SAM: Semantics Aware Instant Messaging for the Networked Semantic Desktop

Thomas Franz and Steffen Staab

ISWeb, University of Koblenz-Landau, Germany
{franz,staab}@uni-koblenz.de

Abstract. While instant messaging (IM) became a mature communication means in business organizations over the last years, IM systems did not follow this evolution comparably. Communicated content is often stored insufficiently and hard to recall, integration into other desktop applications impossible. In this paper, we address these shortcomings and provide concepts for novel instant messaging. In contrast to prior work such as the Haystack system, which integrates IM data into a personal information management application, we enhance IM based on a ready to integrate ontological meta model that introduces semantics to instant messaging and its content to foster advanced management. In particular, we address networked exchange of semantic meta information to integrate IM into the Networked Semantic Desktop. The Semantics Aware Messenger (SAM) is a prototypical implementation of the concepts presented in this paper.

1 Introduction

The objective of the Semantic Desktop is to improve personal information management (PIM) by combining all content available on the desktop and relevant to the user to i) easily manage that content, regardless of which type, and to ii) simplify utilization of it.

The Networked Semantic Desktop as envisioned in [4] describes a networked infrastructure that combines the Semantic Desktop with Social Networking and P2P systems to benefit novel applications such as group collaboration.

In this paper, we address instant messaging (IM) on the Networked Semantic Desktop. In contrast to prior work such as the Haystack system [10], which integrates various desktop sources into a consistent, meta data driven personal information management application, we enhance IM based on a ready to integrate ontological meta model that introduces semantics to instant messaging and its content to foster advanced management. In particular, we address networked exchange of semantic meta information to integrate IM into the Networked Semantic Desktop.

Today, communicating by instant messaging mainly comprises typing messages and viewing incoming messages, while most of the time, no further processing of messages is done or offered so that message content gets lost in plain text communication logs that are more or less accessible depending on the client application. Despite poor traceability, recent studies claim that IM usage has matured and IM is employed for miscellaneous tasks including complex conversation [11, 9]. Accordingly, we consider the content communicated via this media as of increasing value that should be recallable and integrated into the Networked Semantic Desktop.

The objective of this paper is to tackle traceability shortcomings of IM, improve management of IM content, and move IM towards the Networked Semantic Desktop. We are proposing i) an ontological meta model for instant messaging which ii) supports integration into the Networked Semantic Desktop, and iii) introduces meta data and semantics for IM to enable iv) sophisticated reutilization of instant messaging data. Based on the meta model and an v) identification scheme for IM data including meta information and semantics we vi) enable networked exchange of such information via IM to vii) ground novel applications as envisioned in [4].

In Sect. 2, we sketch a typical IM scenario to indicate shortcomings of current IM systems (Sect. 3), explain our concepts to overcome these shortcomings (Sect. 4), and illustrate the implementation of these concepts by examples of that scenario in Sect. 6. We give a detailed overview of the ontological meta model in Sect. 5, and contrast our work with related work to provide a conclusion in Sect. 7. In Sect. 8, we suggest future research and give an outlook.

2 Scenario

The extracts of chat conversations in this section render a common instant messaging scenario and indicate different functions and particularities of IM.

The scenario: Steffen, being the lecturer of the Semantic Web lecture uses IM to get some quick responses concerning organizational issues from Thomas, who held the last exercise session for the lecture:

```
[09:29:06] Steffen: how was the exercise session?
[09:29:33] Steffen: did you tell them the date of the exam?
[09:30:33] Thomas: solutions were ok, participation was weak
[09:30:56] Thomas: yes, i guess about 20 will sign up for it
```

Listing 1.1. Exercise Session

At a later time, Thomas informs Steffen about his work on a paper he is writing for the Semantic Desktop Workshop.

```
[12:07:45]  Thomas: i will put new versions of the paper for
the sdws at http://isweb.papers.x.y/sam.tex
[12:08:18] Steffen: ok, what is going to change?
[12:08:35] Thomas: describe an IM scenario to indicate current
shortcomings, propose improvements, and demonstrate SAM
in terms of the scenario
```

Listing 1.2. Semantic Desktop Paper

After lunch, Thomas contacts Steffen about the scenario he mentioned in Listing 1.2.

```
[13:55:04] Thomas: any ideas for a suitable scenario?
[13:55:24] Steffen: why don't use this conversation?
[13:55:43] Thomas: right! its a sufficient example of a
work related chat
[13:56:15] Thomas: i'll use our today's earlier chats
as well. they nicely indicate different functions of IM
```

Listing 1.3. Scenario for the Paper

Later on, Steffen talks to Bernhard, a co-worker in project X:

```
[17:15:42] Steffen: wrt the project you might be
interested in what thomas is currently doing; i'll send
you what thomas told me about that so far
[17:16:03] Bernhard: thanks, i'll contact him when
i've read it
```

Listing 1.4. Project X Work

2.1 Terminology and Observations

Isaacs et al. [9] discovered that – in professional environments – IM messages mostly are work-related (61.8%), followed by scheduling and coordinating ones (30.8%) and those that resemble simple questions and information (27.8%).[1]

Based on that terminology, we classify the chat excerpts (Listing 1.1 to 1.4) as follows: Listing 1.1 is a sample of *simple questions and information*, while Listings 1.2, 1.3, 1.4 represent *work-related* messages. In the given scenario, we excluded scheduling/coordinating conversations, as they resemble a typical IM function, but do not contribute much here.

2.2 Use Cases

Due to the fact that most IM conversations are about work, the content of such conversations needs to be available for later reuse as illustrated by the two following use cases.

Use Case 1: About one week after the day when the listed conversations took place, Steffen wants to check where Thomas stored that file on the server, and what exactly he stated about his current work. As Thomas is not available he cannot ask him again.

Use Case 2: In order to track project development, and summarize the current stage of project X, Steffen wants to compile all project X related content, including messages that deal with the project.

[1] Messages could be classified for more than one category.

3 Accomplishing the Use Cases Today

Today's instant messengers usually store messages in plain text logs and provide a user interface to view the logs, sometimes ordered by message date or filtered by user. More sophisticated messengers may supply an additional search over the message logs. Accomplishing the use cases with current systems reveals the following shortcomings:

1. Weak Message Classification:
   Finding appropriate messages by browsing the message logs requires high user effort as the given classifications (by user, by date) do not narrow the search space enough to easily find messages: Given that Steffen does not recall the exact day when Thomas told him about his current work, he has to read all the messages from several days to find the one he seeks.
2. Keyword search is unsuitable due to missing content semantics and particularities of chat conversation style:
   (a) A term denoting the subject of a message, or significantly distinguishing a message from others is not necessarily contained in a message so that creating efficient search strings is delicate. Entering a query that finds the message Steffen looks for in use case 1 may be difficult as Thomas did not use keywords like "store", "server", or "file" that directly relate to the semantics of his message in Listing 1.2.
   (b) Ambiguity of search terms further decreases the average relevance of search results. If Steffen searches for *paper*, he may receive messages that deal with different concepts of paper such as *writing paper*, *abrasive paper*, and *research paper* while only the latter is relevant for him.
3. Missing Context:
   (a) Instant messages are rather short, and informal [7, 6, 12] therefore become meaningless without context. In Listing 1.1, Thomas said "the solutions were ok, participation was weak". Without the message's context it is hard to predict which solutions Thomas points at. Current IM systems do not provide message context so that identifying relevant messages is difficult.
   (b) Topic switching and interleaving messages are particularities of IM conversation. Listing 1.1 has interleaving messages, as Thomas' first message replies to Steffen's first message although it appears after Steffen's second message. The context of interleaving messages is not based on the sequence in which they appear in time so that even browsing message logs ordered by time does not necessarily provide relevant context.
4. Missing Messaging Semantics:
   Current IM clients do not identify message properties, e.g. the creation date, or sender of a message. Consequently, relations between them cannot be exploited:
   (a) Missing messaging semantics inhibit integration into the Networked Semantic Desktop.
   (b) Information exchange is of low value as just meaningless plain text can be exchanged.

(c) Semantic querying using restrictions on properties is impossible, e.g. querying for messages within a date range, sent by a certain user et cetera.

4 Improvements by SAM

4.1 Message Classification for Message Semantics

The first shortcoming mentioned in Sect. 3 denotes weak message classifications provided by current IM clients. SAM offers a user-definable taxonomy that is used to add semantics to messages by annotating them with entries from the taxonomy. For instance, Steffen might define the category *work* with two subcategories *teaching* and *projectX*. If he annotates any message related to project X with the corresponding entry in the taxonomy, accomplishing the second use case is as easy as browsing for all messages annotated with *projectX*. Message classification also benefits search, as queries can restrict search results to be annotated with certain taxonomy entries. How annotations and the taxonomy are designed is detailed in Sect. 5, how the user annotates with SAM is explained in Sect. 6.2.

The main drawback of message classification is the user effort required to annotate messages appropriately. This effort is lowered by automatic annotation exchange between conversation partners as detailed in Sect. 4.3 and 6.4, however, manual annotation still has to be done by at least one of a conversation's participants in order to gain benefits. The user interface of SAM tries to minimize this effort as much as possible (see Sect. 6.2) and for future work we propose to integrate automatic message classification based on machine learning technologies.

4.2 Ontological Meta Model

We employ a meta model for instant messaging in form of a unified messaging ontology (cf. Sect. 5) that tackles many of the shortcomings listed in Sect. 3.

The ontological meta model provides semantics for IM entities such as persons, messages, conversations, annotations, and message texts as it identifies and relates such entities to each other by meaningful properties. This permits several enhancements as detailed in the following:

Message Context: Any message is accompanied by its context, i.e. messages link to their following message, their sender and recipient and so on. Accordingly, messages displayed while browsing or in search results are much more informative thus reducing the user effort of determining whether or not they are relevant.

Semantic Querying: Querying becomes more powerful as the ontological meta model permits to define what to query for, e.g. one can not only query for messages but also for users or taxonomy entries. Moreover, restrictions on properties can be defined, e.g. Steffen can request messages sent by Thomas within a certain date range, including the keyword "paper" in their message text. Resulting messages will directly link to related messaging entities to provide context.

Integration: As the ontology unambiguously defines messaging entities it integrates IM into the Networked Semantic Desktop by providing interoperability between applications. For instance, the sender of a message in Steffen's store can be identified as the author of a document on his hard disk, or the sender of an email in his email client. Such features require, however, that applications commit to the same ontology. Thus, SAM does not employ a proprietary representation of persons, but integrates the Friend-of-a-Friend (FOAF[2]) ontology as it is widely recognized for expressing identity.

The ontology abstracts the concept of a message considering interoperability of different message channels as proposed in [13]. A unified view of messaging aims at seamless integration between different messaging applications as it allows to track conversations that comprise different message types and message channels, e.g. receiving an email message and answering with an instant message.

4.3 Meta Data Exchange

All participants of a conversation deal with the same set of messages. As each user decides how to annotate a message and which concepts to have in his taxonomy, there are cases where annotations differ between users, and where one user annotated a message while the other one did not. A common meta model on each peer, unique identification of IM entities, and provenance information established by the messaging ontology enables automatic annotation exchange between peers to either add further message semantics through additional annotations, or add annotations for not yet annotated messages. The latter case is especially important to reduce annotation effort for the user. As each user maintains his own taxonomy, annotation exchange may also introduce new taxonomy entries. SAM offers different user options to deal with incoming annotations as explained in Sect. 6.2. Technical aspects of meta data transfer are mentioned in Sect. 6.4.

Meta data exchange is not only useful to decrease annotation effort, it permits several novel applications. In Listing 1.4, Steffen tells Bernhard to send him, what Thomas told him. Meta data exchange as proposed by SAM allows to automatically integrate messages sent between Thomas and Steffen into Bernhard's data store so that Bernhard can utilize all features of SAM to access these messages.

5 The Ontology

Figure 1 depicts the ontology and defines the namespaces used for the following textual explanation of the ontology. A conversation is modeled by the class

[2] http://www.foaf-project.org/

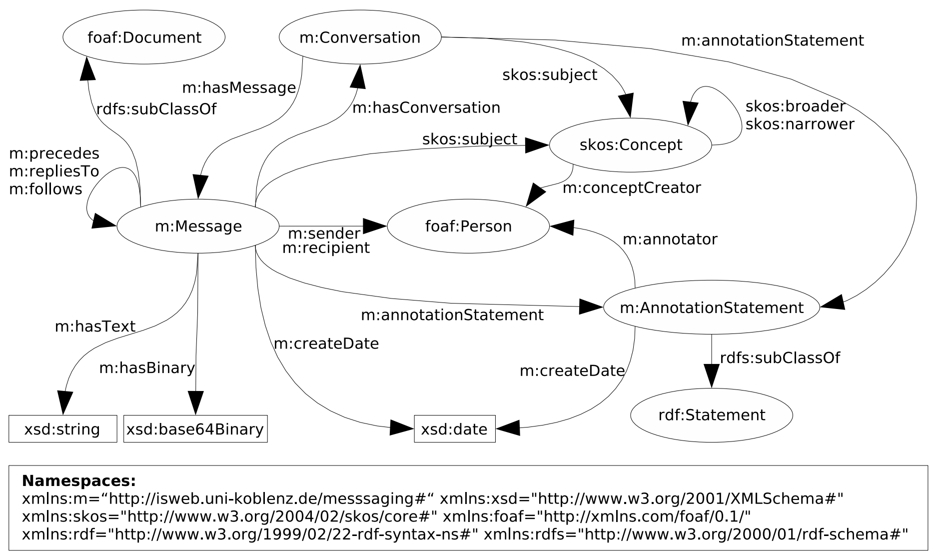


Fig. 1. Unified Messaging Ontology of SAM

`m:Conversation`, which relates to messages exchanged within a conversation by the `m:hasMessage` property. A message is a subclass of `foaf:Document` and is associated to its content by the `m:hasText` and `m:hasBinary` properties. The `m:follows` property and its inverse, `m:precedes`, track the chronological order in which messages appear, while the `m:repliesTo` property records further valuable context information that goes beyond chronological ordering: It relates a message to the message it replies to thus relating these messages based on the semantics of their content. This property is significant to store appropriate context information for interleaving messages as illustrated in Listing 1.1. Section 6.2 explains how this property is set using SAM.

Persons are represented by `foaf:human` as defined in the FOAF ontology, which already features messaging relations, including instant messaging properties such as `foaf:jabberID`.

In order to add semantics to messages and conversations, they are annotated with entries of a taxonomy. The taxonomy is defined using the Simple Knowledge Organization System (SKOS[3]), an ontology to describe concept schemes providing several predefined classes and properties for this purpose. The `skos:narrower` and `skos:broader` properties are used to build a `skos:Concept` hierarchy, while the `skos:subject` property is used to associate *things* - in our case messages and conversations - with concepts.

Employing a standard meta ontology for knowledge representation fosters integration of ontologies that are based on the same meta ontology. However, as the hierarchical structure is established by only two relations, namely *broader*

[3] http://www.w3.org/2004/02/skos/

and *narrower*, transforming existing taxonomies or lexica defined with other meta ontologies to a SKOS representation is straightforward as well. As an example, Wordnet[4] can be transformed to a concept hierarchy defined with SKOS by interpreting the hypernym and hyponym relations of Wordnet as narrower and broader relations of SKOS.

Provenance data for annotations that allows to track who annotated what and when is established by individuals of `m:AnnotionStatement` that references the creator (`m:annotator`) and creation date of an annotation. Any such annotation is a reified statement that points at the resources representing the annotation.

Provenance information is also kept for messages and taxonomy entries by the `m:sender`, and `m:conceptCreator` properties as illustrated in Fig. 1.

6 SAM

6.1 Technologies Enabling SAM

SAM builds upon the instant messaging client BuddySpace[5] [17], which was developed during research on online presence in instant messaging at Open University. BuddySpace is a client for the Jabber[6] network which we extended to use the ontology depicted in Sect. 5. A programming interface was developed that encapsulates the ontological model and provides methods to write to it and read from it, such as adding an annotation, or retrieving messages annotated with a given concept.

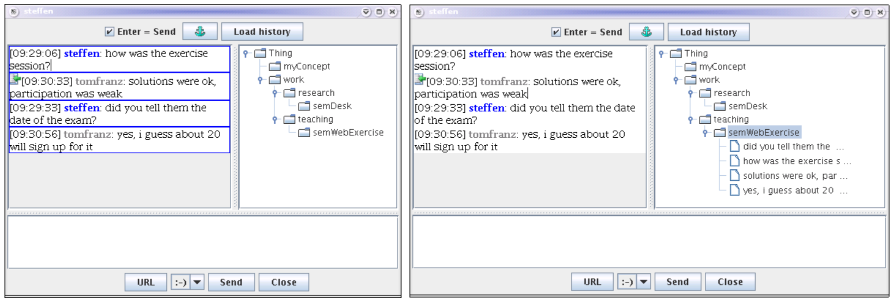
The messaging ontology is defined using the Web Ontology Language (OWL)[1]. It defines the properties and classes as explained in Sect. 5, including appropriate restrictions for them (range, domain, cardinality, functional, inverse, et cetera). Instances of the classes defined in the ontology are represented as RDF to support integration with the Networked Semantic Desktop and to establish a well structured and easy to access data store that simplifies incorporation of meta information, interlinking of resources, and exchange. The Jena[7] RDF API for Java is used to access the store.

The communication protocol used by the Jabber network is the Extensible Messaging and Presence Protocol (XMPP)[15], an XML-based protocol that is well supported by multiple open source programming libraries.

6.2 Annotations and Context

In contrast to common IM clients, the chat window of SAM contains an additional taxonomy panel (cf. Fig. 2). The chat window permits message annotation, taxonomy management, and the addition of context information while chatting. Both, the message panel and the taxonomy panel allow to accomplish multiple

[4] http://wordnet.princeton.edu
[5] http://kmi.open.ac.uk/projects/buddyspace/
[6] http://www.jabber.org
[7] http://jena.sourceforge.net/



(a) Selecting multiple messages. (b) Annotating with a taxonomy entry.

Fig. 2. Annotating Multiple Messages

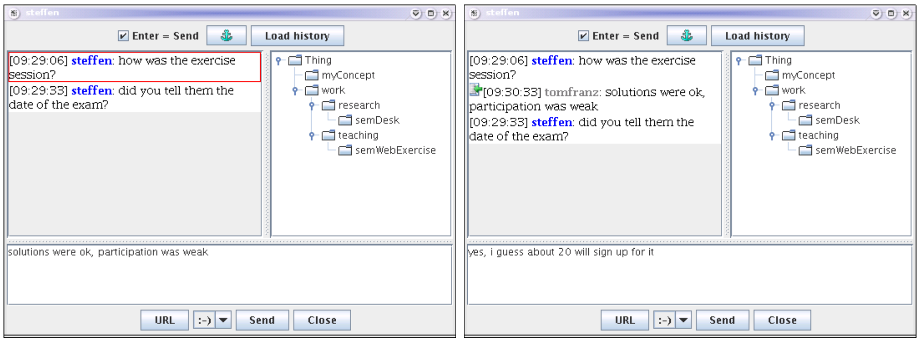
annotations at once to reduce user effort. Annotations are made either by double-clicking on a particular message that automatically annotates that message with all taxonomy entries that are currently selected, or by double-clicking a taxonomy entry which automatically annotates all selected messages with that entry as illustrated in Fig. 2. To further minimize user effort, if no message is selected, double-clicking on a concept contained in the taxonomy automatically annotates the last displayed message. As direct visual feedback, annotated messages are displayed as child nodes in the taxonomy (cf. Fig. 2b).

New annotations are automatically sent to the conversation partner to further reduce annotation effort and gain additional message semantics. We propose different policies (cf. Table 1) that define how new annotations that potentially introduce new taxonomy entries are handled based on how much trust is given to the creator of an incoming annotation.

Table 1. Policies for handling incoming annotations.

Trust Level	New Annotation	New Taxonomy Entry
Low	require user confirmation	require user confirmation
Medium	automatically add annotation	require user confirmation
High	automatically add annotation	automatically add entry

For any created message, the `m:follows`, `m:precedes`, `m:sender`, `m:recipient`, `m:hasText`, and `m:hasConversation` properties are automatically set by SAM to establish context information. The `m:repliesTo` property can be set through the message panel of the chat window as illustrated in Fig. 3: Selecting a message with a right-click automatically sets the `m:repliesTo` property of the next sent message to the selected one. Messages that have this property set are automatically displayed underneath the message they reply to. As IM conversations often have interleaving messages (cf. Listing 1.1) with different topics, this feature does not only provide additional message context, but also eases IM conversation as it assists the user in identifying related messages. All context information created for a message on one client is automatically transferred to the recipient when that message is sent to provide as much meta information as possible on both sides of a conversation. Section 6.4 describes in more detail how the transfer of such information is implemented.



(a) Selecting message to reply to. (b) After sending the message.

Fig. 3. Replying with interleaving messages

6.3 Semantic Search and Semantic Browsing

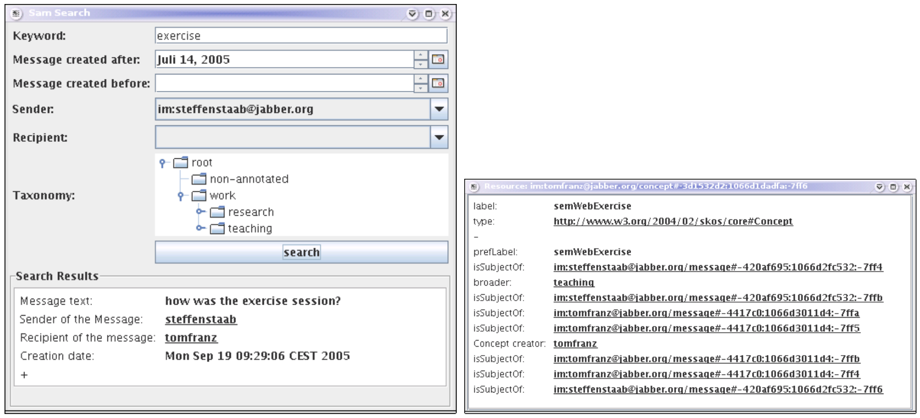
SAM allows to combine full-text search in message texts with semantic search features as illustrated in Fig. 4a. The user can restrict a search by specifying a date range for the message creation time, require specific persons to be the sender and the recipient, and restrict search results to be associated with certain taxonomy entries. Resulting messages are displayed with their context available for further exploration through the property explorer that opens by clicking on non-literal objects such as persons and taxonomy entries (cf. Fig. 4b).

The semantic browser (cf. Fig. 5) allows to view messages classified by the individual taxonomy. Non-annotated messages are associated with an additional taxonomy entry so that the user can still access them. As for search results, object properties (displayed underlined) can be further examined (cf. Fig. 4b).

6.4 Meta Data Transfer

Every messaging entity (e.g. person, message) is identified by its uniform resource identifier (URI) to support global identification and thus exchange of such entities. For example, each new instance of `m:Message` needs to be available for the sender and the recipient as both may want to reutilize it.

SAM exploits the extension mechanism of the XMPP to transfer messaging entities between different SAM clients, and to support automatic meta data exchange. Different extension types, namely *message*, *annotation*, *resourceRequest*,



(a) Semantic search. (b) Property Explorer.

Fig. 4. Semantic Search and Property Explorer

and *resourceResponse* are defined for this purpose. A *message* extension contains the RDF representation of a message while an *annotation* extension contains an instance of a `m:AnnotationStatement`. The two other extensions enable to request and retrieve one or multiple RDF resources with all their properties. The following two use cases exemplify how the extensions are used:

1. When sending a chat message, SAM automatically creates a new instance of `m:Message` with corresponding properties, and attaches its RDF representation in a *message* extension to the XMPP packet that sends the message. The receiving SAM client extracts the RDF data contained in the packet's extension and adds it to its own store.
2. When a client receives an annotation with a taxonomy entry that is not contained in its RDF store, the client repeatedly requests more general (`skos:broader`) taxonomy entries from the sender until a retrieved entry matches an entry in the local taxonomy so that the new taxonomy entry can be correctly inserted into the taxonomy and the annotation becomes effective.

7 Conclusion & Related Work

This paper presents concepts and an implementation of enhanced IM with respect to the Social/Networked Semantic Desktop. SAM introduces rich meta data, including semantics, to instant messaging and its content to provide enhanced management features that exploit such additional information. The main achievement, distinguishing SAM from existing systems, is the establishment of an IM infrastructure to globally exchange content and its semantic meta data in order to gain knowledge. This ability grounds several novel applications such as knowledge collaboration.

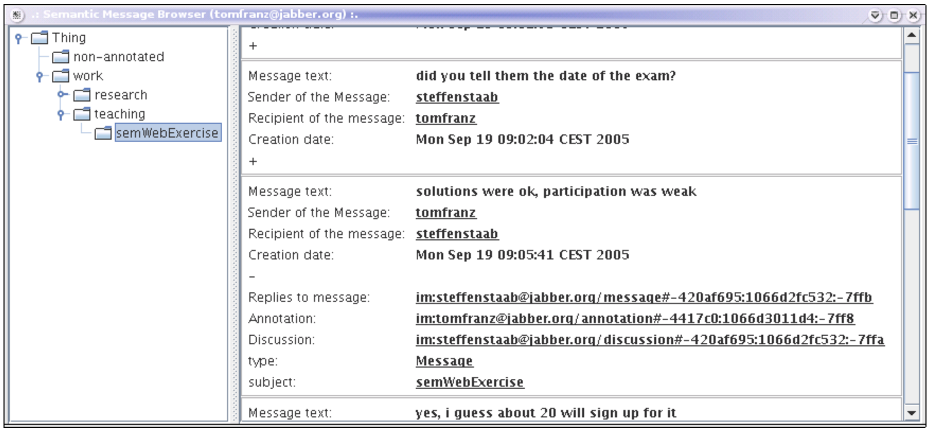


Fig. 5. The Semantic Browser of SAM

Zhang et al. present the *Small World Instant Messenger* in [18]. They build user profiles based on users' bookmarks or homepages, which are then used for expertise search. In contrast to our approach they rather exploit the infrastructure provided by instant messaging without addressing any issues of instant messaging itself. Consequently, they disregard management, reusability, and integration issues while establishing a service on top of instant messaging.

The Haystack system [14] comes with a general notion of messaging including a unified messaging ontology [13] similar to the ontology presented in Sect. 5. While the Haystack system focuses on integrating messaging into a personal information manager, SAM considers the networked exchange of meta information and is ready to integrate with other applications on the Semantic Desktop.

Chirita et al. explain how to use activity based semantic meta data [3] in their desktop search prototype. Exemplarily, they deal with email, file system, and web cache meta data and have developed an architecture that combines such meta data with standard full text search. While our work also combines full-text search and meta data to improve management, we address different enhancements and options for exploitation that are specific to the instant messaging context.

Vogiazou et al. established enhanced symbolic presence for instant messaging [17]. One outcome of this research is the BuddySpace instant messaging client and server component that allow to automatically group *buddies* and visualize their location and presence information respectively. SAM extends BuddySpace by semantic annotations, semantic search, semantic browsing, and (semantic) meta data communication.

The CoAKTinG (Collaborative Advanced Knowledge Technologies in the Grid) project [2] developed a meeting ontology to summarize content of different collaborative technologies. The summarized content is used to provide meeting replays that span content communicated via multiple channels, such as instant messaging, or video conferencing. While CoAKTinG imports BuddySpace communication logs into the meeting ontology, SAM contributes to CoAKTinG by providing already well structured additional (semantic) meta data.

The Gnowsis system [16] provides an architecture and server component for integrating arbitrary applications on the Semantic Desktop. Applications are required to describe their data by ontologies and are connected to the Gnowsis desktop service by plugins. As a result, different data from various desktop applications is unified through a single Gnowsis user interface. As SAM already employs ontologies to represent all its data, integration into the Gnowsis system is at hand.

8 Outlook

Meta data exchange as explained in Sect. 4.3 and 6.4 can enrich knowledge bases but also institutes several applications that go beyond that scope. Taxonomy overlappings between different communication partners represent a shared view, naturally established based on communication of taxonomy entries and their relations. Accordingly, rejecting and accepting incoming taxonomic data is a simplistic example of online collaboration on a concept hierarchy. Further work on generalizing the process model will allow online collaboration that is independent of a specific problem domain.

While IM is employed by business organizations, improving company wide knowledge management through expertise search might be a welcomed feature in businesses. The Bibster project [8] establishes semantic routing based on the expertise of peers. In Bibster, expertise is computed from annotations of bibliographic data with topics from the ACM topic hierarchy. The knowledge base provided by SAM can be exploited similarly, however, not to implement semantic routing but to compute the expertise of users and provide an expertise search.

As mentioned in Sect. 4, we consider automatic message classification as a future improvement. As instant messages differ from other text documents [9, 12], we consider classification of such messages as a challenging task. However, as SAM provides rich message context, any message is usually related to several other messages that may be exploited to improve classification. Moreover, if each SAM client runs a classifier that works on a potentially different knowledge base, we may investigate how to combine different classifiers and their results to improve overall classification quality.

A very significant open issue is how to incorporate security and privacy issues, especially trust as defined in [5] as credibility and reliability of resources.

Application oriented visions include the integration with existing software for the Semantic Desktop such as the Haystack or the Gnowsis systems.

Acknowledgments

We would like to thank Arup Malakar for his contributions to the development of SAM. This work is conducted with respect to the upcoming project Knowledge

Sharing and Reuse across Media (X-Media), funded by the Information Society Technologies (IST) programme of the 6th Framework Programme.

References

1. Grigoris Antoniou and Frank van Harmelen. Web Ontology Language: OWL. In *Handbook on Ontologies*, pages 67–92. 2004.
2. Michelle Bachler, Simon Buckingham Shum, Yun-Heh Chen-Burger, Jeff Dalton, David De Roure, Marc Eisenstadt, Jiri Komzak, Danius Michaelides, Kevin Page, Stephen Potter, Nigel Shadbolt, and Austin Tate. Collaboration in the Semantic Grid: a Basis For E-Learning. In *Grid Learning Services Workshop, 7th International Conference On Intelligent Tutoring Systems*, pages 1–12, 2004.
3. Paul-Alexandru Chirita, Rita Gavriloaie, Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Activity Based Metadata For Semantic Desktop Search. In *ESWC*, pages 439–454, 2005.
4. Stefan Decker and Martin R. Frank. The Networked Semantic Desktop. In *WWW Workshop On Application Design, Development and Implementation Issues in the Semantic Web*, 2004.
5. Jennifer Golbeck, Bijan Parsia, and James A. Hendler. Trust Networks On the Semantic Web. In *CIA*, pages 238–249, 2003.
6. Rebecca E. Grinter and Margery Eldridge. Y Do Tngrs Luv 2 Txt Msg? In *ECSCW*, pages 219–238, 2001.
7. Rebecca E. Grinter and Leysia Palen. Instant Messaging in Teen Life. In *CSCW*, pages 21–30, 2002.
8. Peter Haase, Jeen Broekstra, Marc Ehrig, Maarten Menken, Peter Mika, Mariusz Olko, Michal Plechawski, Pawel Pyszlak, Björn Schnizler, Ronny Siebes, Steffen Staab, and Christoph Tempich. Bibster - A Semantics-Based Bibliographic Peer-to-Peer System. In *International Semantic Web Conference*, pages 122–136, 2004.
9. Ellen Isaacs, Alan Walendowski, Steve Whittaker, Diane J. Schiano, and Candace A. Kamm. The Character, Functions, and Styles of Instant Messaging in the Workplace. In *CSCW*, pages 11–20, 2002.
10. David R. Karger, Karun Bakshi, David Huynh, Dennis Quan, and Vineet Sinha. Haystack: A General-Purpose Information Management Tool For End Users Based On Semistructured Data. In *CIDR*, pages 13–26, 2005.
11. Michael J. Muller, Mary Elizabeth Raven, Sandra L. Kogan, David R. Millen, and Kenneth Carey. Introducing Chat Into Business Organizations: Toward an Instant Messaging Maturity Model. In *GROUP*, pages 50–57, 2003.
12. Bonnie A. Nardi, Steve Whittaker, and Erin Bradner. Interaction and Outeraction: Instant Messaging in Action. In *CSCW*, pages 79–88, 2000.
13. D. Quan, K. Bakshi, and D. Karger. A Unified Abstraction For Messaging On the Semantic Web. In *The Twelfth International World Wide Web Conference*, 2003.
14. Dennis Quan, David Huynh, and David R. Karger. Haystack: A Platform For Authoring End User Semantic Web Applications. In *International Semantic Web Conference*, pages 738–753, 2003.
15. Ed. Saint-Andre. RFC 3920: Extensible messaging and presence protocol (XMPP). Technical report.
16. Leo Sauermann. The Gnowsis Semantic Desktop For Information Integration. In *WM 2005: Professional Knowledge Management*, pages 39–42, 2005.

17. Yanna Vogiazou, Marc Eisenstadt, Martin Dzbor, and JiríKomzak. From Buddyspace to CitiTag: Large-Scale Symbolic Presence For Community Building and Spontaneous Play. In *ACM SAC*, pages 1600–1606, 2005.
18. Jun Zhang and Marshall W. van Alstyne. SWIM: Fostering Social Network Based Information Search. In *CHI Extended Abstracts*, page 1568, 2004.