Artefact-Actor-Networks as tie between social networks and artefact networks

Wolfgang Reinhardt
University of Paderborn
Institute for Computer Science
33102 Paderborn, Germany
Email: wolle@upb.de

Matthias Moi
University of Paderborn
Institute for Computer Science
33102 Paderborn, Germany
Email: moisun@upb.de

Tobias Varlemann
University of Paderborn
Institute for Computer Science
33102 Paderborn, Germany
Email: tobiashv@upb.de

***Abstract*—Social networks reflect communication, cooperation and loose acquaintances in networked communities. Numerous metrics allow to expose connections, important persons or clusters within these communities. Furthermore, networks can be spanned to connect documents, blog entries or wiki articles. We call such a network an artefact network. In this paper we introduce the approach of Artefact-Actor-Networks that tries to connect social networks and artefact networks in order to make claims on the semantical connections between persons and manifold artefacts. We present practical use cases for Artefact-Actor-Networks and discuss generic and specific semantical requirements and added values through the existence of Artefact-Actor-Networks.**

I. Introduction

Computer mediated communication (CMC) has evolved to an important factor of industry, science and research within the last decades. As [1] puts it, we produce joint products and achieve higher productivity by electronic collaboration between distributed teams of humans, computer applications, and/or autonomous robots. Todays communication and cooperation is mainly web-based and relies on efficient technologies. So-called social software has found its way in businesses and connects persons within an organisation with experts outside. Collaboration is more and more dependent on the fast and reliable retrieval of these experts and relevant artefacts that help solving a problem or task.

Social networks and their analysis have emerged as a key technique in analysing groups of people working or living together. Mitchell defines a social network as '*a specific set of linkages among a defined set of persons with the additional property that the characteristics of these linkages as a whole may be used to interpret the social behaviour of the persons involved*' [2]. By extending Mitchell's interpretation of a social network, we get a more generic definition of social networks.

Social networks represent social structures by means of ties between nodes. These nodes correspond to actors, like persons or other individuals. Individuals as actors from different types can be commingled into heterogeneous networks. Edges in a social network can be seen as a special type of association or dependence between nodes respectively actors. Social networks spring up if people communicate, work or share data between each other. To analyse a social network there exist several techniques in the area of social network analysis (SNA). The SNA grew up essentially by techniques from modern sociology [3]. It uses several metrics to make statements about the structure of the network, the central node or the proximity of nodes. If the nodes in such a network are no longer individuals or groups, but artefacts such as pictures, blog entries, videos or wiki articles, we call these networks *artefact networks*. Here, the same metrics from SNA can be applied. Edges between these artefacts represent the type of connection or common contents of the artefacts. Artefact networks try to make statements about how artefacts are linked and used. If two artefacts are related, it seems that there exists a semantic relation between them. It applies to make them machine readable and evaluable.

In this paper we present an approach to combine social networks with artefact networks. The resulting Artefact-Actor-Networks allow making claims about the ties between artefacts from multiple sources and the actors involved in their creation, modification and linkage. By application of semantic web technologies and visualisations, Artefact-Actor-Networks represent a useful theoretical foundation of modern computer supported cooperative work and learning. They help to understand how communities are using artefacts and which role these artefacts play for object-centred sociality [4]. Artefact-Actor-Networks are saving the semantic context and supply the storage of metadata from people and artefacts. In other words, they deliver a great collection of data to connect social networks with person and artefact networks.

The remainder of this paper is structured as follows: first we introduce the theoretical background of this paper and present a brief overview on the basic principles of our work. We describe the different semantic relations of Artefact-Actor-Networks and define concepts of semantic similarity. Following this we introduce the approach of Artefact-Actor-Networks in more detail and provide use cases for its application followed by an inspection of possibilities of tracking the dynamics of our networks. Finally we conclude our paper with an outlook on upcoming applications of Artefact-Actor-Networks and prototypes under development.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

II. Theoretical Background

In this section we give a brief overview of the basic concepts and techniques that we used for the development of the model of Artefact-Actor-Networks. These concepts include computer-supported cooperative work (CSCW), social network analysis (SNA), semantic web, and semantic relations.

A. Computer Supported Communication and Collaboration

Computer mediated communication (CMC) and the more specific computer-supported cooperative work (CSCW) have evolved as important factors of industry, science and research within the last 25 years since its definition in 1984 by Greif and Cashman. CSCW is an interdisciplinary field of research where researchers use their specific viewpoint on the topic. Bowers and Benford [5] thus define CSCW in a very broad and universally way: *"In its most general form, CSCW examines the possibilities and effects of technological support for humans involved in collaborative group communication and work processes."* Ellis *et al.*[6] on the other hand emphasize that not only computers, but all technological assistance need to be taken into account when defining CSCW. So they define that CSCW *"looks at how groups work and seeks to discover how technology (especially computers) can help them work"*. CSCW deals with the support of social interaction on different layers. Teufel *et al.* distinguishes between communication, coordination and collaboration [7] while others include co-existence and consensus. Co-existence is the basic foundation for collaboration while communication allows to make arrangements, share ideas and pass over data. Coordination is needed when the single activities of users are dependent while consensus denotes the act of jointly making decisions and thus marks the highest level of collaboration.

Due to the huge amount of data and linkages between artefacts and their respective authors and editors, theoretical models and practical techniques for capturing the relations and dynamics of communication and collaboration need to be developed. Social network analysis (SNA) is an entrenched method of analysing such communication networks. With Artefact-Actor-Networks we propose a novel model of capturing relations and dependencies between artefacts and actors in a network.

B. Social Network Analysis

With SNA and it's metrics you can get various statements about a social network. In social networks we model a social network as a graph with nodes and edges between these nodes. A simple metric is the degree of a network, which results by the highest degree over all nodes. The degree of one node is calculated by adding all incoming and outgoing edges, if we talk about directed graphs as networks. Otherwise we count every edge exactly once. A very interesting and fundamental metric in SNA is the density of a network, which is limited by the maximum number of edges that can exist. It provides the integrity or the fragmentary of a network. The higher the density in a network, the stronger the awareness amongst the participants is [8]. Additionally, it is possible to calculate the local density on a subnetwork by explicitly declaring an actor to start from. This provides a measure, which describes the proportion between real connections and all possible connections to other actors. Other interesting metrics are the centrality measures defined by Freemann in [9]. With them you can show the popularity in a network. It is generally accepted that the more connections running over an actor, the more important his role is in the network. Now we give a brief overview on Freeman's measures in centrality.

- *Based on degree:* An actor is central if the direct links to other actors in the network are maximal. For example, this is a metric for the possible activity in communication.
- *Closeness measure* Here, the one actor is most central, which has many short direct or indirect connections to other actors. Since the calculation of the centrality measure of a node implies the sum of path instances, we can find out how independent an actor is.
- *Betweenness measure* With this measure we calculate the probability that a node is intersected between a shortest path of two other arbitrary nodes. A node is assumed to be central, if the paths running over this node are maximal, by selecting paths from pairs of nodes.

C. Semantic Web

The goal of the WorldWideWeb was to create a universal medium for the exchange of data [10]. With the ability to create links to every place of the WorldWideWeb, it opens a new area of information exchange. But the elements of the Web have no specified meaning which make it hard for computers and people to compare Information from different sites or to create meaningful results for a web search.

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.* [11]

Berners-Lee explained in his paper from 2001 a way to attach structure to the meaningful content of the Web. He added a new layer to the WorldWideWeb, which describes the information in a structured way, so that it is readable by computers and able to conduct automated reasoning. This layer adds logic to the web, which means to use rules to make inferences to the information. Agents, computer or human, can use this logical layer to collect information about things of interest and make new logical links to things that are also interesting.

This structured data consists of RDF triple of the form subject-predicate-object, which is in the form of simple sentences that express some semantics. All three parts of the triples base on a URI, which makes it easy to reference a specific document, a web page or a person and to add new predicates for a special purpose just by adding a new URI. These triples can be written in various formats but the most common is XML. XML can stand alone or can be embedded in HTML web pages as hidden fields. If anyone can create new predicates, then there will be more then one predicate for one

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

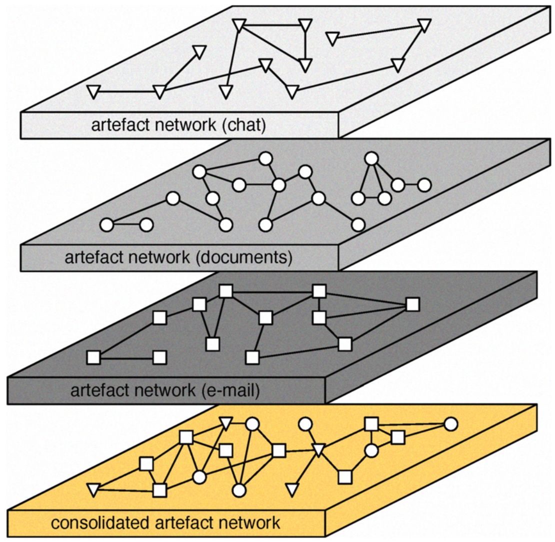

Fig. 1. Consolidated artefact network resulting from three layers

thing or for two similar things. Ontologies connect predicates with a meaning, in such a way that an ontology formally defines relations among the predicates. The most common ontology has a taxonomy and a set of rules. Taxonomies create classes that can be used as subjects and object and relations between classes, in this way the class *TwitterAccount* is also an *OnlineAccount*. A relation between the class *Person* and *TwitterAccount* could be *hasOnlineAccount*. Inference rules express if a *TwitterAccount* has created a *TwitterPost* and a *Person* has a *TwitterAccount* then the *Person* has created the *TwitterPost*.

III. Introducing Artefact-Actor-Networks

Artefact-Actor-Networks are an approach to connect social networks and artefact networks, with the goal to create more meaningful semantic connections between artefacts and actors. To connect artefacts and actors under and between each other, semantic relations are required. Every relation in the network connects objects by a semantic context like *isAutor* or *isRightHolder*. With the help of Artefact-Actor-Networks participation in the life cycle of artefacts as well as significant connections to involved actors will be outlined.

Artefact-Actor-Networks are consolidating multilayered social networks and artefact networks in an integrated network. Therefore, we consider the communication and collaboration with each tool (e.g. chat, e-mail and documents) as a single layer of the respective network. We unite these single layers in both social and artefact networks to consolidated networks that contain all actors and artefacts respectively (cf. figure 1). While in the consolidated social network we can only make statements concerning the relations between actors and in the consolidated artefact network we can only analyse the relations between artefacts, Artefact-Actor-Networks (cf. figure 2) also contain semantic relations between actors and artefacts.

This connection enables a new retrieval method for relevant artefacts and persons in complex data sets. With the semantic connection between artefacts and actors it is easy to relate all artefacts to a person it was involved with. On the other hand one can easily find relevant artefacts and involved persons based on each artefact. Thus it becomes more facile to determine experts to certain questions and to find problem-related data. Using the semantic analyses and the ratings of their results mentioned in section V, additional metadata and contextual information for artefacts can be stored and thus be used for clustering artefacts and actors. With the storage of snapshots of the Artefact-Actor-Networks it becomes feasible to reproduce the dynamic evolution of the networks and to gain new insights in group dynamics and working processes within organisations.

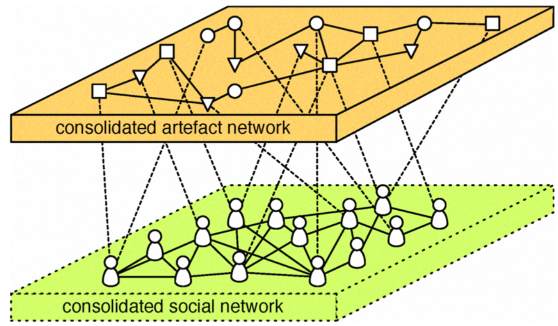

Fig. 2. Artefact-Actor-Network with semantic relations between artefacts and actors

The next two sections introduce semantic relations, present a measure to calculate the similarity between artefacts and shows how to calculate the semantic dense over a whole Artefact-Actor-Network.

IV. Semantic relations

In the context of Artefact-Actor-Networks there exist semantic relations between actors and artefacts (AA relation), actors and actors (ACT^2 relation) and between artefacts and artefacts (ART^2 relation). In this section we present the features of semantic relations in Artefact-Actor-Networks and after that we give an introduction on how to apply metrics from II-B. A semantic relation is a relation which says something about the context of it.

A. Actor-Actor-Relation (ACT^2 relation)

ACT^2 relations describe the nature of relationships between involved persons. They characterise simple connections, friendships or kinships. Furthermore, they can show the kind of media people are communicating with.

The Friend of a Friend (FOAF) project [12] developed a RDF vocabulary, to express interests, connections and activities of persons. It covers constructs to describe properties of persons like *firstname* or *surname* but it is extensible to define own *knows* relations for each person too. Such a reference may point to virtual identities like Twitter- or a ICQ-IDs. The FOAF vocabulary is not sufficient to represent all ACT^2 relations in the Artefact-Actor-Network. For instance, there exists no relation to specify the kind of relation between a person to its

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

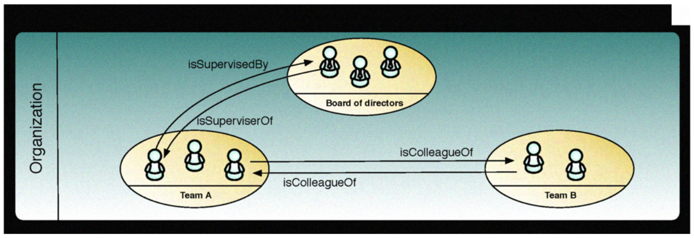

Fig. 3. Examples for ACT^2 relations in an organisation

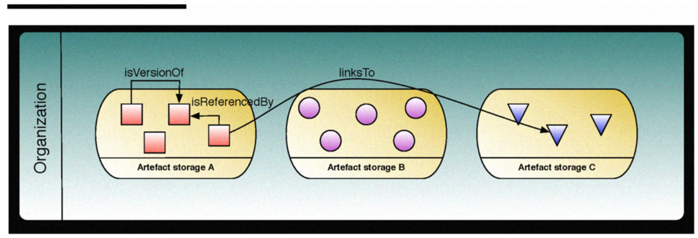

Fig. 4. Examples for ART^2 relations in an organisation

organisation. For these special semantic relationships we have to extend the vocabulary, which can easily be achieved.

As an example for ACT^2 relations may be described as follows. A person is a teammate or a colleague in our scenario. See figure 3. Of course it is true that two teammates are colleagues too. You can see an organisation with three different teams. They can communicate between each other or share different artefacts. All people from team A are teammates among each other, but a person from team B is not a teammate to a person of team A. If one wishes to define the hierarchy in an organisation, one can use self-defined relations of type *isSupervised* or *isSuperviserOf*. In this example you can see that the board of directors are supervising colleagues of team A. Note that we did not indicated all relations in the figure for better survey. You can clearly see that it is easy to extend the set of relations by your own requirements. With specific relation types you are able to analyse different communication channels. People communicate as teammates, colleagues or friends. So you can get information about which kind of communication channels like e-mail, chat etc. people prefer to communicate with their supervisors, teammates or colleagues.

B. Artefact-Artefact-Relation (ART^2 relation)

In the previous section we described semantic relations between persons as individuals. Now we take a look at ART^2 relations between artefacts in artefact networks. ART^2 relations provide information on how artefacts are connected. The Dublin Core standard [13] and the SIOC project [14] provide currently useful expedient kinds of relations. For example artefacts can reference other artefacts with *references* or an artefact can be a derivative of an artefact in an earlier version with *isVersionOf*. Also the relations *hasPart* and *isPartOf* are useful if artefacts exist of multiple parts. Furthermore, we can reuse the relations *replaces* stating that an artefact replaces another one and *requires* saying that another artefact is required to make use of the current one (e.g. a picture within a blog entry). The SIOC project provides a set of useful relations for the context of web applications like e-mails, blogs or bulletin boards. Three important relations to reuse are *replyOf*, which marks an artefact as the reply of another, *linksTo*, expressing the linkage between two artefacts, and *nextVersion*, if an artefact has multiple versions and to represent the history of it.

Figure 4 presents a simple example which shows how artefacts between artefact storages like e-mail, file system or blog entries can be connected. The different symbols represent various artefact types. Different artefacts can be connected between each other; for example a blog entry can link to an artefact of another type.

C. Artefact-Actor-Relation (AA relation)

The last type of relation we use in the concept of Artefact-Actor-Networks are AA relations. This type of relation also has many different semantic shapes. They connect actors with their respective artefacts. For each artefact there exist a set of actors which have semantic relations to it, because artefacts will always be created, modified and annotated by different actors. Therefore, AA relations give information about the kind of connectivity between artefacts and actors. Like in the last section we are able to use Dublin Core and SIOC to get a basic set of sematic relations. The main relations in this context are *creator*, *publisher* and *rightHolder*, which describe who the creator of an artefact was, who published it and who holds the rights on it. The main relations from the SIOC Project are *creator*, *modifier* and *owner*, which describe an actor as the creator, the modifier or the owner of an specific artefact. The LOM standard [15] was created for relations between learning objects and involved entities (persons or organisations). We can use metadata from the LOM standard that describe the way of participation in a learning object. Possible shapes of values are: *author*, *publisher*, *editor* or *initiator*.

We can use different relations of the existing projects for AA relations, but as before, we have to extend the vocabularies to express other roles like *commentator*, *forwarder* or *discussant* out of the environment of current SNSs.

Figure 5 represents different semantic relations that result from the collaborated work between different project spaces in an organisation. For example, actors are owner, modifier or author of an artefact. Project spaces have various artefacts, which can be referenced by actors or other artefacts.

V. Semantic similarity and semantic closeness

Artefact-Actor-Networks are characterised through two main values. Semantic similarity is a tuple based value, between artefacts or actors, which describes the similarity between two objects based on their attributes. The other value, semantic closeness, gives information about the denseness of the network in respect of given semantic terms.

To determine the semantic similarity of two artefacts, we need metadata of the objects. There are numerous ways of obtaining metadata for artefacts. We will not cover these

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

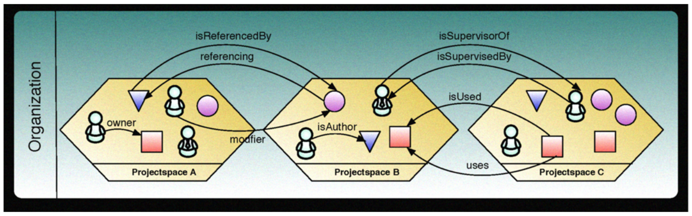

Fig. 5. Examples for AA^2 relations in an organisation

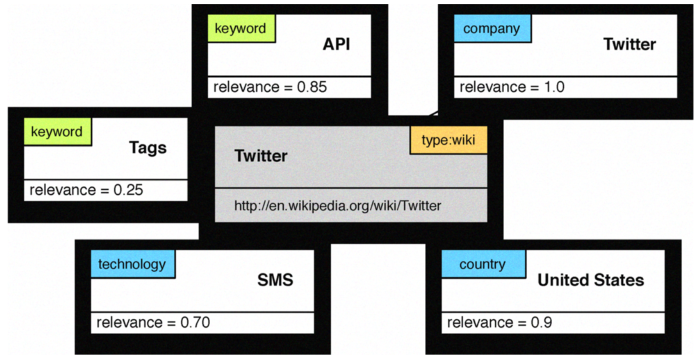

Fig. 6. Relevant keywords and named entities for a wiki artefact about Twitter

possibilities in this paper. Amongst others, the metadata contains semantically relevant things such as keywords or named entities. Semantic metadata can be extracted through external libraries and services like OpenCalais [16] or AlchemyAPI [17]. Picture 6 shows exemplary keywords and named entities (*technology*, *country*, *company*) for a wiki artefact about the Twitter micro-blogging service.

We have to calculate the relevance for every extracted keyword and named entity, which describes the semantical relevance of the metadata for describing the artefact. Several techniques of information retrieval and natural language processing can be used for the calculation of this relevance. One of these techniques is the inverse document frequency (tf-idf) [18], [19], to determine how good a keyword separates an artefact from all other artefacts. Tf-idf uses the fact that if the keyword has a large frequency in the whole set of keywords, it has only small relevance to describe an artefact. Processing of the relevance has to be done in continuous intervals, caused by the fact that tf-idf is based on the existing keyword corpus from the set of artefacts and thus has to be re-calculated as soon as new artefacts have to be stored.

Two artefacts are semantically similar, if the semantic metadata of the artefacts are similar. To determine the semantic similarity, we compare the relevance of the metadata of two artefacts. We distinguish metadata of artefacts in different concepts like *keywords* or *named entities*. Examples for *named entities* are *companies*, *technologies* or *persons*. Every artefact may have several concepts. An artefact interprets its referenced concepts as attributes. By using RDF to represent artefacts, we have no redundantly stored concepts. A concept may be referenced by many artefacts in the network. To compute the similarity between two artefacts, there must exist at least one equal concept between them. Otherwise the semantic similarity is zero. For a better understanding of our concept we divide the process to calculate the semantic similarity into short steps.

1) Relevance of concepts for an artefact: As discussed previously an artefact may have arbitrary many concepts with specified relevances. Services like OpenCalais [16] and AlchemyAPI [17] deliver information about keywords and named entities with their respective relevance for the artefact. Directly extracted keywords can be weighted through information retrieval methods like tf-idf.

2) Normalising of relevances: The relevance of the attributes are absolute values with no respect to other attributes. But to compute the semantic similarity between two artefacts it is necessary to normalise the values to get the weight of one relevance in respect to all others. In our approach we normalise the attributes to the value 1. Denote that all relevance factors are mapped into the continuous interval $(0, 1]$. The sum of all relevances is at most one.

3) Computation of semantic similarity: To compute the similarity between two artefacts, we take into account all common attributes of the artefacts. Pairwise, the difference between the normalised values is calculated and weighted by the minimum of the normalised values of both attributes. Then all pairs will be summed up. The resulting value is the similarity of both artefacts in respect to the weight of their attributes.

Definition Let A and B be two artefacts from the Artefact-Actor-Network with at least one common concept and let $NRel(A_{c_i})$ denote the normalised semantic relevance at artefact A of the i-th common concept of A and B. Then the semantic similarity of A and B $SemSim(A, B)$ is defined as:

$$\sum_{i=1}^{\#\{cc\}} \left[\min\left(NRel(A_{c_i}), NRel(B_{c_i})\right) \cdot ConSim(A_{c_i}, B_{c_i})^2 \right] \tag{1}$$

with

$$ConSim(A_{ci}, B_{ci}) = 1 - |NRel(A_{c_i}) - NRel(B_{c_i})| \tag{2}$$

As you can see the plot in figure 7 of $SemSim(A, B)$ is a linear function. For a common concept between two artefacts A and B the relevancies are on the x- and y-axis. The value of the semantic similarity is represented by the z-axis. If the relevance x equals y, the semantic similarity is maximal for a given conecpt.

For example a common concept *SMS* which is a technology, the relevance of this concept must not necessarily be equal to both artefacts. If the relevancies are same, then $ConSim$ returns 1, which means that the semantic similarity value will not be weakened, because the current concept is identically important to both artefacts. The minimum of the normalised relevancies in the first part of the formula guarantees, that the

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

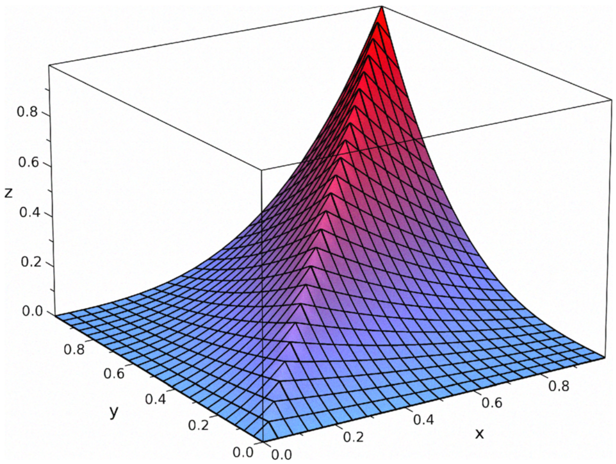

Fig. 7. Plot of the $SemSim$ formula

semantic similarity value in every iteration is not greater then the smallest relevance. If two artefacts have the same concepts and for every concept equal relevance, then it must be that the semantic similarity is exactly 1. Differences on relevancies for common concepts affect alleviative to the semantic similarity between two artefacts. In an evaluation process we decided to square $ConSim$ which means that a small difference of the relevance will affect less alleviative.

In the next subsequent section we present how to compute semantic closeness in the whole Artefact-Actor-Network. It allows you to detect semantic groupings with search terms referencing to conecpts with the help of edges generated from $SemSim$.

4) Computation of semantic closeness: The semantic closeness of the Artefact-Actor-Network only examines edges of the network which are referencing to a given term as a parameter and uses these edges to compute the network closeness in fact of the term. Note that a term is pointing to a set of concepts of the Artefact-Actor-Network. This is done by calculating the ratio between the existing edges and the possible edges and is called the selectivity. Selectivity denotes how strong the network is connected. A high semantic closeness denote a high importance of the term in the network.

Definition Let E_{term} denote the edges that process the term and E are all possible edges. Then the semantic closeness of the Artefact-Actor-Network with respect to a given search term $SemClose(term)$ is defined as:

$$\frac{2 \times E_{term}}{E^2 - E} \tag{3}$$

5) Practical example: With relations that describe the semantic similarity between artefacts Artefact-Actor-Networks are interesting in practise. Let there be a set of Wiki articles. Every article is represented as an artefact. With our approach artefacts are connected, if the semantic similarity between them is greater than zero. That implies, that contextual similar articles are connected between each other automatically and can be treated or visualised together. This gives the reader of an article the ability to enlarge upon his topic by getting access to semantically similar content. Furthermore, the reader can find like-minded people, because artefacts are connected to persons.

VI. Use cases for Artefact-Actor-Networks

As introduced the concept of Artefact-Actor-Networks, semantic relations and similarity measures in previous sections, we will now give a brief of practical applications for Artefact-Actor-Networks.

In this section we present three diverse use cases for the application of the Artefact-Actor-Network approach. In doing so, specific semantic requirements for Artefact-Actor-Networks get derivated and the value-added through the application of Artefact-Actor-Networks are discussed.

A. Blogosphere

The blogosphere denotes the collection of all blogs (and micro-blogs) and their interconnection [20]. Blogging and micro-blogging provide dynamic content that represents the opinions and beliefs of the bloggers and commentators. The analysis of the blogosphere helps in understanding how memes spread within distinct communities, relevant sources of information and other phenomena. Current approaches (e.g. [21], [22]) for the analysis of the blogosphere make use of explicit hyperlinks between blogs. From the analysis of artefacts from the blogosphere and their conversion in an Artefact-Actor-Network we hope to gain a deeper understanding of the roles of the persons involved. Artefact-Actor-Networks will help us to determine the social importance of an artefact and outline semantically similar artefacts and bloggers.

From the analysis of blogs and micro-blogs a specific set of semantic relations arises that can represent the connections between artefacts from the blogosphere and the bloggers. The most important AA relation is *isCreatedBy*, which describes that a certain artefact was created by a certain blogger. The relations *inReplyTo*, *isForwardOf*, *linksTo* und *isCommentTo* are ART^2 relations and describe the typical blogosphere actions of answering, citing, linking or commenting of an artefact.

One major goal of adopting Artefact-Actor-Networks within the blogosphere is to allow an easy retrieval of blogs and micro-blogs that are semantically similar to each other without explicitly linking each other. For the well-known microblogging service Twitter [23] this would mean to identify other users that tweet about semantically similar content, link on the same blogs or re-tweet the same users. With the application of Artefact-Actor-Networks in the blogosphere is becomes feasible for the first time to connect blogs with micro-blogs and taking into account user actions like commenting, linking or forwarding of artefacts. Furthermore, it will be easy to derive a large number of artefacts linking to a specific artefact.

B. Collaborative Software Engineering

In the use case of collaborative software engineering we have to cope with many different artefacts that distinguish in

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

content, organisational structure, composition and usage aim. Besides source code there are specification documents, wiki articles, entries in bug and issue trackers, calendar entries, e-mails and many more. Enlisted persons of a software project can act in diverse roles and can be involved in various ways in the change of artefacts.

There are special ART^2 relations in the case of collaborative software engineering like *isVersionOf*, *extendsArtefact*, *isBugreportFor* or *documentsArtefact*. Between team members specific ACT^2 relations like *areTeammembers* or *isPairProgrammingMateOf* can exist. The specific AA relations can express semantics like *isProductOwner*, *isDeveloperOf*, *isReporterOf*, *addedEvent* and many more.

Our main aim with adopting Artefact-Actor-Networks to the use case of collaborative software engineering projects and teams is to increase artefact awareness and transparency in software engineering projects by noticeably highlighting the connections between artefacts and users.

In [24] we present our vision of a community-embedded collaborative development environment that make use of Artefact-Actor-Networks in more detail. Having the semantical connections between artefact, we can derive topic-connected artefact networks that can help solving specific problems during specification, implementation, or documentation. Using the mutual relations between intra-project and inter-project developers and artefacts, it becomes feasible using the experience and knowledge of experts from outside the own organisation and thus enhance the personal learning and working in software engineering projects.

C. Wikis

Recently wikis have become a major mean of knowledge management in many companies and organisations. As an essential part of corporate knowledge management wikis will have to fulfil special requirements with regard to information retrieval and expert recommendation. But a major downside of wikis is their lack of artefact and user awareness. So wikis will not provide a user with similar articles based on the semantic content of an article or present active users within a special knowledge domain that can act as experts in the specific field. With the concept of Artefact-Actor-Networks it is possible to support viewer and writer of wiki articles. By interpreting every wiki article as an artefact, we can define semantic relations between them. If we have various wikis in an organisation, every wiki article will be stored as an artefact in detached artefact stores. From section IV follows, that these artefacts may be connected via semantic relations. Every wiki article is connected to an author. To simplify matters we neglect different versions of an article. The various Wikis in an organisation are mapped to project spaces. With a wiki reader and writer extension it is possible to show viewers of an article semantically relevant articles or authors have written relevant articles. In position of the author the writer extension enables a feature to show relevant articles during the writing process. The main advantage is the ability to show readers and writers of articles semantically relevant information over various project spaces of an organisation.

VII. Implementation of Artefact-Actor-Networks

The backend system of Artefact-Actor-Networks is implemented as modularised component system based on the OSGi service platform [25]. We designed the backend system easily extensible by future protocols and functions. Different crawler components for data provisioning make it possible to inspect simple HTML pages as well as pages with specific structures such as MediaWiki [26] pages or WordPress [27] blog entries and their specific metadata. We use the Jena Framework [28] to efficiently store, manage, process, and retrieve semantical RDF data. With SparQL [29] the Jena Framework offers a very efficient query language for the data pool that not only can queried by rule-based engines but also via the built-in inference engine. We used this mechanism to represent the different layers of artefact networks and actor networks by inheritance of RDF types.

The storage engine additionally triggers other analysers that can be split into two groups: text analysers and network analysers. The former are investigating the full text of artefacts in order to derive additional data like keywords, topics or named entities. We used services like OpenCalais [16] and Orchestr8 [17] to derive these data. Both of the projects expose their functionality via a web service API. Preliminary test show that the services yield good results for texts longer than 300 characters. The network analysers belong to the second group of analysers and are only operating on the stored RDF data (e.g. the component that computes the semantic similarity between two artefacts).

A. The backend system of Artefact-Actor-Network

The backend system of Artefact-Actor-Network was designed to ensure extensibility and flexibility for future enhancements. In order to assure flexibility to the greatest extend possible we choose the OSGi service platform that allows to add or remove components during the runtime. This makes it possible to add to the running application additional parsers for specific data formats, accessor modules for further protocols like SMTP, SVN or FTP as well as fresh data analysers using different services or calculating the inverse document frequency (tf-idf).

The backend system is mainly composed of three components cf. figure 8): 1) the crawler component that ensures the data provisioning, 2) the datastore component, which is responsible for the storage, retrieval and reasoning, and 3) the analyser component. The access to the system as granted via a web service interface and a frontend that makes use of this interface. The single components are using mechanism of the OSGi service platform to communicate with each other.

The crawling subsystem is addressed via the superordinate crawler component that provides an OSGi service with the according interface. The crawler expects jobs with data sources that are to be accessed. For each data source a selection of components is created that is needed to finish the job. For

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

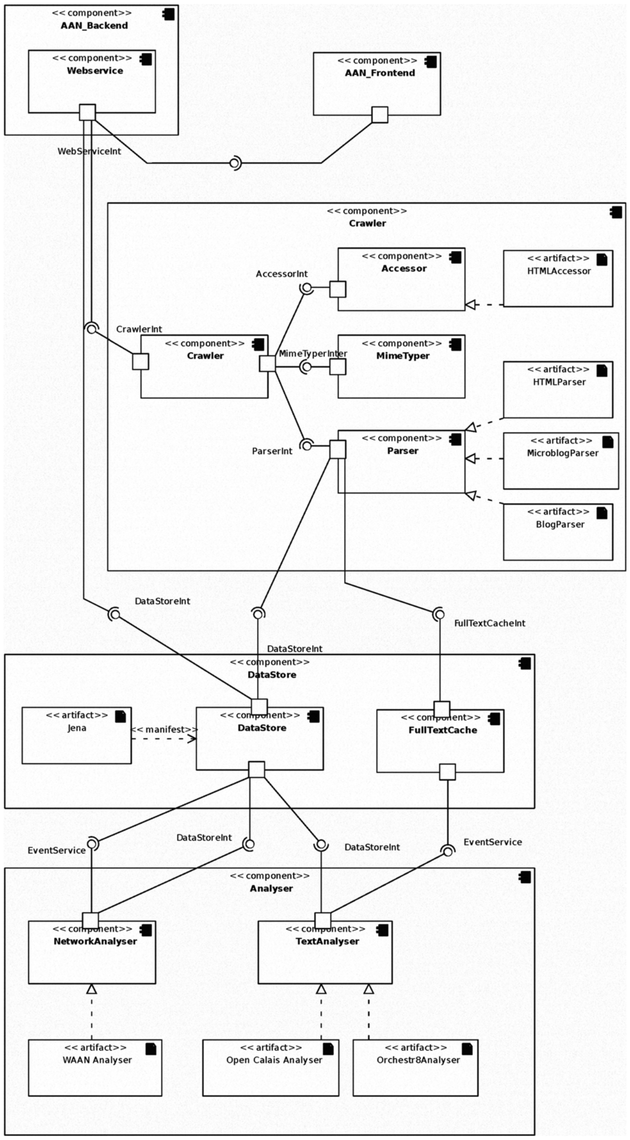

Fig. 8. Component model of the Artefact-Actor-Network backend system

accessing a simple HTML page you would need an accessor that can handle the according protocol (HTTPAccessor), a mimetyper to ease the selection of a parser and a parser that extracts data from the data source according to its mimetype (HTMLParser). The parser then writes the extracted metadata into the datastore and the full text into the FullTextCache component that both are reachable via OSGi services.

The data storage subsystem is responsible for the storage of all data and provides a FullTextCache component that store the full texts of selected text-based artefacts, which can be fetched by the different text analysers. The full texts are cached for a certain amount of time and can be re-downloaded if required. This mechanism is useful for very large text inventories that do not need to be stored permanently. The second component of this subsystem is the datastore, which basically stores all extracted and created metadata for artefacts and actors and ensures the reasoning based on an inference engine provided by the Jena framework. Furthermore the Jena Framework allows to have the data in memory, stored in the filesystem or in a relational database. The access to stored or inferred data is provided via a SparQL interface or with the Jena model. Every time when data is written to the data storage component, an event in the built-in OSGi event service is fired where for example the analysers are listening to. This service works according to the whiteboard pattern [30] meaning that the analysers provide an event service that is searched and used by the event-launching component

The main duty of the third component is to enhance the data gained during the crawling processes by additional metadata and further statements about artefacts, actors and the whole Artefact-Actor-Network. All analysers that are linked to the data storage component via the OSGi even service are listening for events that they can handle. The analysers currently are split into two groups that differ in the observed events; text analysers are listening to events from the FullTextCache while network analysers are listening to changes in the datastore. The former analysers are investigating the full texts of artefacts and extract keywords and named entities that they store as RDF data in the datastore. The latter analysers can compute the semantical similarity between artefacts and thus enhance the information quality of the whole Artefact-Actor-Network.

B. RDF model and inference of networks

RDF models provide the possibility to inherit RDF types. This capability and the option of the Jena framework to directly access single stages of the inheritance chain make it possible to model the different layers of Artefact-Actor-Networks. Figure 9 shows a narrowed version of our RDF model. The left side of the models shows our basic structure of the Artefact-Actor-Network (AANBase). Here we defined basic types for actors, artefacts and fundamental relations between them. All other specialised types are derived from these basic ones. Figure 9 exemplary depicts the WebArtefact which inherits from Artefact and itself is the upper type of MicroblogArtefact. MicroblogActor however inherits from the basic type Actor. Based on this type of structuring the data, it becomes feasible to store a MicroblogArtefact in the datastore and to infer that it is a WebArtefact and thus belongs to the Artefact-Network. We can proceed in the same way with a MicroblogActor who is part of the Actor-Network due to the inheritance. This way we not only can discern the two main consolidated networks but also all the sub-networks originate from the inheritance of types. For example we are able to distinguish between a network of WebArtefacts and one of MicroblogArtefacts. If we implement a TwitterArtefact and a IdenticaArtefact that specialise a MicroblogArtefact one can even distinguish between those the two new networks that are spanned by the according artefacts.

This structure of inheritance not only covers the inheritance of RDF types but also of relations. That way the relation

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

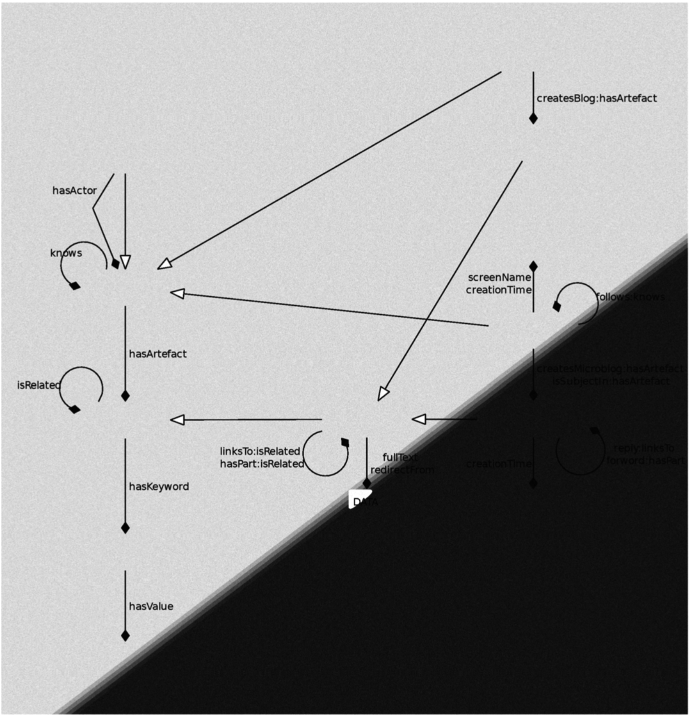

Fig. 9. Artefact-Actor-Network RDF-Model

linksTo from the Web model inherits from the relation *isRelated* from AANBase. That way it is possible to make general claims about the relations of all artefacts in the Artefact-Network.

The Jena framework uses rule-based reasoners to create this structure of inheritance and makes it searchable. At this we are using a weak form of reasoners that is restricted to the possibilities of RDF-schema (RDFS, [31]). RDFS contains amongst others the properties *subClassOf* and *subPropertyOf*. Stronger attributes from OWL [32] are not necessary for the hierarchical structure. Using this function we are able to store all relevant data in a common, standardised data structure but can distinguish between the different layers of the Artefact-Actor-Network at the same time.

C. Keyword extraction using external services

We use OpenCalais [16] and Orchestr8 [17] to retrieve additional metadata not already extracted during the crawling and parsing process. Therefore these services use advanced techniques of natural language processing (NLP) and information retrieval that are not covered in this paper. Each service provides an API that accepts text, extracts metadata and returns this data as RDF coded data. Both services are capable of extracting keywords and key phrases, named entities such as persons, technologies, countries or companies. Furthermore they categorise analysed texts to a own taxonomy. All these information is stored as special forms of the keyword type shown in figure 9.

VIII. Conclusion and Future Work

In this paper we have envisaged our approach of Artefact-Actor-Networks, which connect social networks and artefact networks through semantic relations. With Artefact-Actor-Networks we contribute to the field of computer mediated communication and computer-supported cooperative work a model that allows the storage and analysis of multi-layered communication and collaboration in digital networks. By offering an open and extensible set of semantic relations in the ART^2, ACT^2 and AA relations we enable the adoption of Artefact-Actor-Networks in multiple use case scenarios and domains.

The analysis of a specific Artefact-Actor-Network with respect to its semantic closeness enables new ways of analysing and visualising cooperative work. The augmented awareness of group processes and the transparent semantic connections between users and artefacts will ease the understanding of problems and contextual support as well as expert finding and expertise rating. Furthermore, the joining of social networks with artefact networks enable new possibilities in the assistance of informal learning processes. By employing the metrics from SNA in the context of Artefact-Actor-Networks, relevant artefacts and persons can be filtered from a huge dataset.

First prototypes showing the prototypical application of Artefact-Actor-Networks within the domains of wikis and the blogosphere are currently under development and will be ready for presentation with the final version of this paper. Furthermore, we are working on ways of visualising the dynamics of Artefact-Actor-Networks in order to make claims about changes in the social network or the importance of single artefacts and their impact on the alteration of the network.

Acknowledgment

Part of the research reported in this paper conducted in relation to the work of the MATURE project funded by the EU under contract no. 216356 [33]. It was also partially supported financially by the MATURE project. We would like to thank the anonymous reviewers for their valuable comments and all that helped us with valuable discussions and critical questions.

References

[1] CollaborateCom 2009 Programme Committee, "Call for paper for the CollaborateCom 2009," http://www.collaboratecom.org/docs/CollaborateCom09_cfp.pdf, 2009.

[2] J. C. Mitchell, *Social Networks in urban situations: Analyses of personal relationships in Central African towns.* Manchester: University Press, 1969.

[3] B. Holzer, *Netzwerke [Networks].* transcript, 2006.

[4] K. K. Cetina, "Sociality with Objects: Social Relations in Postsocial Knowledge Societies," *Theory Culture Society*, vol. 14, no. 4, pp. 1–30, 1997.

[5] J. Bowers and S. D. Benford, Eds., *Studies in Computer-Supported Cooperative Work: Theory, Practice, and Design - Human Factors in Information Technology.* Amsterdam: Elsevier Science, 1991.

[6] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: Some issues and experiences," *Communications of the ACM*, vol. 34, no. 1, pp. 39–58, 1991.

[7] S. Teufel, C. Sauter, T. Mühlherr, and K. Bauknecht, *Computerunterstützung für die Gruppenarbeit [Computer support in group work].* Bonn: Addison-Wesley, 1995.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308

[8] B. Wellmann, "Applying network analysis to the study of support," in *Social Networks and Social Support*. Gottlieb, 1981, pp. 171–200.
[9] L. S. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, pp. 215–239, 1979.
[10] I. Herman, "Semantic web activity statement," http://www.w3.org/2001/sw/Activity.html, retrieved on 2009-07-17, June 2009.
[11] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American Magazine*, May 2001.
[12] FOAF, "The Friend of a Friend (FOAF) project," http://www.foaf-project.org/, retrieved on 2009-05-17, 2009.
[13] DCMI, "Dublin Core Metadata Initiative," http://dublincore.org/, retrieved on 2009-05-19, 2009.
[14] SIOC, "The semantically-interlinked online communities (sioc) project," http://sioc-project.org/, retrieved on 2009-05-24, 2009.
[15] Learning Technology Standards Committee Learning Standards Committee of the IEEE, "Draft standard for learning object metadata," http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, retrieved on 2009-05-17, 2002.
[16] T. Reuters, "Calais: Connect. everything," http://www.opencalais.com/about, retrieved on 2009-06-20, 2009.
[17] Orchestr8 LLC, "Alchemyapi," http://www.alchemyapi.com/, retrieved on 2009-07-17, July 2009.
[18] G. Salton and M. J. McGill, *Introduction to modern information retrieval*. McGraw-Hill, 1983.
[19] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
[20] M. Keren, *Blogosphere: The New Political Arena*. Lexington Books, 2006.
[21] Technorati Inc., "Technorati," http://technorati.com/, retrieved on 2009-07-17, July 2009.
[22] University of Toronto, "Blogscope," http://www.blogscope.net/, retrieved on 2009-07-17, July 2009.
[23] Twitter Inc., "Twitter," http://twitter.com, retrieved on 2009-07-17, July 2009.
[24] W. Reinhardt and S. Rinne, "An architecture to support learning, awareness, and transparency in social software engineering," in *Forthcoming: Proceedings of the Special Track on MashUps for Learning at ICL 2009*, M. Ebner and S. Schaffert, Eds., 2009.
[25] O. Alliance, "Open services gateway initiative (osgi)," http://www.osgi.org, retrieved on 2009-09-23, 2009.
[26] Wikimedia Foundation, "Mediawiki," http://www.mediawiki.org, retrieved on 2009-09-23, 2009.
[27] WordPress, "Wordpress," http://wordpress.org/, retrieved on 2009-09-23, 2009.
[28] Jena, "Jena – A Semantic Web Framework for Java," http://jena.sourceforge.net/, retrieved on 2009-05-17, 2009.
[29] W3C, "Sparql query language for rdf," http://www.w3.org/TR/rdf-sparql-query/, retrieved on 2009-05-30, 2009.
[30] OSGi Alliance, "Listener Pattern Considered Harmful: The "Whiteboard" Pattern," OSGi, Technical Whitepaper, 2004.
[31] W3C, "Ressource description framework schema (rdfs)," http://www.w3.org/TR/rdf-schema/, retrieved on 2009-09-23, 2009.
[32] W3C, "Web ontology language (owl)," http://www.w3.org/TR/owl-features/, retrieved on 2009-09-23, 2009.
[33] MATURE, "MATURE. A large scale integrating project," http://mature-ip.eu/, retrieved on 2009-09-23, 2009.
[34] T. M. H. Reenskaug, "Model-view-controller pattern," http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html, retrieved on 2009-07-24, 1978.
[35] V. D. Veksler, A. Grintsvayg, R. Lindsey, and W. D. Gray, "A proxy for all your semantic needs," in *29th Annual Meeting of the Cognitive Science Society, CogSci2007*, 2007.
[36] M. Krötzsch, "Semanticweb.org," http://semanticweb.org, retrieved on 2009-07-17, July 2009.
[37] P. Scheir, C. Ghidini, and S. Lindstaedt, "Improving search on the semantic desktop using associative retrieval techniques," in *working notes of The Innovation Conference for Knowledge Management, New Media Technologies and Semantic Technologies (TRIPLE-I)*, 2007.
[38] C. Thomas and A. P. Sheth, "Semantic convergence of wikipedia articles," in *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 600–606.
[39] Wikipedia, "Social network (version as of 2009-05-20, 12:40)," http://en.wikipedia.org/w/index.php?title=Social_network&oldid=291166313, retrieved on 2009-05-22, 2009.
[40] S. Feld, "Social Structural Determinants of Similarity among Associates," *American Sociological Review*, vol. 47, 1982.
[41] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press., 1994.
[42] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, no. 1, pp. 60–67, 1967.
[43] M. Granovetter, "The strength of weak ties: A network theory revisited," *Sociological Theory*, vol. 1, pp. 201–233, 1983.
[44] M. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.
[45] W3C, "Ressource description framework (rdf)," http://www.w3.org/RDF/, retrieved on 2009-05-30, 2009.
[46] E. V. J. Ian Davis, "Relationship: A vocabulary for describing relationships between people," http://vocab.org/relationship/.html, retrieved on 2009-05-24, 2009.
[47] A. Richter and M. Koch, "Funktionen von social-networking-diensten [functions of social networking services]," in *Proceedings der Multikonferenz Wirtschaftsinformatik*, 2008.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8308
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8308