

xOperator – An Extensible Semantic Agent for Instant Messaging Networks

Sebastian Dietzold¹, Jörg Unbehauen², and Sören Auer¹

¹ Universität Leipzig, Department of Computer Science
Johannisdgasse 26, D-04103 Leipzig, Germany
{dietzold,auer}@informatik.uni-leipzig.de

² Leuphana - University of Lneburg, Faculty III Environmental Sciences and
Engineering, Volgershall 1, D-21339 Lneburg
joerg@unbehauen.net

Abstract. Instant Messaging is in addition to Web and Email the most popular service on the Internet¹. 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 Background and Application Context

With estimated more than 500 Million users 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 network². The latter is an open standard and the basis for many other IM networks such as Google Talk, Meebo and Gizmo.

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 with a network of trusted knowledge sources. An IM user can query his local knowledge sources using a controlled (but easily extensible) language based on

¹ According to a sum up available at:

http://en.wikipedia.org/wiki/Instant_messaging#User_base

² <http://www.jabber.org/>

Artificial Intelligence Markup Language (AIML) templates[6]. The AIML component translates natural language into SPARQL queries according to predefined 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 neighbouring 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 neighbourhood. 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. Future versions of xOperator will allow more fine grained access control mechanisms, based upon group and individual policies.

2 Communication Scenarios

This section describes the three demonstrated agent communication settings for the xOperator system demonstration. Figure 1 shows a schematic depiction of the communication scenarios. The figure is divided vertically into four layers.

The first two layers represent the World Wide Web. Mutually interlinked RDF documents (such as FOAF documents) reference each other using relations such as `rdf:seeAlso`. These RDF documents could have been generated manually, exported from databases or could be generated from other information sources. For the system demonstration we use the Semantic Web Conference Corpus³ together with an instance of our semantic Wiki OntoWiki [1], a number of FOAF profile documents and private and public calendars⁴.

The lower two layers in Figure 1 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. 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

³ <http://data.semanticweb.org/>

⁴ We use Masahide Kanzaki's ical2rdf service for the live conversion of iCal calendars (e.g. from the Google's calendar service).

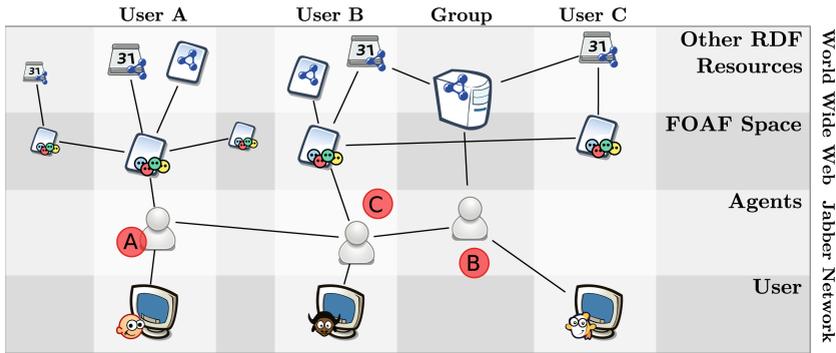


Fig. 1. Agent communication scenarios: (a) personal agent, (b) group agent, (c) agent network

a SPARQL endpoint or neighbouring 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 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. 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 communication scenarios of the demo are presented in the remainder of this section.

Personal Agent (A). A user of an Instant Messaging network installs his own personal agent and configures information sources he owns or trusts. Information sources can be for example a FOAF profile of the user containing personal information about the user and about relationships to other people he knows and where to find further information about these. Additionally the FOAF profile can link to other RDF documents which contain more information about the user and his activities (e.g. the users' iCal calendar). Such links span a network of information sources as depicted in Figure 1. Each user maintains his own information and links to information sources of his acquaintances. Depending on the query, the agent will access the respective resources. The following example queries are possible, when FOAF profiles are known to the agent: Give me the phone / homepage / ... of Frank! What is the birthday of Michael? Where is Dave now? Who knows Alex?

Group Agent (B). This communication scenario differs from the Personal Agent scenario in that multiple users gain access to the same agent. The agent does not only access remote documents but can also use triple stores for answering queries. For agents themselves, however, the distinction between RDF sources on the Web and information contained in a local triple store is not relevant. When used within a corporate setting this triple store can for example

contain a directory with information about employees or customers. A group agent accessing DBpedia on the other hand can enrich the xOperator query answering capabilities with background knowledge. This, e.g., enables queries such as: Which airports are easily reachable for members of my workgroup?

Agent Network (C). This scenario extends the two previous ones by allowing communication and interaction between agents. The rationale is to exploit the trust and provenance characteristics of the Instant Messaging network: Questions about or related to acquaintances in my network of trust can best be answered by their respective agents. Hence, agents should be able to talk to other agents on the IM network. A personal agent uses the IM account of its respective owner, has access to his contact list and is thus a part of its owner's social network. The agent is able to recognise other personal agents of acquaintances in the contact list and it is possible for agents to communicate without interfering with the communication of their owners. After other agents are identified it is possible to forward SPARQL queries (originating from a user question) to these agents, collect their answers and present them to the user. This enables, e.g., queries such as: "What is the next possible meeting date and place for members of my workgroup?" (based on the evaluation of workgroup members' calendars).

3 Demonstration Overview

The xOperator concept was implemented in Java and is available as open-source software together with an online demo from the AKSW group website⁵. Figure 2 shows the communication with the xOperator agent by means of an ordinary

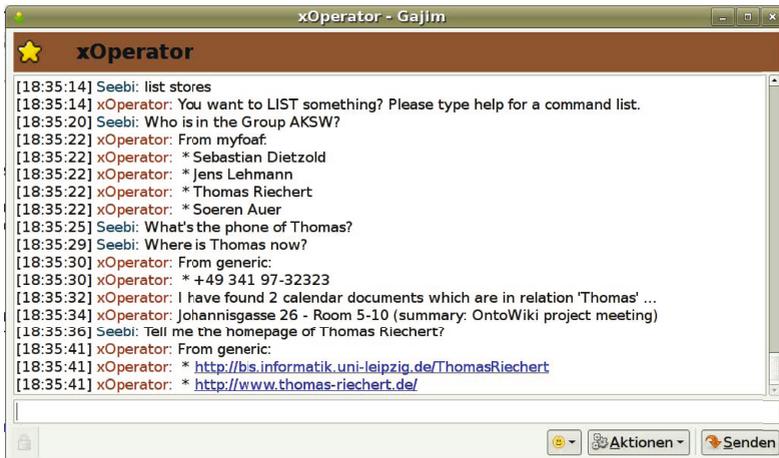


Fig. 2. Communication with xOperator by means of an ordinary Jabber client

⁵ <http://aksw.org/Projects/xOperator>

Jabber client, which facilitates the instant evaluation by interested users. In order to instantly try xOperator the AKSW maintains a publicly available xOperator instance with the xOperator account `xoperator-demo@aksw.org`, which can be simply added to a Jabber contact list. Example queries can be run on this demo instance, such as:

- `tell me (the) * of *` e.g. the phone of Sebastian
- `where is * now` e.g. where is Sebastian now

On the `help` xOperator will respond with information about the usage. Some data stores such as the DBpedia endpoint and the AKSW group OntoWiki are already pre-configured for the use within the xOperator demo. Additional ones can be easily added using the keywords `add ds`. The keyword `add template` allows xOperator users to add AIML templates on-the-fly. Parameters are the AIML pattern (using `*` as a placeholder) and a SPARQL query with `%%n%%`-references to the value of the n^{th} placeholder in the AIML pattern.

4 Related Work

Agent software is a rapidly developing area of research. In [4] the author proposed a typology of agents. According to this typology xOperator is an interface agent for the Semantic Web. Proposals and first prototypes which are closely related to xOperator and inspired its development are Dan Brickley's JQbus⁶ and Chris Schmidt's SPARQL over XMPP⁷. 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 [5,2]. Finally, in [3] 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.

References

1. Auer, S., Dietzold, S., Riechert, T.: OntoWiki - A Tool for Social, Semantic Collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006)
2. Franz, T., Staab, S.: SAM: Semantics Aware Instant Messaging for the Networked Semantic Desktop. In: Semantic Desktop Workshop at the ISWC (2005)
3. Freese, E.: Enhancing AIML Bots using Semantic Web Technologies. In: Proc. of Extreme Markup Languages (2007)
4. Nwana, H.S.: Software Agents: An Overview. Knowledge Engineering Review 11(3), 205–244 (1996)
5. Osterfeld, F., Kiesel, M., Schwarz, S.: Nabu - A Semantic Archive for XMPP Instant Messaging. In: Semantic Desktop Workshop at the ISWC (2005)
6. Wallace, R.: Artificial Intelligence Markup Language (AIML). Working draft, A.L.I.C.E. AI Foundation, February 18 (2005)

⁶ <http://svn.foaf-project.org/foaftown/jqbus/intro.html>

⁷ <http://crschmidt.net/semweb/sparqlxmpp/>