Information Environments for Software Agents

Stefan Haustein

University of Dortmund, Artificial Intelligence Unit
haustein@ls8.cs.uni-dortmund.de

Abstract. For biological agents, the perceived world does not consist of other agents only. Also software agents can have a shared world model they operate on. In this paper we discuss information systems as an important part of software agents' environment and try to identify the corresponding demands, leading us to build an information system for software agents on top of a conventional database. Thus we were able to add "agentified" access to the advantages of current information systems like reliability and speed.

1 Introduction

The COMRIS project aims to develop, demonstrate and experimentally evaluate a scalable approach to integrating the inhabited information spaces schema with a concept of software agents. The COMRIS vision of co-habited mixed-reality information spaces emphasizes the co-habitation of software and human agents in a pair of closely coupled spaces, a virtual and a real one. The COMRIS project uses a conference center as the thematic space and concrete context of work. Each participant wears his personal assistant, an electronic badge and ear-phone device, wirelessly hooked into an Intranet. This personal assistant – the COMRIS parrot – realizes a bidirectional link between the real and the virtual space. It observes what is going on around its host (whereabouts, activities, other people around), and it informs its host about potentially useful encounters, ongoing demonstrations that may be worthwhile attending, and so on. This information is gathered by several personal representatives, the software agents that participate in the virtual conference on behalf of a real person. Each of these has the purpose to represent and further a particular interest or objective of the real participant, including those interests that this participant is not explicitly attending to.

In the COMRIS and similar agent scenarios, information systems can take over two important roles: Firstly, information systems can serve as the world abstraction for software agents. Thus a virtual environment that does not consist of other agents only can be created: Passive objects like the COMRIS conference schedule and other general conference information are not modeled as agents but stored in an agentified information system. Using an web interface, also human users can access this information. Secondly, information systems can serve as a "blackboard" [HR85] [KHM99] system for software agents. COMRIS information gathering agents are helping the personal representative agents (PRA) [PANS98] improving the matchmaking process between participants. These two types of

agents do not communicate directly: In order to become more independent from the current access characteristics of the world wide web the gatherers are already collecting information about the known participants before the conference starts. The results are stored persistently in the information system and are available immediately when demanded by PRAs. The persistent storage capabilities can also be used to save the state of an agents easily.

2 Existing Solutions

The first solution we considered was a set of tables in a conventional relational database. Normalization [Ull88] provides a more or less clear and unambiguous way to come from the conference ontology to a set of tables. The first problem we encountered was that several concepts of our domain ontology are inherited from other concepts. Inheritance of concepts suggests that an Object–Oriented Database Management System (OODBMS) could be a more suitable solution. Also with introduction of sets as table fields [Wag98], it is possible to encode relations to sets containing references to related objects instead of table joins, which overcomes a general performance problem in traditional databases [LP83] The disadvantage of having more possibilities to model relations is that all persons involved in modeling would not only have to agree on a domain ontology, but also on the concrete implementation in an object–oriented database. Already [The90] explicitly addresses a similar point: "Performance indicators have almost nothing to do with data models and must not appear in them".

The Object Database Management Group (ODMG) standard [Cat97] seems clearer in this point by removing SQL concepts like tables and rows and replacing them by classes and instances, but ODMG carries around its own baggage from its persistent object storage roots. The conceptual overload raises not only negotiation problems but also makes it more difficult to provide agents with domain independent capabilities like general reasoning or learning. Additionally, also knowledge about the "implementation" of the ontology into a database schema would have to be maintained somehow[1].

Frame systems [Min85] [BS85] are the initial root of object oriented programming and are theoretically also able to serve as an information system for our agents. A very relevant example for a frame system is Ontobroker [DEFS99]. Ontobroker is an ontology–based tool for accessing distributed and semi-structured information developed at the AIFB institute of the University of Karlsruhe. Ontobroker stores knowledge in a central frame system and provides a query interface to the stored data. Data collection from the web is delegated to wrappers, accessing external databases, and the "Ontocrawler". The "Ontocrawler" is a program that is able to extract information from specially annotated web pages. Our problem – not only concerning Ontobroker but frame systems in general – is that we are performing all reasoning in the agents, and therefore we do not need advanced reasoning and classification capabilities in the information

[1] A good example that it makes sense to reduce conceptional overload may be the JAVA programming language compared to C++

system, mixing up the storage and application layer. In frame systems we have a better logical foundation than in OODBMS, but we are not willing to trade off performance.

3 The COMRIS Information Layer

Because of the raised problems, we did not want to use an existing information system without any adaption. So we decided to build our information system on top of a conventional database. The additions we made are an agent communication interface and the introduction of relations as first class members of the data model: Thus the characteristics of relations can be specified directly like in class diagrams of the Unified Modeling Language (UML) [FS97] instead of explicitly using container classes like the ODMG standard. The corresponding subset of UML class diagrams may be used to specify the ontology and thus also the data model. So the modeling process is able to concentrate on the "real work" instead of problems like choosing the "right" data structures etc.

For the agent communication, a direct mapping from the ontology to an XML encoding is used, embedded in an XML encoded FIPA ACL frame [Fou97]. XML can be parsed easily in several programming languages. Using style sheets, XML can be displayed by any web browser in the near future, enabling the software agents to send messages to "human agents" directly. An XML structure can be extended easily by adding XML-attributes without changing the general XML element structure, thus enabling agents using probabilistic data models to communicate with non-probabilistic services without the need of two completely different content languages.

Utilizing the simplicity of the data model, we were able to develop an instance browser and a web interface for the COMRIS information layer without great effort. The instance browser can be used to inspect and modify the whole content of the information system avoiding possibilities of syntactical errors completely. It also provides the advantages of a graphical user interface like selection lists for relations etc. The web interface of the information system can be used to make information like the conference schedule etc. accessible for human users directly.

4 Conclusion

With an agentified interface, information systems can play an important role in the environment of software agents. Beneath being a part of the perceived world of the agents, information systems can provide services like persistent storage and blackboard functionality that helps agent designers to focus on their main objectives and design agents as lean and powerful entities. In multi–agent systems, where agents are designed by different vendors, a simple data model directly derived from the domain ontology can reduce negotiation demands.

A more detailed technical documentation of the COMRIS information layer is available at:

http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/IL

Acknowledgements

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

References

[BS85] R. J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Cat97] R. G. G. Cattell, editor. *The Object Database Standard: ODMG 2.0.* Morgan Kaufmann, 1997.

[DEFS99] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. *Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, Boston, 1999*, chapter Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. Kluwer Academic Publisher, Boston, 1999.

[Fou97] Foundation for Intelligent Physical Agents. Agent communication language, 1997. http://www.fipa.org/spec/f8a22.zip.

[FS97] M. Fowler and K. Scott. *UML Distilled. Applying the Standard Object Modelling Language.* Adison-Wesley-Longman, 1997.

[HR85] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3), 1985.

[KHM99] Martin Kollingbaum, Tapio Heikkilae, and Duncan McFarlane. Persistent agents for manufacturing systems. In *AOIS 1999 Workshop at the Third International Conference on Autonomous Agents*, 1999.

[LP83] R. Lorie and W. Plouffe. Complex objects and their use in design transactions. In *Proc. ACM SIGMOD Database Week – Enginieering Design Applications*, May 1983.

[Min85] Marvin Minsky. A framwork for representing knowledge. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, chapter III, pages 245–263. Morgan Kaufman, Los Altos, CA, 1985.

[PANS98] Enric Plaza, Josep Llus Arcos, Pablo Noriega, and Carles Sierra. Competing agents in agent-mediated institutions. *Personal Technologies Journal*, 2(3):1–9, 1998.

[The90] The Committee for Advanced DBMS Function. Third-generation data base system manifesto. Technical report, University of California, Berkeley, April 1990.

[Ull88] Jeffrey D. Ullman. *Principles of database and knowledge–base systems*, volume 1. Computer Science Press, Rockville, MD, 1988.

[Wag98] Gerd Wagner. *Foundations of Knowledge Systems with Applications to Databases and Agents.* Kluwer Academic Publisher, 1998.