

Combination of Agent- and Blackboard-Technologies for Business Applications

Stefan Haustein, Sascha Lüdecke
University of Dortmund, AI Unit
{haustein, luedecke}@ls8.cs.uni-dortmund.de

June 22, 1999

Abstract

Business applications create new demands to agent systems: Reliability and extensibility become new challenges beyond a general proof of concept. In this article we show how a combination of blackboard and agent techniques can help meeting these requirements: The COMRIS agent architecture uses an object oriented information system as shared blackboard for agents, simplifying agent communication by far. Additionally, other general advantages of information systems are provided to software agents using an agentified access interface.

1 Introduction

In contrast to pure academic environments, business applications raise demands to agent systems beyond a general proof of concept. Agent systems offer several advantages like scalability and distribution. For business systems, these advantages have to be merged with additional requirements of business applications. In this article we show how combining blackboard [HR85] and agent techniques can help meeting these criteria. We present the COMRIS system that already implements a general blackboard functionality for agent systems and we show the possibility of using the existing system for business applications.

Some of the most important requirements to agent systems for business applications are:

1. *Reliability:* In contrast to pure research applications, in real business applications system faults may have intolerable consequences like loss of huge amounts of money, avoidable usage of resources, or additional environment wasting. The system should be robust against at least short time failures.

2. *Adaptability and extensibility*: “One fits all” is not true for business applications. The system should be adaptable to the special needs of certain industries or companies.
3. *Simplicity, transparency and easy maintenance*: If it is not transparent what the system does, it will not be trusted and thus not be used. It should be possible to inspect and to manipulate the system easily, e.g. when unpredicted situations are leading to wrong system behavior.
4. *Integration*: Real business applications will have to be integrated into existing software environments and to cooperate with existing legacy systems.

In order to meet these requirements, the combination of an agent system with an object oriented information system as shared blackboard offers several advantages. Agents can use the blackboard for simplified indirect communication instead of needing direct communication links between agents. Furthermore, agents can also make use of the general advantages an information system provides. The following list shows how the application can profit from this architecture corresponding to the identified demands to business agent systems:

1. *Reliability*: By backing up major planning steps to the information system, the system becomes safe against short period failure. Unpredicted conditions leading to system failure may be identified and corrected by inspecting and manipulating the content of the blackboard. Agents can use the blackboard to publish usable but not optimal results in order to meet anytime criterions.
2. *Adaptability and Extensibility*: When direct inter-agent communication is replaced by indirect communication over the blackboard, integrating new agents into the system becomes much easier: Details about services and interfaces of the different agents are not needed to be known, the minimum requirement is only the blackboard communication.

In the minimum case, only the ontology and the blackboard-interface need to be defined. The agents do not need the capability to communicate to all other agents directly. Effort in negotiating interfaces between different vendors is minimized.

3. *Simplicity and transparency*: The overall state of the system can be observed much easier by inspecting the blackboard than having to looking at the last n messages, especially if the blackboard provides a suitable user interface (fig. 1).
4. *Integration*: Wrappers to other applications can use the blackboard as destination for converted information and source for updates. So, in contrast to direct communication between wrappers and agents, development of wrappers is more independent from the concrete agents.

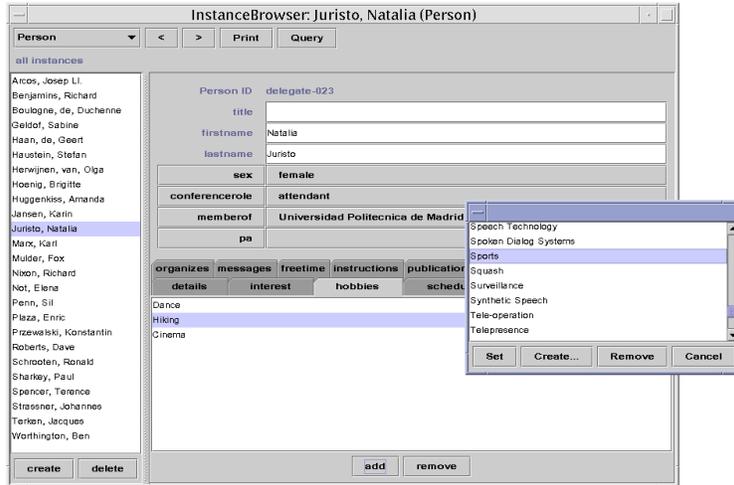


Figure 1: Instance Browser of the COMRIS information layer

2 The COMRIS Agent System

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 personal representation agents (PRA) [PANS98] 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.

2.1 Architecture

The COMRIS architecture is layered system: The uppermost layer is a set of operational agents. These agents are personal assistants (PA) and personal representative agents (PRA) [PANS98]. Beneath his PA, each participants owns a set of PRA, representing and defending the different interests of its owner.

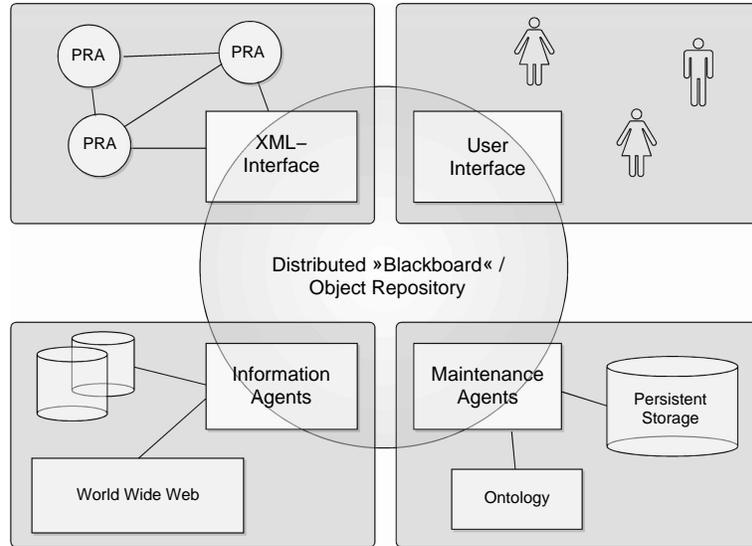


Figure 2: The architecture of the COMRIS agent systems

Some service modules that are addressed by the agents directly, like the language generation module, also reside in that layer.

The layer below the agents is the information layer, an information system that provides several services for the agents: It consists of an user interface for editing control information without having to care about machine readable format. Beneath their initialization and control information agents can store further information persistently in the information system or use it as blackboard for indirect information interchange with other agents.

The information layer itself can also be divided into two layers: Below the informations system are simple agents performing service tasks in order to enrich the content of the information system with gathered information from different sources. Thus, information gathering is decoupled from the operational agents in order to avoid redundant queries and to become more independent from the access characteristics of the sources.

The information system itself is built on top of a conventional database. It contains an agent communication interface and introduces relations as first class members of the data model: Thus the properties 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 [BPSM98] 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.

2.2 Sample Scenario

In an example scenario of COMRIS, a conference participant registers for the conference using the web interface of the information layer. The registration process automatically launches the PA of the participant. At the same time, a gathering agent tries to scan information about the participant in the Internet.

When the participant activates his interest in finding persons for forming a project consortium later, the information about other relevant participants is already available and does not need to be searched in the web. Redundant queries are avoided. The PRAs are able to negotiate the interests among each other and inform the corresponding PA about a wished meeting. If the message passes the competence and relevance filter of the PA, the PA sends the message to his participant via the parrot device. The language generation uses the information layer to enrich the meeting suggestion with further context information not contained in the original message.

If the meeting is actually confirmed, it is stored in the information system persistently and will survive power failures or similar problems crashing the agents. The participant can use one of the computers distributed over the conference center and the HTML interface of the information layer in order to get an overview of his conference schedule.

3 Sample Business Application

A sample business application for the COMRIS architecture could be modeling a production chain. Process optimizations are performed by reordering production steps in order to save conversion costs.

The units corresponding to the COMRIS PRAs are planning agents that are able to optimize plans in respect to one step of the production chain. Starting

point may be a new production task that is simply appended to the production plan. The agents then create alternatives optimized for their step of the production chain. The revised plans are written to the blackboard for “discussion”. If an agent finds a plan created by another agent, it adds the corresponding costs of its own step in the plan, if possible. So even locally optimized plans violating restrictions of other steps are eliminated.

The equivalents to the COMRIS information agents can collect subcontractor information instead of information about conference participants. The gathered information is stored in the information system uniformly readied corresponding to the ontology.

Maintenance agents look after the stored information in the background. They derive information using deduction rules stored in the ontology. Thus the time consumption and overall costs for a certain production plan are added automatically.

Since all plans are stored in the Information System, the user is able to inspect and manipulate the current state at any time he wishes.

4 Conclusion and Related Systems

The COMRIS-System is able to link external an existing systems and allows fast crash recovery. It provides transparency, extensibility, persistency, and easy access.

Programming agents is simplified by indirect communication over the blackboard. Only the blackboard interface needs to be provided, direct inter-agent interfaces are optional. New agents can be added easily to the system, and the operation of agents can be simply observed by monitoring the blackboard.

A disadvantage of the presented architecture is that the system depends on a central unit that can become a bottleneck.

Usage of information systems as blackboard is also suggested in [KHM99], but only in a planning state. The COMRIS information layer is already fully operational. An existing agent system for the sketched application field is MAPS [WD98], where the agents are communicating directly in contrast to our approach. The system also has some similarities to other systems that are implementing some aspects of the COMRIS information layer, e.g. object oriented information systems like CONCEPTBASE [JGJ⁺95] or OntoBroker [DEFS99].

Acknowledgments

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

References

- [BPSM98] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen, editors. *Extensible Markup Language (XML) 1.0*. World Wide Web Consortium, 1998. <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [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*. Addison-Wesley-Longman, 1997.
- [Gru93] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In Nicola Guarino and Roberto Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.
- [HR85] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3), 1985.
- [JGJ⁺95] M. Jarke, R. Gellersdörfer, M.A. Jeusfeld, M. Staudt, and S. Eherer. Conceptbase – a deductive object base for meta data management. *Journal of Intelligent Information Systems*, 4(2):167–192, 1995. Special Issue on Advances in Deductive Object-Oriented Databases.
- [KHM99] Martin Kollingbaum, Tapio Heikkilä, and Duncan McFarlane. Persistent agents for manufacturing systems. In *AOIS 1999 Workshop at the Third International Conference on Autonomous Agents*, 1999.
- [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.
- [WD98] J. Wellner and W. Dilger. Maps - a multi-agent production planning system. In A. Holsten, G. Joeris, C. Klauck, M. Klusch, H.-J. Müller, and J.P. Müller, editors, *Intelligent Agents in Information and Process Management, Workshop at the 22nd German Annual Conference on Artificial Intelligence (KI98)*, pages 71–78, Bremen, 1998. TZI-Bericht Nr. 9, Universität Bremen.