

# xOperator - An extensible Semantic Agent for Instant Messaging Networks

Jörg Unbehauen, Michael Martin, Sebastian Hellmann, Sebastian Dietzold, and Sören Auer

Universität Leipzig, Department of Computer Science  
Johannisgasse 26, D-04103 Leipzig, Germany,  
{unbehauen|martin|hellmann|dietzold|auer}@informatik.uni-leipzig.de

**Abstract.** Instant Messaging is in addition to Web and Email the most popular service on the Internet. With xOperator we demonstrate the implementation of a strategy which deeply integrates Instant Messaging networks with the Semantic Web. The xOperator concept is based on the idea of creating an overlay network of collaborative information agents on top of social IM networks. It can be queried using a controlled and easily extensible language based on AIML templates. Such a deep integration of semantic technologies and Instant Messaging bears a number of advantages and benefits for users when compared to the separated use of Semantic Web technologies and IM, the most important ones being context awareness as well as provenance and trust. Our demonstration showcases how the xOperator approach naturally facilitates enterprise and personal information management as well as access to large scale heterogeneous information sources.

## 1 Introduction

With estimated more than 1 Billion users<sup>1</sup> Instant Messaging (IM) is in addition to Web and Email the most popular service on the Internet. IM is used to maintain a list of close contacts (such as friends or co-workers), to synchronously communicate with those, exchange files or meet in groups for discussions. Examples of IM networks are ICQ, Skype, AIM or the Jabber protocol and network<sup>2</sup>. The latter is an open standard and the basis for many other IM networks such as Google Talk, Meebo and Gizmo.

While there are some proposals and first attempts to bring semantic technologies together with IM (e.g. [6, 4, 8]) in this paper we present a strategy and implementation called xOperator, which deeply integrates both realms in order to maximise benefits for prospective users.

The xOperator concept is based on the idea of additionally equipping an users' IM identity with a number of information sources this user owns or trusts (e.g. his FOAF profile, iCal calendar etc.). Thus the social IM network is overlaid

<sup>1</sup> [http://en.wikipedia.org/wiki/Instant\\_messaging#User\\_base](http://en.wikipedia.org/wiki/Instant_messaging#User_base)

<sup>2</sup> <http://www.jabber.org/>

with a network of trusted knowledge sources. An IM user can query this network of knowledge sources using a controlled (but easily extensible) language based on Artificial Intelligence Markup Language (AIML) templates[9]. The AIML component translates natural language into SPARQL queries according to pre-defined templates. In order to pass the generated machine interpretable queries to other xOperator agents of friends in the social IM network xOperator makes use of the standard message exchange mechanisms provided by the IM network. After evaluation of the query by the neighboring xOperator agents results are transferred back, filtered, aggregated and presented to the querying user.

Such a deep integration of semantic technologies and IM bears a number of advantages and benefits for users when compared to the separated use of Semantic Web technologies and IM. From our point of view the two most crucial ones are:

- **Context awareness.** Users are not required to world wide uniquely identify entities, when it is clear what/who is meant from the context of their social network neighborhood. When asked for the current whereabouts of Sebastian, for example, xOperator can easily identify which person in my social network has the name Sebastian and can answer my query without the need for further clarification.
- **Provenance and trust.** IM networks represent carefully balanced networks of trust. People only admit friends and colleagues to their contact list, who they trust seeing their online presence, not being bothered by SPAM and sharing contact details with. Overlaying such a social network with a network for semantic knowledge sharing and querying naturally solves many issues of provenance and trust.

This SWC challenge submission is structured as follows: after presenting the technical architecture of xOperator in Section 2 we exhibit some usage scenarios in Section 3. We report about how the xOperator fits into the Semantic Web Challenge in Section 4 and present the related work and a conclusion in Section 5 and 6. A first version of xOperator was already presented in ESWC 2008 [3], for the Semantic Web Challenge the software was significantly enhanced and extended (in particular with group and individual based access control mechanisms).

## 2 Technical Architecture

The xOperator agent is a mediator between the Jabber Instant Messaging network<sup>3</sup> on one side and the World Wide Web on the other side. xOperator is client in both networks. He communicates anonymously (or using configured authentication credentials) on the WWW by talking HTTP with Web servers.

---

<sup>3</sup> Our implementation currently supports the Jabber network, but can be easily extended to other IM networks such as Skype, ICQ or MSN Messenger

On the Jabber network xOperator utilizes the Extensible Messaging and Presence Protocol (XMPP, [7]) using the Jabber account information provided by its owner. Jabber clients only communicate with the XMPP server associated with the user account. Jabber user accounts have the same syntax as email addresses (e.g. `soerenauer@jabber.ccc.de`). The respective Jabber server cares about routing messages to the server associated with the target account or temporarily stores the message in case the target account is not online or its server is not reachable. Since 2004 XMPP is a standard of the Internet Engineering Task Force and is widely used by various services (e.g. Google Talk). Fig. 1 depicts the general technical architecture of xOperator.

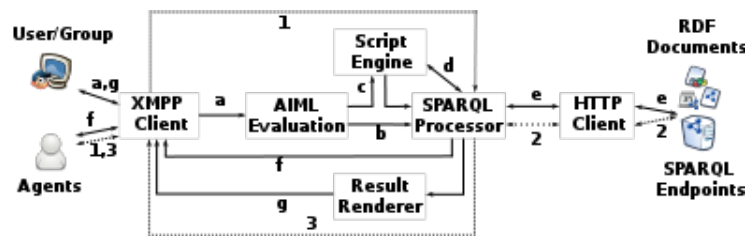


Fig. 1. Technical architecture of xOperator.

The agent works essentially in two operational modes:

1. *Answer natural language questions posed by a user* using SPARQL queries and respond to the user in natural language according to a predefined template (depicted in Fig. 1 by an uninterrupted line). Questions posed by a user (a) are either directly mapped to a SPARQL query template (b) or SPARQL queries are generated by a query script (c), which might obtain additional information by means of sub queries (d). The resulting SPARQL query will be evaluated on resources of the user (e), as well as passed on the Jabber network to neighboring agents for evaluation (f). All returned results are collected and prepared by a result renderer for presentation to the user (g).
2. *Receive SPARQL queries from neighboring agents* (depicted in Fig. 1 by an dotted line) (1) on the IM network, evaluate these queries (2) on the basis of locally known RDF documents and SPARQL endpoints and send answers as XML SPARQL Result Set [1] back via XMPP (3).

In both cases the agent evaluates SPARQL queries by querying a remote SPARQL endpoint via HTTP GET Request according to the SPARQL HTTP Bindings [2] or by retrieving an RDF document as well via HTTP and evaluating the query by means of a local SPARQL query processor.

### 3 Communication Scenarios

In this section we describe two usage scenarios of the xOperator agent. These examples demonstrate how the agent takes advantage of the social context and how processing is simplified by that. Figure 2 shows a schematic depiction of the communication scenarios. The figure is divided vertically into three layers.

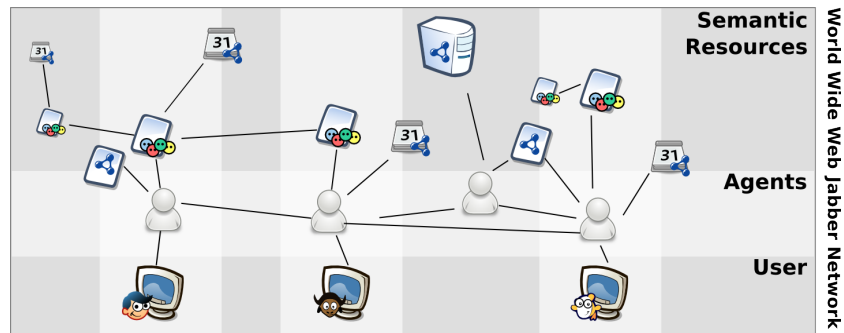


Fig. 2. Agent communication scenario

The upper layer represents the World Wide Web. Here mutually interlinked RDF documents (such as FOAF documents) reference each other using relations such as `rdf:seeAlso`. These RDF documents have been generated manually, exported from databases or were generated from other information sources. Tools used include for example the *bibtex2rdf*<sup>4</sup> application by Wolf Sibersky, the *iCal2rdf*<sup>5</sup> service provided by Dan Connolly and the *Facebook Foaf Generator*<sup>6</sup> by Matthew Rowe. In addition the SPARQL endpoint of *DBpedia*<sup>7</sup> and of *DBLP* is used.

The lower two layers in Figure 2 represent the Jabber Network. Here users are interacting synchronously with each other, as well as users with artificial agents (such as xOperator) and agents with each other. These communication flows follow the paths of the social overlay network. This network is defined by the friendships between the users and modeled in the buddy lists of the Jabber accounts.

A user can pose queries in natural language to an agent and the agent transforms the query into one or multiple SPARQL queries. Thus generated SPARQL queries can be forwarded either to a SPARQL endpoint or neighboring agents via the IM networks transport protocol (XMPP in the case of Jabber). Queries are forwarded to all neighboring agents, but not beyond to prevent flooding of the

<sup>4</sup> <http://www.l3s.de/siberski/bibtex2rdf/>

<sup>5</sup> <http://www.kanzaki.com/courier/ical2rdf>

<sup>6</sup> <http://www.dcs.shef.ac.uk/mrowe/foafgenerator.html>

<sup>7</sup> <http://dbpedia.org/>

network. SPARQL endpoints evaluate the query using a local knowledge base, dynamically load RDF documents from the Web or convert Web accessible information sources into RDF. Owners of these neighboring xOperator agents can employ xOperator's fine grained access control mechanisms based upon group and individual policies to allow or disallow querying of certain resources.

The results of SPARQL endpoints or other agents are collected, aggregated, filtered and presented to the user depending on the query as list, table or natural language response.

The different scenarios are presented in the remainder of this section and employ different uses of the communication network.

**Personal information querying** This scenario focuses on processing personal information, in our case data from calendars, address books and messaging information. The data is acquired by RDF mapping tools or manually created. Each user manages his data in his own xOperator instance, leaving every user to decide what to publish and what not. To enable the user to access the data we define a set of templates and describe the associated processing instructions. Each template is designed to match a certain type of question. We present some sample questions and describe the way they make use of the xOperator environment.

1. Tell me the birthday of Sebastian?
2. Where is Michael now?

An input like question 1 is matched by a template TELL ME \* OF \*. Stop word removal, as described in [3] results in the asterisks to match **birthday** and **Sebastian**. The template then will use the matches of the asterisks to query the FOAF ontology for attributes containing birthday. The property identified is then used for querying the neighboring stores.

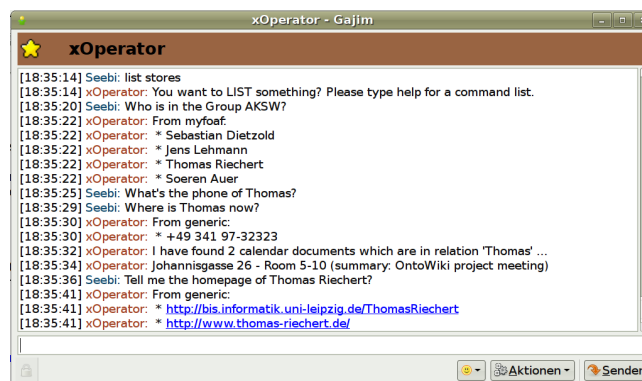


Fig. 3. Personal Information Querying scenario with an ordinary Jabber client

The same collaborative information acquisition is also used by the template matching question 2. It extracts the name of a person according to the pattern `WHERE IS * NOW` and exploits the association between the different information spaces FOAF and iCal in order to search for calendar information. iCal resources referenced in the FOAF profiles are filtered for data about current events. These are examined for presence related data to determine the location of **Michael**. Results of these queries are presented to the user as shown in Figure 3.

**Group agent querying** xOperator can as well be used for querying very large knowledge stores made available by means of a group agent. A group agent is independent of a particular user identity and can be queried by a possibly large group of (authorized) users. We demonstrate the principle with examples using DBpedia. Templates were created to query DBpedia and to allow domain independent information access. Sample question are:

1. **What is a jaguar?**
2. **What is a jaguar the feline?**

The pattern matching algorithms allocates question 1 with a template that extracts **jaguar** and queries DBpedia about concepts associated with jaguars. Having identified the concept of interest it is used in the template matching question 2 for a semantically supported information recovery about the **feline**. A public xOperator group agent with additional example queries is made available for the SWC evaluators at: <http://aksw.org/Projects/xOperator/SWC2008>.

## 4 xOperator in the Semantic Web Challenge

xOperator fulfills the requirements of the Semantic Web Challenge in the following ways:

The meaning of data plays a central role in xOperator:

- The meaning of data is represented using formal descriptions, these can be all syntactic representations, which can be easily transformed to RDF. E.g. xOperator handles iCal calendar, BibTeX and geo data. RDF is the primary data representation formalism in xOperator.
- Data is processed in interesting ways to derive useful information, in particular xOperator coherently integrates information from various data sources, data-specific end-user contributed query templates enable easy querying and group based access control mechanisms provide additional context boundaries for queries in a IM neighborhood.
- The information processing as done by xOperator plays a central role in achieving things that alternative technologies cannot do as well - especially xOperator's context awareness as well as the provenance and trust measures derived from overlaying the IM network with semantic information sources would be very complicated to achieve otherwise.

The information sources used are:

- under diverse ownership since each IM participant makes his dedicated information sources available to his contacts
- heterogeneous: (a) syntactically - since xOperator is syntax agnostic as long as there is some transformation to the RDF data model possible, (b) structurally xOperator does not have any specific limitations, and (c) semantically xOperator is able to deal with arbitrary vocabularies and information schemes,
- representing substantial quantities of real world data (e.g. DBpedia background knowledge, a large network of FOAF profiles or bibliographic data).

Finally, xOperator is an end-user application, i.e. it provides direct practical value to domain experts in various fields, such as members of virtual communities practice, enterprise users or leisure users. Furthermore, xOperator implements several of the additional desirable SWC features:

- It provides an attractive and functional interface for human users. xOperator is instantly usable for multi-million participants of the Jabber compatible IM networks, simply by adding existing xOperator agents to their contact list. Using IM clients implemented as Web applications (such as Meebo) xOperator is also directly usable on the Web (without the need for installing additional software).
- xOperator is novel, in the sense that it is the first application that comprehensively integrates IM networks and semantic knowledge representation.
- xOperator's functionality goes beyond pure information retrieval. E.g. users can integrate new knowledge sources, develop / share new query templates and define fine-grained access policies for members in their contact list.
- There is potential for commercial xOperator applications, such as integration of the technology into commercial IM networks or corporate IM applications such as IBM's Lotus Sametime and business models based on the provisioning of public xOperator instances.
- xOperator is accessible on a range of devices with IM clients - these are available for virtually every internet enabled consumer device including traditional computers as well as netbooks, mobile phones or PDAs.

## 5 Related Work

Proposals and first prototypes which are closely related to xOperator and inspired its development are Dan Brickley's JQbus<sup>8</sup> and Chris Schmidt's SPARQL over XMPP<sup>9</sup>. However, both works are limited to the pure transportation of SPARQL queries over XMPP. Quite different but the xOperator approach nicely complementing are works regarding the semantic annotation of IM messages [6, 4]. Finally, in [5] the author enhanced AIML bots by generating AIML categories from RDF models. Different to xOperator, these categories are static and represent only a fixed set of statements.

<sup>8</sup> <http://svn.foaf-project.org/foaftown/jqbus/intro.html>

<sup>9</sup> <http://crschmidt.net/semweb/sparqlxmpp/>

## 6 Conclusions and Future Work

With the xOperator concept and its implementation, we have shown how a deeply and synergistic coupling of Semantic Web technology and Instant Messaging networks can be achieved. The approach naturally combines the well-balanced trust and provenance characteristics of IM networks with semantic representations and query answering of the Semantic Web. The xOperator approach goes significantly beyond existing work which mainly focused either on the semantic annotation of IM messages or on using IM networks solely as transport layers for SPARQL queries. xOperator on the other hand overlays the IM network with a network of personal (and group) agents, which have access to knowledge bases and Web resources of their respective owners. The neighborhood of a user in the network can be easily queried by asking questions in a subset of natural language. By that xOperator resembles knowledge sharing and exchange in offline communities, such as a group of co-workers or friends. The xOperator example scenarios showcase how it naturally facilitates contacts and calendar management as well as access to large scale heterogeneous information sources. In addition to that, its extensible design allows a straightforward and effortless adoption to many other application scenarios such as, for example, sharing of experiment results in Biomedicine or sharing of account information in Customer Relationship Management. In addition to adopting xOperator to new domain application we view the xOperator architecture as a solid basis for further technological integration of IM networks and the Semantic Web. This could include adding light-weight reasoning capabilities to xOperator or the automatic creation of AIML categories by applying NLP techniques.

## References

1. Dave Beckett and Jeen Broekstra. SPARQL Query Results XML Format. W3c candidate recommendation, World Wide Web Consortium (W3C), April 2006.
2. Kendall Grant Clark. SPARQL Protocol for RDF. W3c recommendation, World Wide Web Consortium (W3C), 2007.
3. Sebastian Dietzold, Jörg Unbehauen, and Sören Auer. xOperator - Interconnecting the Semantic Web and Instant Messaging Networks. In *Proc. of 5th European Semantic Web Conference, ESWC 2008*, pages 19–33, 2008.
4. Thomas Franz and Steffen Staab. SAM: Semantics Aware Instant Messaging for the Networked Semantic Desktop. In *Semantic Desktop Workshop at ISWC*, 2005.
5. Eric Freese. Enhancing aiml bots using semantic web technologies. In *Proc. of Extreme Markup Languages*, 2007.
6. Frank Osterfeld, Malte Kiesel, and Sven Schwarz. Nabu - A Semantic Archive for XMPP Instant Messaging. In *Semantic Desktop Workshop at the ISWC*, 2005.
7. P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920, The Internet Engineering Task Force (IETF), October 2004.
8. Simon Buckingham Shum, David De Roure, Marc Eisenstadt, Nigel Shadbolt, and Austin Tate. CoAKTinG: Collaborative advanced knowledge technologies in the grid. In *Proc. of 2. Workshop on Adv. Collab. Env. at the HPDC-11*, 2002.
9. Richard Wallace. Artificial intelligence markup language (aiml). Working draft, A.L.I.C.E. AI Foundation, 2005. 18 February 2005.