and companies operate. Global markets and the requirement for rapid innovation demand for alliances between individual companies. Such alliances are created on different scales ranging from short- to long-term. A long-term alliance is typically a merger of companies or individual organizational units. Short- to mid-term alliances are commonly created to perform joint collaborations with the goal of fulfilling business objectives. Organizations have become *open enterprises systems* (OES) that offer capabilities as services. Capabilities can be discovered and composed to form new alliances. However, such systems do not only span automated interactions among (technical) services, but require humans actors to be in the loop. Today's Web applications facilitate interactive knowledge sharing, information exchange, user-centered content creation, and collaboration on the WWW. Even in business environments, *Web 2.0* tools increasingly provide users the free choice to interact or collaborate with each other in virtual communities. The Web becomes thereby a medium of interwoven human and service interactions. These principles have also changed models for computing on the Web by utilizing human manpower through crowdsourcing platforms (e.g., Amazon Mechanical Turk [4]).

There are two obstacles hampering the establishment of seamless communications and collaborations across organizational boundaries: (i) the dynamic discovery and composition of resources and services, and (ii) flexible and context-aware interactions between people residing in different departments and companies. Here we address challenges related to human interactions in dynamic service-oriented systems. Semantic technologies and platforms [5] provide the means to automate the discovery and interactions of compositions. Semantically-enriched collaboration services provide the means for flexible interaction support. The technical composition layer of a service-oriented system (SOA) has received considerable attention in recent years from both the research community and industry. Considerably less attention was devoted to human aspects and interaction preferences in such systems. For example, people use services to perform collaborations.

We focus on *social aspects* in cross-organizational collaborations enabled by SOA. In order to take advantage of social preferences, we propose social network principles to overcome limited information flows in collaborative environments. Social interactions between network members allows to influence and control information flows.

Challenges and Approach Outline In this work we address challenges related to the automated management of social network based on interactions in cross-organizational collaborations.

- Top-down composition and interaction model are typically designed for long-term use. Dynamic environments that are short- to medium-lived such as open enterprise systems require *dynamic interaction models*. Flexible interactions with the purpose of communicating, coordinating, and collaborating need to be supported in a service-oriented manner.
- Theories found in social network analysis are promising candidate techniques to support flexible interactions. Since interactions take place dynamically, capturing the purpose and context of interactions to infer meaningful social relations remains challenging.
- Social network principles such as formation algorithms help to overcome limited information exchange in separated collaborative networks through propagation of profile data. From the technical point of view, adaptive information flows need to be supported using services technology. Information needs to be discovered and exchanged based on the underlying social network.

The Semantic Web and related technologies have made important contributions to pave the way towards the effective interoperability of enterprise systems and infrastructures. Due to the proliferation of Web 2.0 collaboration principles and Semantic Web technologies, a combination of these approaches seems to be promising to create novel cross-enterprise collaboration systems. In the following we give an outline of our approach.

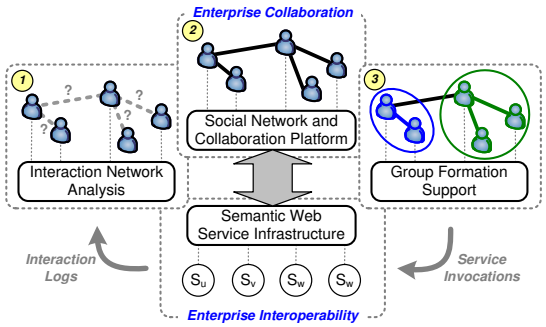


Figure 1: Enterprise collaboration and interoperability through social and Semantic Web techniques.

Figure 1 illustrates the fundamental motivation of applying *and* combining Semantic Web methodologies with Web 2.0 concepts. We show two main building blocks (i) *Enterprise Interoperability* and (ii) *Enterprise Collaboration* to support a seamless service-oriented infrastructure for cross-organizational collaboration in open enterprise systems. The *Semantic Web Service Infrastructure* provides the means to enable efficient and dynamic interactions spanning humans that belong to different organizational units. Underneath, Web services build an abstraction mechanism for intra-organizational infrastructures and resources and therefore, are the ideal technical grounding to enable interactions across organizational boundaries. Observing interactions and collecting collaboration data (*Interaction Network Analysis*) helps to support humans in building up new relationships by recommending new partners or notifying about possibly interesting business opportunities. A *Social Network and Collaboration Platform* allows people to manage their personal contacts and interact with well-known collaboration partners in context of certain projects. *Group Formation Support* concepts applied in collaborative networks allow actors to discover unconnected members using profile information, to build alliances, and to dynamically establish reliable information flows in order to exchange profiles.

Our Contributions. In this paper we deal with:

- *Cross-Organizational Application Model.* Cross-organizational scenarios are supported considering social aspects of interacting humans on the Web and technological interoperability using Semantic Web concepts.
- *Group Formation.* Formation is typically based upon sophisticated member discovery techniques. Thus, enabling actors to share personal profiles and information in a trustworthy manner is a key concept of our work. We discuss a social trust based access control (TBAC) mechanism that accounts for dynamically changing trust relations.
- *Specification and Implementation.* We discuss the implementation of social overlay networks using today's Web technologies, including Semantic Web services, interaction mining techniques, public key infrastructures, and the Friend-Of-A-Friend (FOAF) ontology.
- *Evaluation and Discussion.* We evaluate proposed models and their application in virtual communities, and derive general findings for designing applications for socially-enhanced service-oriented environments.

Structure of this Work. The remainder of this paper is organized as follows. In Section 2 we outline our approach of linking Semantic Web paradigms with social network concepts, and introduce a large-scale collaboration platform utilizing techniques from both domains. Concepts for distributed social network management are further presented in Section 3. We specify and implement this system as shown in Section 4. Then, we evaluate and discuss our work in Section 5. Section 6 deals with related work and Section 7 concludes the paper.

2. SOCIAL OVERLAYS IN SEMANTIC SOA

Enterprise collaboration and interoperability services are going to become an invisible, pervasive, and self-adaptive knowledge and business utility for any industrial sector and domain. The goal is to enable rapid set-up, efficient management and effective operation of different forms of business collaborations, from the most traditionally supply chains to the most advanced and dynamic business ecosystems. Figure 3 shows an overview of our layered approach to enable reliable and flexible formation of collaboration groups: (i) the *Service Layer* provides the technical infrastructure to semantically describe and host Web services in order to enable cross-organizational collaborations; (ii) the *Interaction Layer* provides the means of Web service-based human interactions; e.g., allows actors to communicate and collaborate with others using dedicated services from the bottom layer; (iii) the *Monitoring Layer*, observes interactions collected

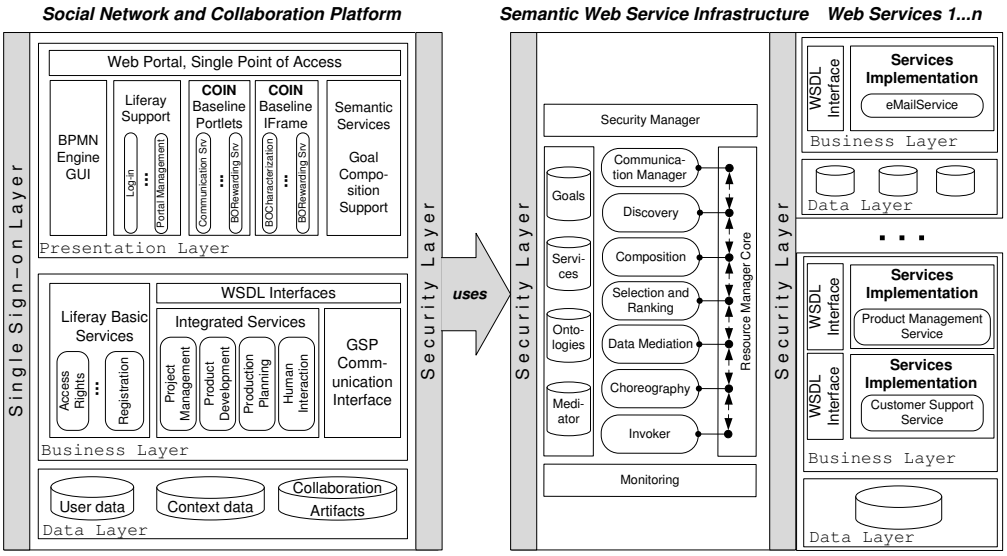


Figure 2: The COIN Framework enabling cross-organizational collaboration and interoperability.

from various sources (i.e., interaction services); and (iv) the *Discovery Layer* discovers social relations gathered through mining of interactions and profile properties, and supports group formation based on evaluating network links.

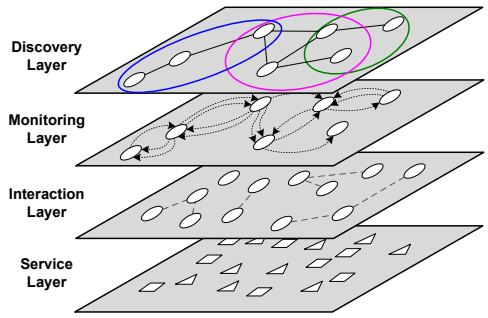


Figure 3: Model for social overlay networks.

2.1 Semantic Web Service Infrastructure

In order to realize the vision of cross-organizational collaboration and interoperability, various multi-national research projects, such as within the EU Seventh Framework Program[1], are conducted. The COIN project[2], where our contributions of this paper are embedded, aims at developing a basic platform for future Web based cross-organizational collaborations. In the following, we discuss the architectural model of semantically-enriched social OESs and outline utilized major concepts on each layer.

The COIN project aims at providing an open, self-adaptive integrative solution for *Enterprise Interoperability* and *Enterprise Collaboration*. Service orientation is a well-suited and widely adopted concept in collaboration scenarios, therefore, COIN utilizes state of the art SOA concepts, including Semantic Web technologies and Software-as-a-Service (SaaS) models (see [17] for more details). With respect to Enterprise Collaboration, COIN supports numerous features that focus on product development, production planning and manufacturing, and project management in networks of enterprises. As a fundamental aspect, human interactions exist in all forms and phases of virtual organizations and play a major role in the success of collaborations within open enterprise networks. Therefore, understanding human interactions and providing advanced support for efficient and effective interactions, is one of the key objectives in COIN's Enterprise Collaboration research track.

The COIN Framework (see Figure 2) consists of (i) the Social Network and Collaboration Platform (SCP) that provides fundamental features that are required in (nearly) every collaboration scenario, and (ii) a Semantic Web Service Infrastructure (SSI) that allows extensions with services following the SaaS model from third party providers. The SCP is designed for and tightly coupled to a community portal that provides an effective way to configure and personalize the SCP for specific end-users by providing customized services and tools. Single sign-on- and security mechanisms span services and tools across layers. The SSI relies on Semantic Web technologies, implemented by the Web Service Modeling eXecution environment (WSMX)[3] [19] and is utilized to discover, bind, compose, and use third-party services at run time. Because of its extensibility and configurability, the COIN platform can be applied in a wide variety of different collaboration scenarios, ranging from traditional production planning to social campaigning and interest group formations in professional virtual communities. For enabling context-aware interactions, the following baseline components are of major interest (i) user data, including skills and interest profiles, (ii) context data, such as current ongoing activities and user preferences, (iii) integrated baseline services for communication and coordination (e.g., e-mail notifications, and instant messengers), (iv) the SCP as the platform to host extended human interaction services.

2.2 Human Interaction Layer

Open enterprise systems that allow to form virtual organizations pose additional challenges to human interaction

[1] http://cordis.europa.eu/fp7
[2] http://www.coin-ip.eu
[3] http://www.wsmx.org

support. Typically such virtual organizations are temporary alliances that form and dissolve again. Various actors from different physical organizations are involved collaborating and working on joint activities. Figure 4 shows a semantic representation (i.e., an ontology) of utilized concepts, grouped in communication, coordination and collaboration entities.

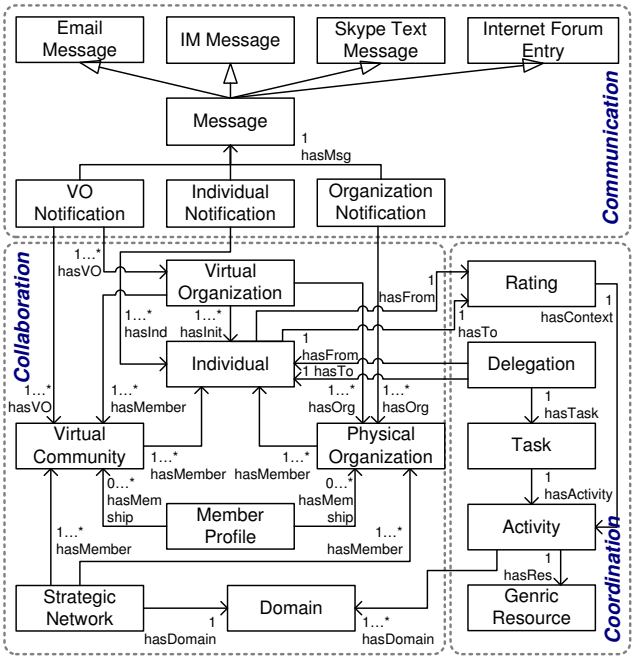


Figure 4: Enterprise collaboration ontology.

Various artifacts need to be created in order to integrate common WSDL-based Web services into the Semantic Web infrastructure of WSMX [38]. We provide a basic description that acts as the underlying basis for the rest of this paper in the following:

- *Enterprise Collaboration Ontology:* A collection of predefined semantic concepts establishes data interoperability through transformation, mediation, and reasoning. As depicted by Figure 4 the basic enterprise collaboration entities and their relations are well defined in a baseline ontology.

- *Semantic Goals:* A client specifies the objective to be achieved in terms of a goal [36], and the system resolves this by automated detection, composition, and execution of Web services. This concept allows dynamic discovery based on functional as well as non-functional properties, and advanced composability of services and service instances respectively.

- *Grounding Descriptions:* Since WSMX functionalities operate on semantic descriptions of messages, non-semantic messages require transformations to semantic representations and vice versa (i.e., lowering and lifting scripts).

- *Semantically-enriched WSDL Interface:* Data types used by Web service interfaces (WSDL) need to be linked to corresponding grounding scripts that mediate data between standard SOAP messages and semantic goals (RDF).

We utilize Semantic Web technologies to cope with inherent dynamics of open enterprise systems and to keep the environment manageable. In particular, we use the WSMX [19] platform to enable

- *Cross-Organizational Abstraction.* Since members from various domains and organizations need to interact, we use Semantic Web Services as an abstraction from organizational structures in order to distribute communication facilities. Typically members of virtual communities use their organizations' resources and infrastructure; Web services resolve the need (semantic goal) of interaction to actual SOAP requests and additionally mediate between differing ontological concepts.

- *Context-aware Interaction Channel Selection.* Selecting appropriate communication, coordination, and collaboration service does not only depend on functional needs, but also on contextual constraints. For instance, the delivery of a message (described by a semantic goal) can be achieved through e-mail services, instant messaging, or postings in Internet forums. The appropriate channel can be selected based on user data (location, privacy rules) and messages (priority, size).

2.3 Monitoring Layer

Interactions are observed and collected to determine social relations. We designed the system to manage relations by evaluating occurring interactions and therefore, unburden network participants – at least partly – from managing their relations manually. Logging invocations of collaboration services is the basis for advanced interaction analysis, and allows to infer social relations that are described by objectively measured metrics, such as average response times, availability, or reciprocity.

Formally, a virtual community is a special kind of social network, where the single actors participate to perform activities. A community is modeled as a directed graph, where vertices V represent the actors that are connected through edges E. A directed edge from actor u to v is denoted as e_{uv}. Activities A are a fundamental part of our model; thus, we describe the graph model of a community as $G = (V, E, A)$. The concept of an activity $a \in A$ is used to include a set of participants. Thus, in short, activities describe the collaboration boundaries and goals. Network members interact in scope of particular activities (i.e., to reach certain goals). Interactions are collected to determine (i) the center of interest of single network members by evaluating the frequency of used keywords [31, 33], and (ii) the strength of a social relation by determining the similarity of the center of interests [35]. Since these techniques have been extensively discussed in previous work, we do not present a detailed description in this paper.

2.4 Discovering Relevant Social Networks

In our framework, an actor has several *passive* links, modeled as FOAF relations, that express business/personal contacts (typically emerged from previous collaborations), but not describing that interactions are performed along these links. An actor can *activate* these links by initiating a new collaboration, e.g., setting up a joint activity. However, due

to resource constraints, members can only participate in a limited amount of concurrent activities, and thus, the number of simultaneously active links is limited. Hence, collaboration partners are discovered and selected carefully, considering required effort and received benefit.

Direct relations are established to create a typical social network. Since single members usually build up strong relations to only a small amount of partners, reliable information flows through collaborative networks, such as exchanging expertise and interest profiles, are limited. Thus, the discovery layer allows actors to exchange business contacts by sharing and propagating (parts of) profiles over intermediate nodes. Each actor's connectivity to other community members is determined by issuing keyword-based queries [32] denoted by the query context Q. The query context is described by a pool of keywords (e.g., describing certain expertise areas) picked from global taxonomies. Using logged interaction data (and additional manual ratings) the link weight from one actor to another is calculated using a *social trust* metric that is discussed in detail in the next section.

3. SOCIAL NETWORK MANAGEMENT

This section discusses a framework enabling distributed profile management in large-scale Web-based open enterprise systems. Profiles are shared among members and evaluated to discover potential collaboration opportunities based on interest similarities, coverage of expertise needs, project participation, and organizational memberships.

3.1 Architectural Overview and Design

Since information sharing with mostly unknown individuals in large-scale environments is a delicate matter, our framework applies common security standards to encrypt sensitive information and therefore, enables selective sharing of information. We adopt one of the most popular encryption concepts, in particular *public key infrastructure* (PKI) [1] for that purpose.

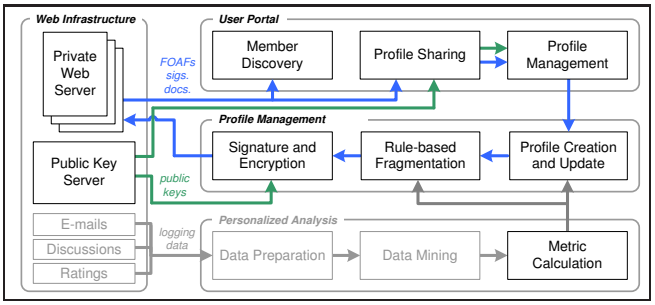


Figure 5: Architecture supporting discovery in self-managed social networks of open enterprise systems.

The fundamental architecture of our framework is depicted in Figure 5. Basically, the left side consists of globally available components, such as various Web servers owned by individuals and organizations, public key servers, and collaboration tools hosted in a semantic Web services environment, including e-mail infrastructures, discussions forums, and rating platforms. The right side comprises distributed components that are replicated for each user (and groups of users forming closed communities respectively) to manage their profiles from their personal point of view. The architecture consists of the following three layers: (i) *Personalized Analysis* enables data aggregation from collaboration tools and data mining to determine collaboration relations. Basically, the strength of social relations is inferred by calculated various interaction and behavior metrics from mining e-mail data or Internet forum entries [31, 33]. (ii) *Profile Management* includes features to semi-automatically create and update FOAF profiles with calculated metrics. Profiles are encrypted and valid signatures created so that only close collaboration partners can decrypt and use them for discovering actors. (iii) the *User Portal* hosts tools to discover potential partners and sharing and managing personal profiles.

3.2 Emergence of Social Relations and Trust

We believe that trust and reputation mechanisms are key to the success of open dynamic service-oriented environments. However, trust is emerging based on evidence, i.e., interaction behavior. Interactions, for example, may be categorized in terms of success (e.g., failed or finished) and importance. Therefore, a key aspect of our approach is the monitoring and analysis of interactions to automatically determine trust. We argue that in large-scale SOA-based systems, only automatic trust determination is feasible. In particular, manually assigned ratings are time-intensive and suffer from several drawbacks, such as unfairness, discrimination or low incentives for humans to provide trust ratings.

Trust Definition. In contrast to a common security perspective, social trust refers to the interpretation of previous collaboration behavior [33] and the similarity of dynamically adapting interests [16, 35]. Especially in collaborative environments, where users are exposed to higher risks than in common social network scenarios [12], and where business is at stake, considering social trust is essential to effectively guide interactions. Much research effort has been spent on defining and formalizing trust models (for instance, [18, 40]).

Here, we define trust as follows: *Trust reflects the expectation one actor has about another's future behavior to perform given activities dependably, securely, and reliably based on experiences collected from previous interactions.*

Interaction Metrics. In order to support the emergence of social relations, we utilize the following two metrics:

Interest Similarity isim. This metric determines the overlap of actor interests, which is an important measure to find motivated partners in the same interest area. We manage keywords used by actors u and v as interest profile vectors $\mathbf{p}_u$ and $\mathbf{p}_v$ respectively (see [35] for details), and determine the similarity of profiles through the cosine between their profile vectors (Eq. 1). The result is a value between 0 (no overlap) and 1 (full overlap).

$$isim(u,v) = cos(\mathbf{p}_u, \mathbf{p}_v) = \frac{\mathbf{p}_u \cdot \mathbf{p}_v}{|\mathbf{p}_u||\mathbf{p}_v|} \quad (1)$$

Reciprocity recpr. A typical social behavior metric is reciprocity [14] that here reflects the ratio between obtained and provided support in a community. Let $REQ(u,v)$ be the set of u's sent support requests to v, and $RES(u,v)$ the set of u's provided responses to v's requests. Then we define reciprocity in $[-1,1]$ as in Eq. 2; hence, 0 reflects a balanced relation of mutual give and take.

$$recpr(u,v) = \frac{|RES(u,v)| - |REQ(u,v)|}{|RES(u,v)| + |REQ(u,v)|} \quad (2)$$

The actual strength (weight w respectively) of a social trust relation is determined by normalizing, combining and weighting these metrics whenever a discovery process is started, i.e., a query issued. While *isim* is a globally valid metric, *recpr* is bound to distinct contexts Q (e.g., expertise areas). In particular, interactions bound to all activities whose description match at least one of the query keywords issued for discovering neighbor nodes are considered when calculating *recpr*. Currently we employ flat keyword-based matching only, however for more advanced ontology matching techniques see [10, 13]. Eq. 3 allows for the balancing between two cases: (i) *newcomer support* versus (ii) weighting of links of *well established* actors (based on evidence). The factor α can be adjusted based on the requirements for each case. For example, by setting $\alpha = 1$, newcomer support becomes more dominant since $isim_{uv}$ accounts for interest (profile) similarities. Whereas, the other case with $\alpha = 0$ puts stronger emphasis on already established links by accounting for the preference towards existing relations.

$$w^Q(u,v) = \alpha \cdot isim(u,v) + (1-\alpha) \cdot recpr^Q(u,v) \qquad (3)$$

3.3 TBAC - Trust based Access Control

Trust Bases Access Control (TBAC) supports the discovery of collaboration partners and subsequently the formation of groups and networks in open enterprise systems using distributed profile information. The main idea is to allow actors to access the profiles of other network members based on the strength of social relations, e.g., social trust. In other words, only trustworthy partners are allowed to access, in particular read, someone's personal profile information. Key principles of the proposed approach are:

- *Self-managed Distributed Profiles.* Actors manage their personal profiles in a distributed manner, i.e., profiles are fully under control of the respective actors.
- *Public and Private Scopes.* Some profile information may be available public, for instance, expertise area and basic contact details in order to discover new collaboration partners. However, access to sensitive information, e.g., private contact details and friend relations, is restricted.
- *Social Trust-based Access Control.* Access to private fragments of profiles is granted based on strengths of social relations. For instance, close collaboration partners can read larger parts of an actor's profile. Social trust relies on interactions and an update of personal relations can be triggered by actors using logged information from the SOA infrastructure. Note, only logged interactions with personal involvement are used.
- *Public Key Infrastructure (PKI).* PKI is the means to enable public and private profile scopes and to address privacy concerns in open distributed environments.

Transitive Access. As in the real world, information is not only shared between direct neighbors, but can traverse several intermediate nodes. Using this approach allows sharing of profiles along trusted paths even if actors are not directly connected in the social network. This spreading of information relies on the principle of recommendation and propagation of trust respectively [18]. Since all involved parties are connected with a strong trust path, privacy is still maintained. Transitive access is an important concept to overcome inherent limitations of trust based discovery only.

4. IMPLEMENTATION DETAILS

This section deals with the specification and implementation of the proposed social overlay model to realize dynamic discovery in semantically-enriched collaborative open enterprise systems.

4.1 Adaptive Distributed Profile Management

The mainly applied techniques are the Friend-Of-A-Friend (FOAF)[4] ontology, Public Key Infrastructure, in particular GnuPG[5], and Web-Of-Trust (WoT)[6] schemas.

4.1.1 Friend-Of-A-Friend Profile Management

Various concepts and protocols have been proposed to manage open social and collaborative networks in a distributed manner. The Friend-Of-A-Friend (FOAF) concept is one of the most popular ones on the Web. It allows to model user properties, interests and relations with a well-known ontology. We apply FOAF to facilitate the discovery process used to find potential collaboration partners.

```
<?xml version="1.0"?>                                                    1
<rdf:RDF xmlns:foaf="http://xmlns.com/foaf/0.1/"                         2
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"           3
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"                4
       xmlns:foaf="http://xmlns.com/foaf/0.1/"                           5
       xmlns:dc="http://purl.org/dc/elements/1.1/"                       6
       xmlns:wot="http://xmlns.com/wot/0.1/"                             7
<foaf:Person rdf:ID="me">                                                8
 <foaf:name>Florian Skopik</foaf:name>                                   9
 <foaf:nick>florian</foaf:nick>                                          10
 <foaf:mbox_sha1sum>a4b378...</foaf:mbox_sha1sum>                        11
 <wot:haskey rdf:nodeID="KeyFS" />                                       12
 <foaf:interest rdf:resource="http://..." />                             13
 <foaf:currentProject>                                                   14
  <foaf:Project>                                                         15
   <dc:title>Implementation Module X</dc:title>                          16
   <dc:description>WS, programming, java</dc:description>               17
   <dc:identifier rdf:resource="http://.../activity#4539"/>             18
  </foaf:Project>                                                        19
 </foaf:currentProject>                                                  20
 <foaf:knows>                                                            21
  <foaf:Person>                                                          22
   <foaf:mbox_sha1sum>1a4578...</foaf:mbox_sha1sum>                      23
   <foaf:name>Daniel Schall</foaf:name>                                 24
  </foaf:Person>                                                         25
 </foaf:knows>                                                           26
</foaf:Person>                                                           27
</rdf:RDF>                                                               28
```

Listing 1: Example of public FOAF file.

Listing 1 shows a simplified example of a public FOAF profile, containing basic personal properties (`name`, `nick`, `interest`) and social relations (`knows`). The Web of Trust (WoT) RDF ontology is used to integrate concept of a public key infrastructure into FOAF profiles, as demonstrated in Listing 2. The property `haskey` links a public key (`pubkey-Address`), `hex_id`, and `fingerprint` to a `person`. Furthermore, a person's private key is used to sign the own FOAF profile and therefore, to guarantee for integrity and authenticity. Notice, the only guarantee regarding authenticity is that the FOAF signer is owner of the registered mail account that has been used to create the key pair.

[4] `http://xmlns.com/foaf/spec/`
[5] `http://www.gnupg.org`
[6] `http://xmlns.com/wot/0.1/`

```
<!-- restricted part of FOAF profile -->
<rdfs:seeAlso>
 <foaf:Document rdf:about="http://.../foaf-private.rdf.asc">
  <wot:encryptedTo>
   <wot:PubKey wot:hex_id="34c5a421b" />
  </wot:encryptedTo>
 </foaf:Document>
</rdfs:seeAlso>

<!-- digital signature for this file -->
<rdf:Description rdf:about="">
 <wot:assurance rdf:resource="foaf.rdf.asc" />
</rdf:Description>

<!-- public key of the owner/signer of this file -->
<wot:PubKey rdf:nodeID="KeyFS">
 <wot:hex_id>3756EA0B</wot:hex_id>
 <wot:length>1024</wot:length>
 <wot:fingerprint>03f4...</wot:fingerprint>
 <wot:pubkeyAddress rdf:resource="http://.../key.asc"/>
 <wot:identity>
  <wot:User>
   <foaf:name>Florian Skopik</foaf:name>
   <foaf:mbox_sha1sum>a4b378...</foaf:mbox_sha1sum>
  </wot:User>
 </wot:identity>
</wot:PubKey>
```

Listing 2: Signing FOAFs (`wot:assurance`) and linking encrypted content (`rdfs:seeAlso`).

Access to parts of a FOAF document may be restricted to certain users (whose public keys are used to encrypt those parts). We utilize this concept for (i) private information, such as private phone numbers or chat accounts that can only be decrypted and used by close neighbors (connected via `knows`), and (ii) personal ratings that are given either *explicitly* (manually) or *implicitly* (through data mining of e-mail logs, instant messaging (IM) logs, or Internet forums). We understand privacy as a major concern when applying mining techniques; hence, mining of metrics is performed from each actor's perspective (or at least limited to certain groups of experts). This means that data is not stored centrally but managed on the client side and private servers.

```
<rdf:RDF xmlns:foaf="http://xmlns.com/foaf/0.1/"
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <foaf:Person>
  <!-- mbox_sha1sum links to public FOAF profile -->
  <foaf:mbox_sha1sum>a4b378...</foaf:mbox_sha1sum>

  <!-- private contact details -->
  <foaf:mbox rdf:resource="mailto:skopik@....tuwien.ac.at"/>
  <foaf:phone>+43 xxxx xxxx</foaf:phone>

  <!-- private chat account -->
  <foaf:account>
   <foaf:OnlineAccount>
    <rdf:type rdf:resource="http://.../OnlineChatAccount" />
    <foaf:accountServiceHomepage rdf:resource="http://..../" />
    <foaf:accountName>florian_skopik</foaf:accountName>
   </foaf:OnlineAccount>
  </foaf:account>

  <!-- attach personalized ratings to known persons -->
  <foaf:knows>
   <foaf:Person>
    <foaf:mbox_sha1sum>1a4578...</foaf:mbox_sha1sum>
    <foaf:tipjar rdf:resource="http://..." rdfs:label="ratings"/>
   </foaf:Person>
  </foaf:knows>
 </foaf:Person>
</rdf:RDF>
```

Listing 3: Private fragment of a FOAF profile.

Listing 3 depicts an example of encrypted private FOAF fragments. While users decide manually which parts of their profiles are shared globally and which are restricted to neighbors only, relation metrics, e.g., derived from personal ratings, are managed automatically by the system. For that purpose, single ratings are stored in a dedicated document (`tipjar`) for each user. This document is processed by various evaluation tools and plugins that are fully under control of the users. Currently, we have three tools for (i) collecting manual ratings, (ii) analyzing Internet forums, and (iii) analyzing e-mail communication in order to assess collaboration performance of known partners and the strength of social ties based on past interactions.

4.1.2 Profile Sharing

The presented concepts enable the discovery of directly connected partners based on common properties, interests, ratings, and contextual constraints (such as projects), but still preserve their privacy. This means that profile owners encrypt sensitive parts of their profiles for their known neighbors, i.e., using their public keys. Since we do not only manage binary knows relations but also calculate the strengths of relations (e.g., social trust), the amount of shared information can be bound to certain strength levels. For instance, whenever one updates his profile, a rule-based system decides based on predefined link thresholds, who is allowed to read private FOAF fragments and encrypt files accordingly.

```
<!-- link encrypted document -->
<foaf:Document rdf:about="http://.../foaf47.rdf">
 <dc:title>Restricted Information</dc:title>
 <wot:assurance>
  <wot:Endorsement rdf:about="http://.../foaf47.rdf.asc">
   <dc:title>signature of friend47 private profile</dc:title>
   <wot:endorser rdf:nodeID="KeyFS"/>
  </wot:Endorsement>
 </wot:assurance>
</foaf:Document>

<!-- encryption information -->
<wot:EncryptedDocument rdf:about="http://.../foaf47.rdf.asc">
 <dc:title>friend47 private profile</dc:title>
 <wot:encryptedTo rdf:nodeID="KeyPartnerX"/>
 <wot:encrypter rdf:nodeID="KeyFS"/>
</EncryptedDocument>
```

Listing 4: Linking encrypted documents in FOAF.

However, single members usually build up strong relations to only a small amount of partners. That hinders the discovery process. In order to overcome that hurdle, we allow propagation of information over several intermediate hubs along strong social paths. Enabling such flows of information enables actors to discover new potential collaboration partners. Technically, we allow actors to link *private* profile information of well connected partners as personally encrypted documents to their own profile. Restricted access is the basis for personalized and reliable sharing of information. We use once more the WoT ontology to link external documents to one's FOAF profile (see excerpt in Listing 4). A detailed implementation perspective regarding processing of XML data is out of scope of this paper, but has been investigated in detail in [34]. A semantically-enriched Web Service based environment allows to notify partners about updated profiles and send them links to encrypted documents. The receivers are able to validate these documents, i.e, verify the authenticity and consistency using the signer's public key and to decrypt information using their own private keys.

4.2 Semantic Service Infrastructure

WSMX (Web Service Modeling eXecution environment) [19] allows to describe and register Web services and thus, supports discovering, selecting, and invoking Web services at run-time in a semantic manner. The actual services are hosted elsewhere, but WSMX builds a semantic abstraction layer for these services by managing additionally required artifacts (as described in Section 2.2). The WSMX platform provides a WS entrypoint to submit semantic goals that need to be fulfilled and the platform itself discovers the best suitable service based on (i) functional properties (FPs), i.e., supported concepts, such as messaging; and (ii) non-functional properties (NFPs), here, contextual constraints including organizational boundaries, people's location and working context.

4.2.1 Registering Semantic Web Services

The first step of registering a common Web service with a WSDL interface in WSMX is to annotate appropriate lowering- and lifting scripts. These XSLT scripts enable the transformation between SOAP messages and ontological representations. Listing 5 shows a small excerpt of a semantically-enriched WSDL file. Here, the complex data type `sendMessageKey` (and its corresponding response) have `loweringSchemaMapping` and `liftingSchemaMapping` respectively attached. Listing 6 shows a lowering script. Here, values of required semantic concepts to build an instance of type `sendMessageKey` are extracted from the enterprise collaboration ontology.

```
<xs:element name="sendMessageKey"
 sawsdl:loweringSchemaMapping="SendEmailMessage-lowering.xslt">
 <xs:complexType>
  <xs:sequence>
   <xs:element minOccurs="0" name="to" type="xs:string"/>
   <xs:element minOccurs="0" name="subject" type="xs:string"/>
   <xs:element minOccurs="0" name="body" type="xs:string"/>
   <xs:element minOccurs="0" name="key" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="sendMessageKeyResponse"
 sawsdl:liftingSchemaMapping="SendEmailMessage-lifting.xslt">
 <xs:complexType>
  <!-- details omitted -->
 </xs:complexType>
</xs:element>
```

Listing 5: Schema mapping annotations in WSDL.

```
<xsl:template match="rdf:Description[rdf:type/@rdf:resource=
'http://www.coin-ip.eu/ontologies/ec#EmailServiceMessage']">
 <email:sendMessageKey>
  <xsl:for-each select="ecg:hasEmailAddress">
   <to><xsl:value-of select="."/></to>
  </xsl:for-each>
  <xsl:for-each select="ecg:hasSubject">
   <subject><xsl:value-of select="."/></subject>
  </xsl:for-each>
  <xsl:for-each select="ecg:hasContent">
   <body><xsl:value-of select="."/></body>
  </xsl:for-each>
  <xsl:for-each select="ecg:hasAuthenticationKey">
   <key><xsl:value-of select="."/></key>
  </xsl:for-each>
 </email:sendMessageKey>
</xsl:template>
</xsl:stylesheet>
```

Listing 6: Lowering script example.

4.2.2 Semantic Goal Description

Listing 7 shows exemplarily a goal defined in WSML[7] for sending a notification via e-mail. For that purpose, NFPs are defined (here: type of discovery), as well as pre- and postconditions for invoking a capable Web service (e.g., defined recipient and message). The block `instance emailRequest` contains the actual parameters that are lowered to a SOAP message and sent to an Email Web service.

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
namespace { _"http://www.coin-ip.eu/goals/ec#",
disc _"http://wiki.wsmx.org/index.php?title=DiscoveryOntology#",
ec _"http://www.coin-ip.eu/ontologies/ec#",
ecp _"http://www.coin-ip.eu/ontologies/ecp#"}

goal MessageGoal
 importsOntology {
  ec#EnterpriseCollaborationOntology,
  ecp#EnterpriseCollaborationProcess
 }

capability MessageGoalCap
 nonFunctionalProperties
  disc#discoveryStrategy hasValue disc#NoPreFilter
  disc#discoveryStrategy hasValue disc#HeavyweightDiscovery
 endNonFunctionalProperties

sharedVariables {?x, ?z, ?y}

precondition MessageGoalPre
 definedBy
  ?x memberOf ec#EmailMessage and
  ?z memberOf ec#Individual and
  ?y memberOf ec#Individual.

postcondition MessageGoalPost
 definedBy
  ecp#messageSent(?z, ?x, ?y).

ontology EmailRequest
 importsOntology {
  ec#EnterpriseCollaborationOntology
 }

instance emailRequest memberOf ec#EmailMessage
 hasAuthenticationKey hasValue "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx"
 hasEmailAddress hasValue "name@infosys.tuwien.ac.at"
 hasSubject hasValue "Notification about project opportunity"
 hasContent hasValue "Dear sir, according to your profile ..."
```

Listing 7: Semantic goal for e-mail message service.

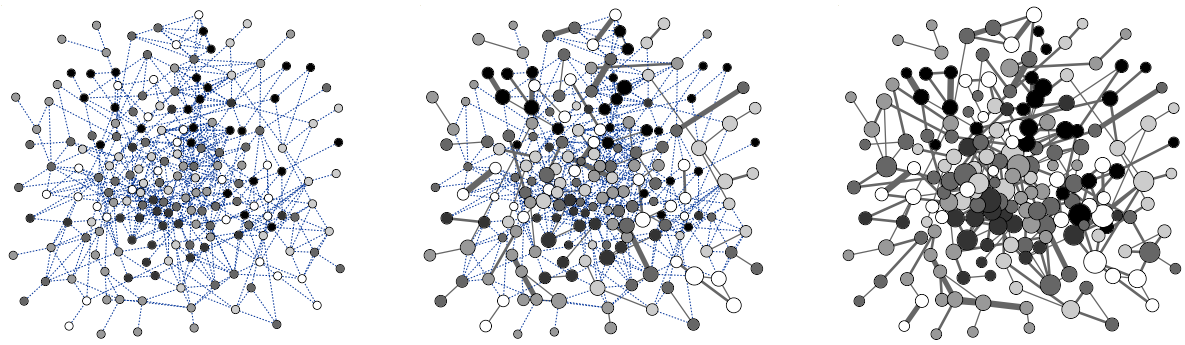
5. EVALUATION AND DISCUSSION

This section deals with evaluation results regarding the whole system as well as discussions of essential findings. In particular, we demonstrate the performance of semantically-enriched service hosting with WSMX, discuss network formation processes using simulation, study member discovery processes through propagating distributed FOAF profiles, and discuss various design decisions with respect to PKI for FOAF.

5.1 WSMX Performance Aspects

The used WSMX setup consists of 38 different Web services, primarily communication services and document management services, 52 ontology parts (the main ontology is the enterprise collaboration ontology depicted in Figure 4, but further ontologies of single services refine some concepts), and 13 semantic goals (e.g., sending a message with a given

[7]Web service modeling language



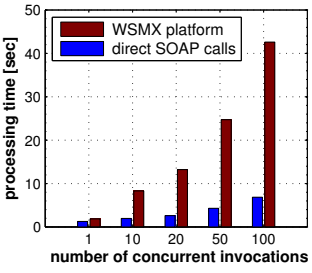
(a) bootstrapping phase: only predicted links ($\alpha = 1$)

(b) formation phase: mix of predicted and emerged links ($\alpha = 0.5$).

(c) saturation phase: only emerged links ($\alpha = 0$)

Figure 6: Network formation process visualization.

content to a particular person). For the following experiments, WSMX and services (implemented using Axis2[8]) are hosted on a server with Intel Xeon 3.2GHz (quad), 10GB RAM, running Tomcat 6 with Axis2 1.4.1 on Ubuntu Linux. Furthermore we perform concurrent calls from a client simulation that runs on a Pentium 4 with 2GB on Windows XP, and is connected with the server through a local 100MBit Ethernet. Figure 7 compares the performance of WSMX with standard SOAP calls that invoke Web services directly for different numbers of concurrent calls.. Note, the additional overhead caused by WSMX is the difference between the two results, since after processing the semantic layer also WSMX invokes a particular WS via SOAP only. In our test environment, invoking a service via WSMX compared to invoking the same service directly takes approximately 5 times longer. The additional processing time is used for lowering a request (such as the goal in Listing 4) to a SOAP message, and, after invoking the service, lifting the response back to the semantic level. Although WSMX adds much additional overhead to service invocation, several advantages can be taken, including, dynamic discovery and selection of best suitable service instances (depending on NFPs), and establishing real cross-enterprise interoperability through data mediation on ontological level. Note, services can be distributed over several WSMX instances to distribute load and increase performance.



WSMX step	time [ms]
lowering	250-400
service invoc.	> 130
lifting	150-250
single invoc.	**time [ms]**
SOAP (avg)	80-120
WSMX (avg)	450-750

Figure 7: WSMX performance comparison.

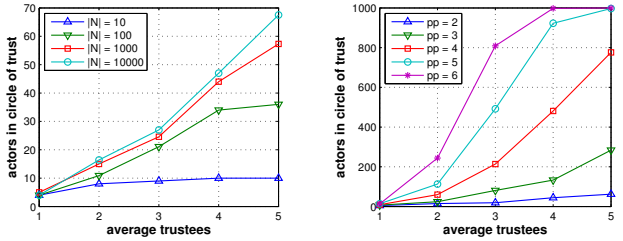
5.2 Network Formation Simulation in SOA

We use a Web service testbed to simulate the interaction behavior in SOA-based communities. The purpose of the Genesis2 framework [22] (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 test environment consists of 200 autonomous services that simulate behavior in common flexible collaboration scenarios. Each service (called actor) has an interest/expertise profile assigned, consisting of 5 to 8 distinct keywords. Profiles may partly overlap. In order to bootstrap collaborations links between actors are predicted based on profile similarities. Typically, interest similarities are a reasonable grounding for future collaboration success and emerging personal relations [39]. During the actual collaboration single actors interact by delegating tasks and requesting support from other members of the community; thus, in our simulation we let random members interact in fixed time intervals. Each interaction is tagged with a maximum of 3 keywords and sent to actors with matching interest profiles. 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 enable us to study the formation process of typical medium scale Web-based communities. In particular we investigate the three phases of (i) bootstrapping, i.e., initiating the formation of a network; (ii) formation phase, i.e., setting up strong links between matching collaboration partners; (iii) saturation phase, i.e., cross-linking emerging small-scale communities with weak links. The aim of this experiment is to determine the effort in terms of monitoring and processing interactions until similar network structures (in the respective evolutionary phases) for different taxonomy complexities emerge. For instance, using less complex taxonomies consisting of only 10 keywords also requires less monitored interactions, since profiles and interaction contexts converge much faster than for more complex taxonomies.

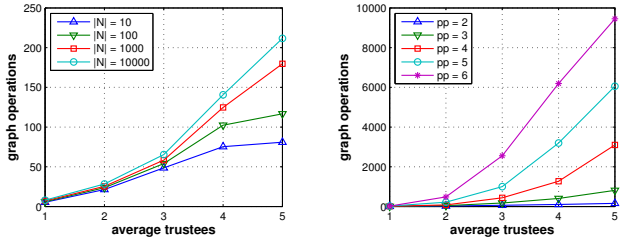
Experiment Results. We study the network formation process of 200 unconnected actors for different environment setting. Depending on the complexity of the global taxonomy that determines interaction contexts, varying amounts of interactions are required in order to guarantee a feasible

[8] `http://ws.apache.org/axis2/`



(a) varying network size $|N|$. (b) varying prop. path pp.

Figure 8: Size of the circle of trust.



(a) varying network size $|N|$. (b) varying prop. path pp.

Figure 9: Required graph operations.

inference of social relations based on interest similarities. We let actors pick tags from a global taxonomy consisting of 10/20/50 keywords according to their interest profiles in order to annotate their interactions, e.g., express the expertise areas of support requests. In order to bootstrap a network formation process (see Figure 6(a)) links are predicted only (see dashed lines) based on actor profile overlaps [35]. Utilizing measured interaction metrics (here reciprocity cf. Eq. 2), social links are established based on evidence about reliable and dependable collaboration behavior. Note, the color of the nodes represent their (static) expertise areas, while their sizes reflect their degree of connectivity in the network. Figure 6(b) shows a network where most members have found at least one trustworthy (e.g., in terms of reciprocity) collaboration partner. Such social links are reflected by solid lines whereas their strengths reflect the level of cooperation. Still, most relations are predicted only (dashed lines). Finally, after a sufficient amount of interactions has been collected to reliably infer relations, a network consisting only of evidence-based relations is maintained in the saturation phase (Figure 6(c)).

We repeat this experiment to find out typically emerging network structures for varying taxonomy complexities (number of tags #tags) and different amounts of interactions (#ia). Table 1 reveals the details. The metrics are (i) number of connected components (nc), (ii) average number of network neighbors (nn), and (iii) network density (nd). Although an optimal connection is hard to determine, these graph metrics deem to be appropriate indicators [30] to describe and compare network structures. Note, the values in brackets int eh bootstrapping phase denote the given metrics if predicted links are treated as evidence-based links.

Table 1: Characteristic network metrics in different evolutionary phases of a formation process.

phase	#tags/#ia	network metrics
bootstr.	10/0	$nc = 200$, $nn = 0(7.62)$, $nd = 0$
	20/0	$nc = 200$, $nn = 0(5.56)$, $nd = 0$
	50/0	$nc = 200$, $nn = 0(1.13)$, $nd = 0$
formation	10/1000	$nc = 99$, $nn = 1.12$, $nd = 0.006$
	20/3000	$nc = 109$, $nn = 0.84$, $nd = 0.005$
	50/5000	$nc = 101$, $nn = 0.98$, $nd = 0.005$
saturation	10/5000	$nc = 4$, $nn = 3.15$, $nd = 0.017$
	20/15000	$nc = 7$, $nn = 2.65$, $nd = 0.014$
	50/25000	$nc = 5$, $nn = 2.89$, $nd = 0.016$

5.3 Member Discovery Simulations

We create synthetic networks with fixed amounts of nodes and power-law distributed edges [29] to evaluate the effects of propagating profile information. This means, encrypted parts of a FOAF profile are shared over multiple hops even between unconnected members, if there is a strong trust path between them. This concept of propagation [18] enables users to extend their *circles of trust* (i.e., all members that can be reached over a strong trust path without exceeding an upper limit of hops) and to discover previously unknown members therein. The complexity of a graph is described by the average outdegree of a node in the long tail of the degree distribution; in other words, the average number of trusted neighbors (trustees) for the bottom 90% of members. We pick random nodes from this set and run experiments for each one until we get stable average results.

The first set of experiments investigates the average size of the *circle of trust*, depending on the number of trustees for different network sizes N and propagation path lengths pp. For that purpose profiles of all neighbors of specified nodes in the network are retrieved recursively until the whole circle is discovered. Figure 8 show that for highly cross-linked graphs (i.e., $avgtrustees > 2$), only short pps (max. 3 or 4 hops) are feasible. Otherwise, virtually all members are in one's *circle of trust*. A second set of experiments highlights the computational complexity of determining the *circle of trust*. While the size of the network does not considerably influence the number of required graph operations from each actor's perspective (at least for small pp), increasing pp in highly cross-linked graphs leads to exponential costs (Figure 9). Graph operations include retrieving referenced nodes and edges, as well as neighbors, predecessors and successors in the network model. Each of these operations means that finally distributed FOAF profiles need to be queried and retrieved from the Web.

5.4 Processing Encrypted FOAF Profiles

We shortly discuss the complexity and required steps to enable the discovery of collaboration partners based on FOAF profile sharing using the security concepts discussed in this paper. For that purpose, we distinguish between three different operations: (i) publishing profiles, (ii) discovering neighbors, i.e., retrieve their (encrypted) profiles, (iii) transitive discovery, i.e., propagation of profile information over one hop. Table 2 summarizes complexities in terms of *number of retrieved documents* (i.e., public/private FOAF fragments, signatures, public/private key files) and *number of required steps* (i.e., file retrieval, encryption, decryption, file update, file upload). Note, we do not measure absolute performance of the proposed profile management approach, because this heavily depends on the hosting environment and IT infrastructure. Symbol n denotes the number of direct neighbors; p the number of distinct private FOAF fragments.

FOAF Profile Publishing. Updating an actor's own

Table 2: Comparison of profile management ops.

operation	#retrieved docs	#steps
FOAF profile publishing	$3+n$	$3+3p+n$
neighbor discovery	$(2+p)\cdot n$	$(3+p)\cdot n$
transitive discovery	$2+2p+n+pn$	$3+3p+n+pn$

profile consists of profile retrieval and update of already existing public/private profile fragments, signing the public fragment with own private key, retrieving the neighbors' public keys, encrypting private fragments individually for strongly connected (e.g., trusted) neighbors, publish public and private fragments on the Web.

Neighbor Discovery. This operation discovers directly connected actors by evaluating their profiles, e.g., interests, project participation, organizational memberships. Evaluating neighbor profiles includes for each single neighbor to retrieve the public profile and public key to validate the signature, retrieval of linked private fragments, decryption of data with own private key.

Transitive Discovery. Transitive profile sharing enables the discovery of unconnected community members. For that purpose intermediate nodes mediate information by retrieving (encrypted) profiles from neighbors, and re-encrypt them for their own (trusted) neighbors. In particular the following steps are performed: retrieve published public/private FOAF fragments of one neighbor, get public key to verify signature, decrypt private fragment with own private key, get public key of other neighbor(s), re-encrypt private fragment, attach this fragment to own FOAF profile, re-sign and re-encrypt own FOAF fragments; optionally, notify interested neighbors about third-party profiles.

6. BACKGROUND AND RELATED WORK

Cross-Organizational Collaborations. The concept of virtual communities is increasingly used to enable the collaboration between geographically distributed members belonging to various organizational units. Studies on distributed teams focus on human performance and interactions [28] as well as *Enterprise 2.0* environments [7]. Service-oriented architectures (SOA) have emerged as the defacto standard to design and implement open enterprise systems. They allow for loose coupling between single components and enable sophisticated discovery mechanisms based on functional (e.g., supported features) and non-functional (e.g., QoS) properties. Web service technology [2] enables cross-organizational interactions in collaborative networks [9].

Monitoring and Self-Organizing Systems. The problem of composition and adaptation is strongly related to organization and control. Self-* principles [6] provide the ability to manage systems autonomously and to dynamically adapt to changes in accordance with objectives and strategies. Enhanced flexibility of complex systems is introduced by establishing a cycle that feeds back environmental conditions to allow the system to adapt its behavior. This MAPE cycle [21] is considered as one of the core mechanisms to achieve adaptability through self-* properties. While autonomic computing allows for autonomous elements and applies these principles to distributed systems, current research efforts left the human element outside the loop. In the context of multi agent systems (MAS), self-configuring social techniques were introduced in [8].

Social Trust in service-oriented systems has become a very important research area. SOA-based infrastructures are typically distributed comprising a large number of available services and huge amounts of interaction logs. Therefore, trust in SOA has to be managed in an automatic manner [25]. Depending on the environment, trust may rely on the outcome of previous interactions [27] and interest similarity [16, 26]. In our approach, metrics express social behavior influenced by the context in which collaborations take place [33]. For instance, *reciprocity* [14] is a concept describing that humans tend to establish a balance between provided support and obtained benefit from collaboration partners.

Social Platforms and Service Communities. Social networks have received tremendous attention recently from both research and academia. A large amount of information is exchanged online using social networking platforms. It becomes thus essential to adapt and influence the information exchange in an automated manner [34]. Selective dissemination of information (SDI) [3, 11] is used filter unnecessary data by considering user profiles.

Social networks become more and more interlinked with enterprises and collaborative platforms [7]. Semantically-enriched service platforms following the SOA paradigm such as WSMX [19] provide the means to discover and compose services in cross-organizational environments based on standardized languages (see WSMO [24]). These platforms not only enable interactions between technical services across boundaries, but also human interactions on top of these services. The convergence of social interactions in flexible service-oriented environments makes it essential to extend well-established data formats for describing the structure of social networks such as FOAF with access control techniques.

The mechanisms for signing RDF graphs have been presented in [15]. The combination of FOAF and SSL [37] enables secure access to FOAF profiles. The embedding of access control mechanisms in FOAF has been illustrated in [20, 23].

7. CONCLUSION AND FUTURE WORK

In this paper, we discussed the application of social network concepts in cross-enterprise collaboration scenarios. While creating dynamic profiles and flexibly discovering people and services is frequently used in typical recommender systems and on social platforms, the application in enterprise scenarios in form of overlay networks is a novelty. Especially, the combination with Semantic Web methodologies, such as Web services, taxonomy-based context management and SOA to achieve data and service interoperability is a new aspect. We proposed an approach to support human collaboration in different domains and organizations in a seamless manner; not only from s social perspective, but also from a technical one.

Our future research includes the application of social overlay networks in real cross-enterprise scenarios. This will be done within the EU FP7 project COIN, where we will collect valuable information regarding the efficiency of the discovery process based on social network structures. Furthermore, we will study network dynamics such as member fluctuation and frequency of re-discovering partners; as well as the feasibility of our approach from a technical point of view, e.g., limits in managing FOAF profiles depending on profile change rates.

Acknowledgments

This work was supported by the European Union FP7 projects COIN (216256) and SM4ALL (224332). The authors like to thank STI Innsbruck for support regarding WSMO, WSML and WSMX.

8. REFERENCES

[1] C. Adams and S. Lloyd. *Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations*. Macmillan Publishing, 1999.

[2] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services - Concepts, Architectures and Applications*. Springer, October 2003.

[3] M. Altinel and M. J. Franklin. Efficient filtering of xml documents for selective dissemination of information. In *VLDB*, pages 53–64, 2000.

[4] Amazon.com. Amazon mechanical turk, last access: 2010. available online: http://www.mturk.com.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[6] A. Berns and S. Ghosh. Dissecting self-* properties. In *SASO*, pages 10–19. IEEE, 2009.

[7] J. Breslin, A. Passant, and S. Decker. Social web applications in enterprise. *The Social Semantic Web*, 48:251–267, 2009.

[8] V. Bryl and P. Giorgini. Self-configuring socio-technical systems: Redesign at runtime. *ITSSA*, 2(1):31–40, 2006.

[9] L. M. Camarinha-Matos and H. Afsarmanesh. Collaborative networks. In *PROLAMAT*, pages 26–40, 2006.

[10] S. Castano, A. Ferrara, and S. Montanelli. Matching ontologies in open networked systems: Techniques and applications. pages 25–63, 2006.

[11] Y. Diao, S. Rizvi, and M. J. Franklin. Towards an internet-scale xml dissemination service. In *VLDB*, pages 612–623, 2004.

[12] C. Dwyer, S. R. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of facebook and myspace. In *AMCIS*, 2007.

[13] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, Berlin, 2007.

[14] A. Falk and U. Fischbacher. A theory of reciprocity. *Games and Economic Behavior*, 54(2):293–315, 2006.

[15] M. Giereth. On partial encryption of rdf-graphs. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *ISWC*, pages 308–322, 2005.

[16] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Trans. on the Web*, 3(4), 2009.

[17] N. Gold, C. Knight, A. Mohan, and M. Munro. Understanding service-oriented software. *IEEE Software*, 21(2):71–77, 2004.

[18] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, pages 403–412, 2004.

[19] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. Wsmx - a semantic service-oriented architecture. In *ICWS*, pages 321–328. IEEE, 2005.

[20] J. Hollenbach, J. Presbrey, and T. Berners-Lee. Using rdf metadata to enable access control on the social semantic web. In *ISWC*, 2005.

[21] IBM. An architectural blueprint for autonomic computing. *Whitepaper*, 2005.

[22] L. Juszczyk and S. Dustdar. Script-based generation of dynamic testbeds for soa. In *ICWS*, pages 195–202. IEEE, 2010.

[23] S. R. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H.-C. Choi. D-foaf: Distributed identity management with access rights delegation. In *ASWC*, pages 140–154, 2006.

[24] R. Lara, D. Roman, A. Polleres, and D. Fensel. A conceptual comparison of wsmo and owl-s. In *ECOWS*, pages 254–269. IEEE, 2004.

[25] Z. Malik and A. Bouguettaya. Reputation bootstrapping for trust establishment among web services. *Internet Computing*, 13(1):40–47, 2009.

[26] Y. Matsuo and H. Yamamoto. Community gravity: Measuring bidirectional effects by trust and rating on online social networks. In *WWW*, pages 751–760, 2009.

[27] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *HICSS*, page 188, 2002.

[28] N. Panteli and R. Davison. The role of subgroups in the communication patterns of global virtual teams. *IEEE Trans. Prof. Com.*, 48(2):191–200, 2005.

[29] A. Reka and Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, June 2002.

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

[31] D. Schall and S. Dustdar. Dynamic context-sensitive pagerank for expertise mining. In *Social Informatics*, pages 160–175. Springer, 2010.

[32] D. Schall and F. Skopik. Mining and composition of emergent collectives in mixed service-oriented systems. In *CEC*. IEEE, 2010.

[33] F. Skopik, D. Schall, and S. Dustdar. Modeling and mining of dynamic trust in complex service-oriented systems. *Inf. Syst.*, 35:735–757, 2010.

[34] F. Skopik, D. Schall, and S. Dustdar. Trust-based adaptation in complex service-oriented systems. In *ICECCS*. IEEE, 2010.

[35] F. Skopik, D. Schall, H. Psaier, and S. Dustdar. Social formation and interactions in evolving service-oriented communities. In *ECOWS*. IEEE, 2010.

[36] M. Stollberg and B. Norton. A refined goal model for semantic web services. In *ICIW*, pages 17–22, 2007.

[37] H. Story, B. Harbulot, I. Jacobi, and M. Jones. Foaf+ssl: Restful authentication for the social web, last access: 2010. http://esw.w3.org/Foaf%2Bssl.

[38] M. Zaremba and T. Vitvar. Wsmx: A solution for b2b mediation and discovery scenarios. In *ESWC*, pages 884–889, 2008.

[39] C.-N. Ziegler and J. Golbeck. Investigating interactions of trust and interest similarity. *Decision Support Systems*, 43(2):460–475, 2007.

[40] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Inf. Syst. Frontiers*, 7(4-5):337–358, 2005.