Utilising an Ontology Based Repository to Connect Web Miners and Application Agents

Stefan Haustein

University of Dortmund, Computer Science VIII,
Baroper Str. 301, D-44221 Dortmund, Germany,
stefan.haustein@udo.edu,
http://www-ai.cs.uni-dortmund.de

Abstract. Ontologies are important for providing a shared understanding of a domain for web mining agents and other agents accessing the gathered information. When the information access is decoupled from the mining process – for example when building a semantic web server – an additional storage compliant to the application ontology is needed. The COMRIS information layer was built to serve that purpose for a system supporting conference participants. It is able to provide permanent access to gathered or aggregated information suitable for both, humans and software agents by providing FIPA ACL and HTML interfaces.

1 Introduction

The goal of the COMRIS project was to design an agent based conference support system. Conference participants were equipped with a wearable electronic device that was able to recognise other participants wearing a similar device. The purpose of the device was to introduce participants to each other, to filter requests and to provide background information depending on the current context [1]. In order to perform its task, the agent controlling the device needed background information during the short period of time a certain context was valid: When a person has passed by, it is too late to introduce that person.

The background information should be provided by a web mining process. Since starting mining on demand seemed too slow for the given application, web mining was already performed beforehand. The approach is similar to using web spiders for search engines: If they were launched just when somebody enters a keyword, web searching would not be really practical.

2 The Mining Task

The gathering agents enrich the conference information by gathering information from different sources in the WWW. In our case, the agents just collect all information available about the registered conference participants, and new persons discovered in the gathering process were not investigated further. The information was used to enrich the knowledge about a person and its relations to other persons (e.g. co-author, project partner).

In the conference scenario, we were using three different types of gathering agents: The CORDIS collector is able to query the CORDIS project database of the European Union, the KA (Knowledge Acquisition) and ILP (Inductive Logic Programming) agents are able to query two different bibliography databases for the corresponding community. Each agent takes into account the special structure of its source, but they all stem from one generic agent. Together, the gathering agents are able to find European projects the conference participants were involved in and most of their publications.

Before the actual start of the conference, a learning step was applied to information gathered for a set of training persons. The Rule Discovery Tool (RDT [2]) of the MOBAL machine learning system [3, 4] was used to learn indicators for a "may-want-to-meet" relation between participants. While the learning step itself was performed off-line, the rules learned were applied to the information gathered in the runtime system, in order to create default instructions for the personal representation agents of the participants.

3 Complex Mining Tasks require Ontologies

When operating on a highly structured information space, it is no longer sufficient to just store words in a huge database. This is where the application ontology comes into play. Both, gathering and application agents need a common language. Also, in order to be able to perform the gathering beforehand, some kind of repository for the gathered information is needed. For the COMRIS conference scenario, the amount of concepts to be modelled, like participants, speakers, talks, sessions, rooms, agents, booths, schedules etc., became quite large. Moreover, all concepts had a lot of complex relations to other concepts.

Using relational tables for this purpose seemed inadequate because of the complicated mapping that is required to transform the ontology to a high number of tables. Also, the table solution seemed inflexible because ontological changes would cause a lot of changes in database tables and additional "agentification" wrappers.

Description Logic [5] systems like KL-ONE [6] provide additional features like automated classification that are computational expensive but not required in the system. All reasoning was intended to be performed by the specialised agents. Like for the relational tables, additional wrappers for an Agent Communication Language (ACL) would be required. Also, using Description Logics would require globally unique slot names, leading to additional negotiation efforts between the project partners designing "their" part of the application ontology.

OntoBroker, a system extracting ontologies from the web, is able to automatically unfold the stored knowledge and provides persistence for the ontology itself [7]. While its centralised structure would be a good starting point for learning mechanisms, the system interface is not agent but human oriented.

4 Information Layer System Architecture

For the given reasons, we decided to build a new kind of information system that

– provides ACL access in the first place,
– is agent based itself,
– and is built on an ontology that is not hard-wired to the system.

The main purpose of the system was to act as blackboard [8, 9] for decoupled communication between the conference organisers entering the initial participant information, the gathering agents annotating this information with web content, and the application agents utilising the gathered information for their tasks helping the conference participants.

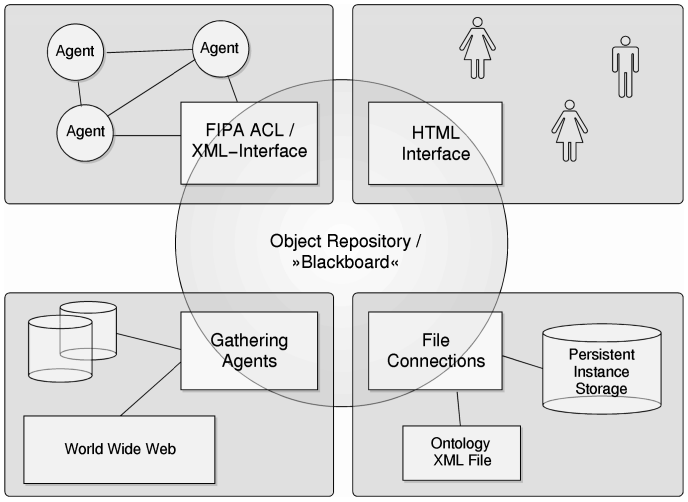

Fig. 1. Information layer architecture overview

The kernel of this system, the COMRIS information layer, provides only a memory representation of information structured corresponding to a given ontology. All other features were delegated to additional modules or agents, performing specialised tasks like:

– Handling communication with other agents
– Applying the learned rules to transform gathered data to default agent instructions
– Synchronisation with the underlying persistent data storage
– Building a generic HTML presentation from the ontology and the actual information layer content

The HTML presentation was not an initial part of the system, but once the system was built, it seemed a waste of resources to set up a separate conference web site built on traditional techniques. Instead, a wrapper agent transformed HTTP requests to ACL messages and forwarded them to the corresponding agents. Figure 1 shows an overview of the system architecture.

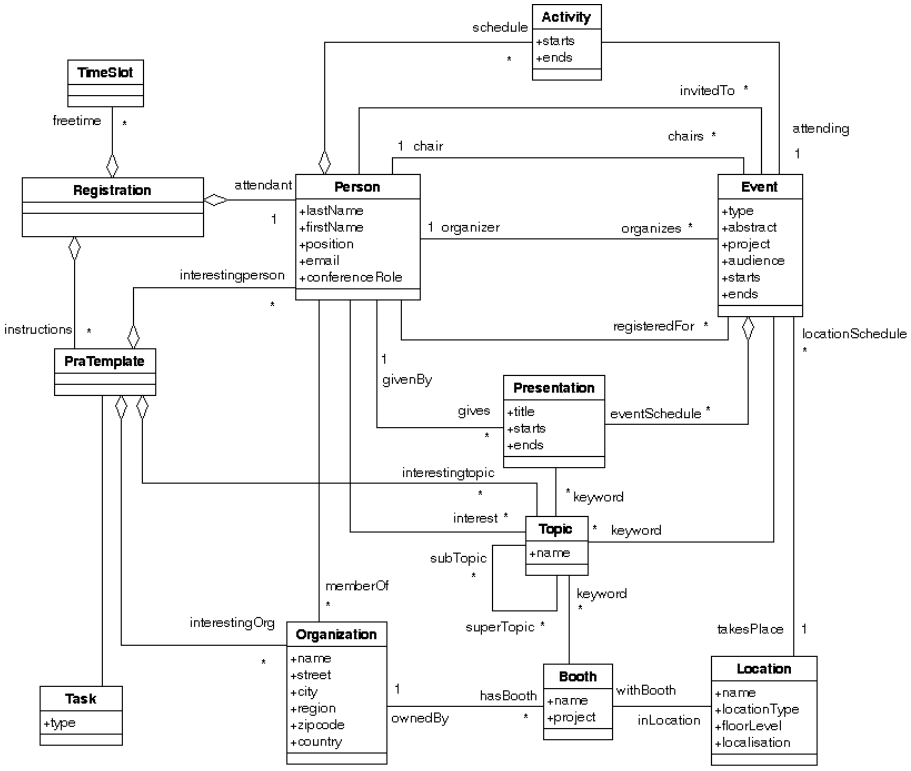

Fig. 2. Sample UML ontology diagram

5 Ontology and Data Model

The information layer uses an object-oriented model for data representation. Objects consist of atomic attributes and relations to other objects. The consistency of relations in both directions is ensured automatically, avoiding inconsistencies inside the system. The concepts and relations are defined application-dependent in an external ontology definition file. All files used by the information layer are stored as XML documents.

The ontology used in the COMRIS information layer is defined using an Unified Modelling Language (UML) model [10, 11] encoded in a simple XML format. Compared to other languages suitable for ontology modelling, UML currently still lacks clearly defined semantics. However, there are significant efforts to solve this problems [12, 13].

Figure 2 shows the UML diagram of the shared parts of the COMRIS ontology. Gathered information about publications and projects was transformed to templates for the Personal Representative Agents (`PraTemplate`) by applying the learned rules. The raw data gathered was also stored in the information system, but was not shared among all agents.

6 Communication and Content Languages

The communication and content languages for software agents and system components are based on XML, too. An XMLified version of FIPA ACL [14] is used as communication language, whereas the actual content language format is derived from the ontology automatically. Figure 6 shows the content language encoding of Tanja Katschenko and Carlos Gomez working at IBM corresponding to the ontology example in the previous section.

Linked Structure

```
<Organization id="555777">
   <name>IBM</name>
</Organization>

<Person id="888543">
   <name>Katschenko</name>
   <firstName>Tanja</firstName>
   <memberOf idref="555777"/>
</Person>

<Person id="878653">
   <name>Gomez</name>
   <firstName>Carlos</firstName>
   <memberOf idref="555777"/>
</Person>
```

Nested Structure

```
<Organization id="555777">
  <name>IBM</name>
  <members>
    <Person id="888543">
      <name>Katschenko</name>
      <firstName>Tanja</firstName>
    </Person>
    <Person id="878653">
      <name>Gomez</name>
      <firstName>Carlos</firstName>
    </Person>
  </members>
</Organization>
```

Fig. 3. Content language examples

Relations between instances can be described using the `idref` attribute, or by embedding related instances in the relation element. The encoding used for sending instances to software agents or other entities can be controlled by the corresponding entity to fit its particular needs best.

Readers familiar with the Resource Description Format (RDF) will have noticed a strong similarity of the formats. While it would be possible to migrate to RDF, there would be no improvement concerning human readability, which turned out crucial for system integration and maintenance. Moreover, RDF uses a property-centric data model, causing compatibility issues with traditional object oriented systems. The high number of RDF syntax variants leads to integration problems with other XML building blocks like XML Schema and XSLT [15] [16]. For those reasons, we will replace the current XML representation by the serialisation format of the Simple Object Access Protocol (SOAP) [17], improving the compactness and readability of the format as well as compatibility to SOAP based third party systems. However, migration to SOAP does not exclude building an additional RDF based interface if required.

7 Query Interface

The information layer supports a subset of OQL [18] as query language for agents. Additional languages may be plugged in by adding corresponding agents. By subscribing to the information layer, it is possible to keep an agent up to date without polling [19].

8 HTML Generation

The information layer contains a module that provides built-in web-server functionality. Since XML is not fully supported by web browsers yet, the server is able to generate HTML dynamically: For any object, the attributes are simply displayed, and the relations are converted to sets of hyperlinks to the related objects (figure 4). The HTML interface can also be used to edit the content of the system using forms generated dynamically based on the ontology.

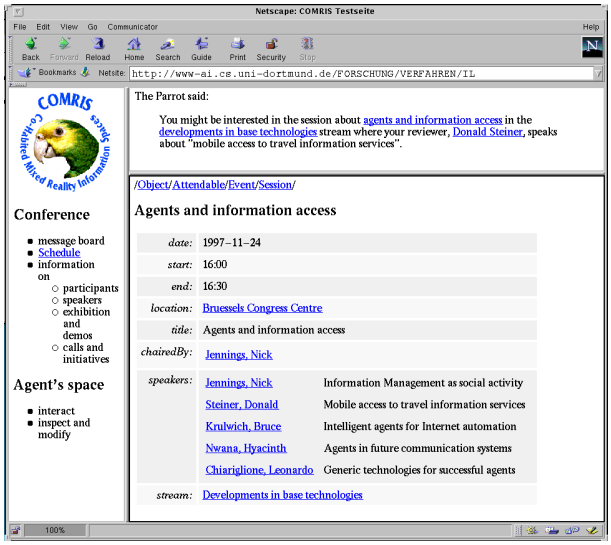

Fig. 4. Access to the Information Layer using a Web Browser.

In the COMRIS project, the HTML interface was used for interaction with the end user as well for as debugging and inspection purposes.

In addition to generic HTML generation, templates can be used in order to generate HTML pages conforming to a given look and feel. In the COMRIS project, we have also used the template mechanism to generate the input structure required by the text generation system TG/2 ([20]) which was used to generate natural language output for the wearable device.

The template mechanism was also used to generate questionnaires for evaluating the mining and learning results of the gatherers [21, 22].

9 Conclusion and Outlook

The main purpose of the implemented system was to provide an ontology based persistent blackboard communication mechanisms for connecting mining and application agents.

Using ontologies and agent technologies enabled a simple extension of the system beyond the original purpose. The system can now also be used to publish structured and massively linked data to the traditional “human readable” web using template based (X)HTML generation. The system proved useful not only for modelling some aspects of a conference but also for other applications with many sets of small and massively linked objects.

Currently, the COMRIS information layer is used for two internal projects and as the training server of MLnet[1]. In the future, it is planned to use the information layer in the MiningMart project for storing and editing data mining meta information.

The most important future developments are to make the information layer compliant to SOAP serialisation [17] and XMI in order to use a standardised XML formats for the message content language and for the ontology definition. It is also planned to include structure translation mechanisms for connecting systems using different but related application ontologies.

Acknowledgements

The research reported in this paper was supported by the ESPRIT LTR 25500 COMRIS project.

References

1. Plaza, E., Arcos, J.L., Noriega, P., Sierra, C.: Competing agents in agent-mediated institutions. Personal Technologies Journal **2** (1998) 1–9
2. Kietz, J.U., Wrobel, S.: Controlling the complexity of learning in logic through syntactic and task–oriented models. In Muggleton, S., ed.: Inductive Logic Programming. Number 38 in The A.P.I.C. Series. Academic Press, London [u.a.] (1992) 335–360
3. Sommer, E., Emde, W., Kietz, J.U., Wrobel, S.: Mobal 4.1b9 User Guide. GMD – German National Research Center for Information Technology, AI Research Division (I3.KI), St. Augustin, Germany (1996)
4. Morik, K., Wrobel, S., Kietz, J.U., Emde, W.: Knowledge Acquisition and Machine Learning - Theory, Methods, and Applications. Academic Press, London (1993)
5. Nebel, B.: Reasoning and Revision in Hybrid Representation Systems. Number 422 in Lecture Notes in Artificial Intelligence. Springer-Verlag (1990)
6. Brachman, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. Cognitive Science **9** (1985) 171–216
7. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology based access to distributed and semi-structured information. In Meersman, R., other, eds.: Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, Boston, 1999. Kluwer Academic Publisher, Boston (1999)
8. Hayes-Roth, B.: A blackboard architecture for control. Artificial Intelligence **26** (1985)
9. Kollingbaum, M., Heikkilae, T., McFarlane, D.: Persistent agents for manufacturing systems. In: AOIS 1999 Workshop at the Third International Conference on Autonomous Agents. (1999)

[1] http://www.mlnet.org/training

10. Object Management Group: OMG Unified Modeling Language Specification, version 1.3. http://www.omg.org/technology/documents/formal/unified_modeling_language.htm (2000)
11. Cranefield, S., Purvis, M.: Uml as an ontology modelling language. In: Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99. (1999)
12. Precise UML Group: The Precise UML Group home page. http://www.puml.org (2001)
13. Clark, T., Evans, A., Kent, S., Brodsky, S., Cook, S.: A feasibility study in rearchitecting UML as a family of languages using a precise OO meta-modeling approach. Report, Precise UML Group (2000) http://www.cs.york.ac.uk/puml/mml/mmf.pdf.
14. Foundation for Intelligent Physical Agents: FIPA web site (2001) http://www.fipa.org/specs/fipa00023/XC00023F.pdf.
15. World Wide Web Consortium: XSL Transformations (XSLT) version 1.0. http://www.w3.org/TR/xslt (1999)
16. Haustein, S.: Semantic Web languages: RDF vs. SOAP serialization. In: Proceedings of the Second International Workshop on the Semantic Web at WWW10. (2001) http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-40/haustein.pdf.
17. Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., Winer, D.: Simple Object Access Protocol (SOAP) 1.1. Note, World Wide Web Consortium (2000) http://www.w3.org/TR/2000/NOTE-SOAP-20000508.
18. Cattell, R.G.G., ed.: The Object Database Standard: ODMG 2.0. Morgan Kaufmann (1997)
19. Haustein, S., Lüdecke, S.: Towards Information Agent Interoperability. In Klusch, M., Kerschberg, L., eds.: Cooperative Information Agents IV – The Future of Information Agents in Cyberspace. Volume 1860 of LNCS., Boston, USA, Springer (2000) 208 – 219
20. Busemann, S.: A shallow formalism for defining personalized text. In: Workshop Professionelle Erstellung von Papier- und Online-Dokumenten at the 22nd Annual German Conference on Artificial Intelligence (KI-98), Bremen (1998)
21. Haustein, S.: Information environments for software agents. In Burgard, W., Christaller, T., Cremers, A.B., eds.: KI-99: Advances in Artificial Intelligence. Volume 1701 of LNAI., Bonn, Germany, Springer Verlag (1999) 295 – 298
22. Haustein, S., Lüdecke, S., Schwering, C.: The Knowledge Agency. In Sierra, C., Gini, M., Rosenschein, J.S., eds.: Proceedings of the Forth International Conference on Autonomous Agents, Barcelona, Spain, ACM SIGART, ACM Press, New York (2000) 205 – 206