Abstract. While instant messaging (IM) became a mature communication means in business organizations over the last years, IM systems did not keep pace with this evolution. Communicated content is often stored insufficiently and hard to recall, integration into other desktop applications impossible. In this paper, we demonstrate how to overcome these shortcomings and in addition introduce novel IM features beyond the scope of today’s shortcomings. We employ an ontological meta model that introduces semantics to instant messaging data and establish semantic annotations of messages. We exploit the gained additional semantics to foster advanced management. Annotation effort is minimized by the simplicity of message tagging, annotation recommendations, and automatic annotation exchange between buddies. The Semantics Aware Messenger (SAM) is a prototypical implementation of the ideas presented here and establishes networked exchange of semantic meta data to reduce user effort and enhance ease of use by exploiting information distributed over multiple buddies. Accordingly, SAM introduces IM to the Semantic Web.

1 Introduction

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 [12, 9]. Accordingly, we consider the content communicated via this media as of increasing value that should be recallable and integrated into other desktop applications.

The objective of this paper is twofold: First, we tackle traceability and data management shortcomings of current IM. Second, we ground novel IM features by providing SAM, a platform for semantics aware instant messaging.

* This is the revised and extended version of [5]
SAM is based on an i) ontological meta model for instant messaging which supports ii) integration with other applications, and introduces iii) meta data and semantics for IM to enable iv) sophisticated reutilization of instant messaging data. We exploit knowledge work (semantic annotations) of buddies to reduce user effort and enhance usability by enabling networked exchange of semantic meta information between buddies, which simultaneously grounds novel application scenarios as envisioned by the Social Semantic Desktop [4].

In Sect. 2, we sketch a typical IM scenario from which we derive common use cases. We then confront current IM systems with these use cases to discuss their shortcomings in Sect. 3. Section 4 gives a detailed overview of the ontological meta model and discusses how we exploit the model, followed by Sect. 5 dealing with message annotation. Section 6 demonstrates the prototype SAM, which implements the concepts presented in this paper. We contrast our work with related work in Sect. 7, and address future research in Sect. 8.

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:

[Steffen]: how went yesterday's exercise session?
[Steffen]: did you tell them the date of the exam?
[Thomas]: the solutions were ok, the room was nearly too small
[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 European Semantic Web Conference.

[Steffen]: hi thomas, did you already start working on the paper for the eswc?
[Thomas]: yes, i put it into the svn under papers/2005/
[Steffen]: ok, what is going to change?
[Thomas]: describe an IM scenario to indicate current shortcomings, propose improvements, and demonstrate SAM in terms of the scenario

Listing 1.2. ESWC Paper

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

[Thomas]: any ideas for a suitable scenario?
[Steffen]: why don't use this conversation?
[Thomas]: right! its a sufficient example of a work related chat
[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-Media:

[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
[Bernhard]: thanks, i’ll contact him when i’ve read it

Listing 1.4. Project X-Media

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%).

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 the ESWC paper, and what exactly he stated about the content of that document. 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-Media, Steffen wants to compile all X-Media related content, including messages that deal with the project.

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 Messages could be classified for more than one category.
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 [8, 7, 13] therefore become meaningless without context. In Listing 1.1, Thomas said "the solutions were ok, the room was nearly too small". 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 other desktop applications.

(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.
Fig. 1. Domain and Range Diagram of the Ontology

4 Ontological Meta Model

4.1 Description of the Model

As depicted in Fig. 1, a conversation is modeled by the class Conversation, which relates to messages exchanged within a conversation by the hasMessage property. A message is a subclass of foaf:Document and is associated to its content by the hasText property. The follows property and its inverse, precedes, are both transitive properties that track the chronological order in which messages appear, while the 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 with SAM and how it is used to improve the user interface.

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 tags. To foster integration, we employ the Simple Knowledge Organization System (SKOS)[1], a standard meta ontology for knowledge representation. Using SKOS, we represent a tag as skos:Concept. Resources – in our case messages and conversations – are associated with concepts by the skos:subject property.

Provenance data for annotations that allows to track who annotated what and when is established by individuals of the class Annotation that references

---

2 Namespace of foaf: http://xmlns.com/foaf/0.1/
3 Namespace of skos: http://www.w3.org/2004/02/skos/core#
the creator (annotator) of the annotation, the annotated message or conversation (annotates), the concept associated with the resource (usedConcept), and the creation date of an annotation.

Provenance information is also kept for messages and concepts by the sender, and conceptCreator properties as illustrated in Fig. 1.

4.2 Exploiting the Model

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. Furthermore, linked context information enables direct access to related information.

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 concepts. 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 we provide 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 (FOAF⁴) 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 [15]. 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.

A common meta model also enables exchange. In Listing 1.4, Steffen tells Bernhard to send him, what Thomas told him. Meta data exchange as proposed by SAM allows to integrate messages sent between Thomas and Steffen into Bernhard’s data store so that Bernhard can utilize the browser and search interface of SAM to access these messages.

⁴ http://www.foaf-project.org/
5 Message Semantics

5.1 Individual Annotation

The first shortcoming mentioned in Sect. 3 denotes weak message classifications provided by current IM clients. SAM enables semantic message annotations utilizing user defined tags as additional classification. In our case, tags denote a topic, subject, or concept associated with a message.

Tagging recently became very popular with systems such as Flickr\(^5\) or del.icio.us\(^6\). It is an easy to comprehend means of adding semantics to arbitrary resources demanding relatively little user effort. While taxonomies and other hierarchical models for organizing concepts require generality and specificity relations between concepts, the model employed for tagging requires no such relations between concepts or tags, respectively. In dependence on taxonomies and reflecting the collaborative character of the systems mentioned before, the term \textit{folksonomy} is often used to denote this model.

Back to the use cases, Steffen might use tags like \textit{project work} and \textit{X-Media} to annotate any message related to project X-Media. Accomplishing the second use case at a later time then becomes as easy as browsing for all messages annotated with corresponding tags. Message classification also benefits search, as query recall can be enhanced by additionally considering message tags (cf. Sect. 6.3). In Sect. 6.2 we explain how the user annotations with SAM.

While message annotation obviously improves management, the additional user effort imposed by it is a significant drawback. This effort is lowered by automatic annotation exchange between conversation partners as detailed in Sect. 5.3 and by automatic tag recommendations (Sect. 5.2).

5.2 Tag Recommendations

Creating an appropriate tag for a message requires cognitive performance. We are seeking to assist and encourage the user to tag messages by offering tag recommendations for each chat message he receives or sends.

The first step of the recommendation generation is to retrieve the base forms for the words contained in the message text by consulting the WordNet lexical database \cite{wordnet}. One portion of the recommendations we provide are the hypernyms as contained in WordNet for each of these base forms. While such recommendations are based on lexical relations only, we complement them with recommendations based on user associations: The delicious website provides related tags for tags based on their co-occurrence, i.e. based on which tags users use together with other tags. Accordingly, we consult the del.icio.us website to retrieve related tags for each of the base forms we have identified before. Example 1 lists the retrieved hypernyms and related tags\(^7\) for a message text of Listing 1.1.

---

\(^5\) http://flickr.com
\(^6\) http://del.icio.us
\(^7\) Based on the data presented at del.icio.us in Nov. 2005
Example 1.
Message text: how went yesterdays exercise session?
Base forms: go, yesterday, exercise, session
Hypernyms (WordNet): time_unit, unit_of_time, labor, labour, toil, discussion, give-and-take, word
Related tags go: java, problems, foolord:links, reference, tutorial, howto, games, strategy, tips, game, kiseido
Related tags yesterday: no results
Related tags exercise: health, fitness, workout, reference, diet, nutrition, howto, music, food, clothing, facemask
Related tags session: no results

5.3 Joint Annotation

Besides the simplicity of message tagging and tag recommendations for messages, we are further easing message annotation by exploiting the annotations of conversation partners. All participants of a conversation deal with the same set of messages. As each user decides which tags to associate to a message, 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 enable automatic annotation exchange between peers. New annotations that have not been considered by the recipient of an annotation add further message semantics, or add semantics at all for not yet annotated messages. While the first case improves management, the latter case is especially important to reduce annotation effort for the user. Finally, new annotations may also introduce new tags which may benefit the recipient. While one might not always want to automatically integrate new annotations as they may introduce unwanted associations, SAM offers different options to deal with incoming annotations as explained in Sect. 6.2. Technical aspects of meta data transfer are mentioned in Sect. 6.4.

6 SAM

6.1 Technologies Enabling SAM

SAM builds upon the instant messaging client BuddySpace\(^8\) [19], which was developed during research on online presence in instant messaging at Open University. BuddySpace is a client for the Jabber\(^9\) network which we extended to use the ontology depicted in Sect. 4.1. 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.

\(^8\) http://kmi.open.ac.uk/projects/buddyspace/
\(^9\) http://www.jabber.org
The messaging ontology is defined using the Web Ontology Language (OWL)[2]. It defines the properties and classes as explained in Sect. 4.1, including appropriate restrictions for them (range, domain, cardinality, functional, inverse, etcetera). Instances of the classes defined in the ontology are represented as RDF to support integration with other applications, 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\textsuperscript{10} 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)[17], an XML-based protocol that is well supported by multiple open source programming libraries.

### 6.2 Annotating Messages with SAM

In contrast to common IM clients, the chat window of SAM contains additional input controls to add message annotations while chatting (cf. Fig. 2). Users can enter new or existing tags as free text, separated by comma. An autocompletion mechanism assists the user in using previously defined tags (cf. Fig. 2a). Additionally, SAM presents a list of suggested tags that are automatically computed for each message (cf. Sect. 5.2). To minimize user effort, multiple messages can

![Fig. 2. Tagging messages.](image)

(a) Typing the tags.  
(b) Viewing the tagged message.

be annotated with multiple concepts at once. If no message is selected, the last displayed message is annotated by default. As direct visual feedback, annotated messages are shown with their annotations (cf. Fig. 2b).

New annotations are automatically sent to the conversation partner to further reduce annotation effort and gain additional message semantics. We propose

\textsuperscript{10}http://jena.sourceforge.net/
different policies (cf. Table 1) that define how new annotations that potentially introduce new concepts are handled based on how much trust is given to the creator of an incoming annotation.

Table 1. Policies for handling incoming annotations.

<table>
<thead>
<tr>
<th>Trust Level</th>
<th>New Annotation</th>
<th>New Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>require user confirmation</td>
<td>require user confirmation</td>
</tr>
<tr>
<td>Medium</td>
<td>automatically add annotation</td>
<td>require user confirmation</td>
</tr>
<tr>
<td>High</td>
<td>automatically add annotation</td>
<td>automatically add entry</td>
</tr>
</tbody>
</table>

For any created message, the follows, precedes, sender, recipient, hasText, and hasConversation properties are automatically set by SAM to establish context information. The 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 repliesTo property of the next sent message to the selected one (Fig. 3a). Messages that have this property set are rendered differently and automatically displayed underneath the message they reply to (Fig. 3b) to improve ease of use. 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.

![Fig. 3](image-url)
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, and require specific persons to be the sender and the recipient. The text input field on top of the search panel combines full-text search and semantic search in an intuitive way. Search terms entered into this field are automatically matched with both, message texts and tag labels so that message semantics introduced by the tags automatically enrich search quality and recall. Resulting messages are displayed with their context available for further exploration through the property explorer that opens by clicking on non-literal (underlined) objects such as persons and concepts (cf. Fig. 4b).

![Semantic search](image1)
![Property Explorer](image2)

The semantic browser (cf. Fig. 5) allows to view messages based on their semantic annotations. The semantic browser follows the navigational model imposed by folksonomies. The tag pane on the left shows all available tags that can be used to restrict the currently displayed messages. The tag pane automatically hides tags that denote sets of messages disjoint with the currently selected set so that only sensible selections, i.e. those that restrict the set without resulting in empty sets, can be made. Currently selected tags are shown in the selected tags panel on the right. Clicking on a tag on the left panel adds the tag to the restricting tags on the right, displays messages associated with currently selected tags, and manipulates the left tag panel accordingly. By clicking on any tag on
the right panel, the selection is revoked. The displayed results can be further explored in the same way as the search results mentioned before.

(a) Selecting one Tag.  
(b) Selecting an additional Tag.

Fig. 5. The Semantic Browser of SAM

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 the class 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. The XMPP provides a special packet type named InfoQuery (IQ) to establish communications consisting of requests and responses. SAM implements an XMPP binding for the RDF query language SPARQL\(^\text{11}\) utilizing IQ packets so that any SAM client can send a query to another SAM client that executes the query on its local RDF graph and returns the results as RDF to the requesting client. The following two use cases exemplify how this functionality is exploited:

1. When sending a chat message, SAM automatically creates a new instance of the Message class 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. Whenever a user annotates a message, the annotation information is transferred to the communication partner as a query result using the SPARQL binding. If the repository of the recipient does not contain the used concept, the recipient sends a SPARQL query to the sender to retrieve the missing information.

\(^\text{11}\) http://www.w3.org/TR/rdf-sparql-query/
7 Related Work

This paper presents concepts and an implementation of enhanced IM. 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.

Zhang et al. present the Small World Instant Messenger in [20]. 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 NABU system [14] is a semantic store for XMPP data, implemented as a server plugin. While SAM introduces message annotations and meta data exchange, NABU focusses on establishing a centralized message store.

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 [19]. 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.

Haystack [16] and Gnowsis [18] are systems for personal information management focussing on the integration of multiple information sources on the desktop to enhance management of that data. SAM provides a meta model and RDF representation for its data so that integration with such systems is at hand.

8 Outlook

To minimize user effort and improve management of IM data, we consider automatic message classification as a future improvement. As instant messages exhibit several particularities compared to other text documents [9, 13], we consider classification of such messages as a challenging task. Rich semantic meta data as introduced by SAM may leverage IM classification. Given a classifier for instant messages, we can investigate how the results of several such classifiers can be combined to further improve classification recall and precision.

A very significant open issue is how to incorporate security and privacy issues, especially trust as defined in [6] as credibility and reliability of resources.
We also propose to investigate relations between tags, buddies, and messages as motivated in [10]. Such information can then be used to compute similarity between users to discover social networks, or enable expertise search et cetera.

Acknowledgments

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) 6th Framework Programme.

References