AN AGENT-BASED APPROACH TO DAM MONITORING

Ingo Mittrup
Institute of Computational Engineering, Ruhr-Universität Bochum
mittrup@inf.bi.rub.de

Kay Smarsly
Institute of Computational Engineering, Ruhr-Universität Bochum
kay.smarsly@rub.de

Dietrich Hartmann
Institute of Computational Engineering, Ruhr-Universität Bochum
hartus@inf.bi.rub.de

Volker Bettzieche
Ruhrverband (Ruhr River Association), Essen
vbe@ruhrverband.de

SUMMARY

Software agents - autonomous, mobile and intelligent software programs - provide all the necessary characteristics to innovate and accelerate the development of distributed applications. They represent powerful and robust software technology for implementing distributed collaborative work flows and complex interaction.
Applying software agents, the Institute of Computational Engineering, in cooperation with the Ruhrverband (Ruhr River Association) is taking an innovative approach to develop a modern dam monitoring system, which is capable of supporting the collaborative work of experts involved in monitoring. While the conventional computer-based monitoring systems consider the remote monitoring, the presentation and the electronic transfer of measured data, the present agent-based approach is focusing the distributed work flow of data analysis and safety assessment. Consequently, the complete work flow of dam monitoring is mapped onto a multi-agent system: regularly performed tasks (e.g. measuring at the dam) are carried out by task-oriented agents. In addition, the involved human experts are assisted by personal agents, which allow a direct communication with the multi-agent system and provide access to specific tasks. Associated with the known advantages of the agent technology, such as robustness and mobility, the introduced concept is a significant enhancement compared to conventional monitoring applications.
The present paper gives a short introduction to dam monitoring, outlines the selected design of the agent-based dam monitoring system and provides an insight into the current implementation. It is to be pointed out that the realization is still incomplete and a matter of research.

INTRODUCTION

The monitoring of security relevant structures is a task of growing importance in civil engineering. Large structures such as bridges and dams demand the use or application of precise measuring systems and the collaborative work of engineers, geologists and geodesists. Considering the time and labour consumed by the acquisition, processing and analysis of measured data, concerned authorities, operators and companies are trying to automate these operational procedures. The existing computer-based solutions focus on remote monitoring and neglect a collaborative analysis of measured data. However, an appropriate and effective modern monitoring system has to conduct all of the routine tasks performed by experts involved in monitoring.
The Institute of Computational Engineering, in cooperation with the Ruhrverband (Ruhr River Association), is currently developing a dam monitoring system based on software agents. The nucleus of the system's conceptual design is based upon the autonomous and collaborative analysis of measured data, associated with intelligent agents adopting the part of the experts generally involved in dam monitoring.

DAM MONITORING

The aim of dam monitoring is to provide indicators for anomalous structure behaviour. These indicators are used to take necessary countermeasures in due time and without any reduction in safety. In Germany, the rules of dam monitoring are given in the German Code E DIN 19700 (2001). Furthermore, recommendations for measuring devices have been published by the German Association for Water Resources and Land Improvement (DVWK) [1]. A typical example for the configuration of measuring devices is shown in figure 1.

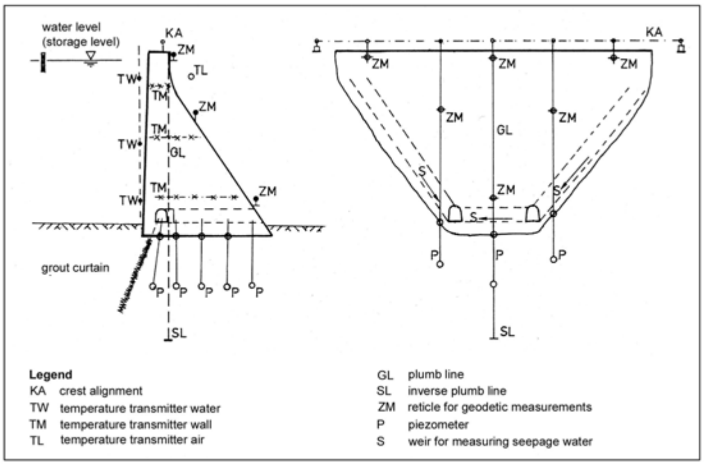

Figure 1 Example for the configuration of measuring devices [1]

The concept of dam monitoring is based on the systematic acquisition of all the relevant parameters, which encompass static, hydrologic and operational safety. Therefore, each dam structure must be provided with a measuring and control system, which, then, has to be adapted to the type, size and location of the structure.
The conceptual design of a monitoring system has to consider the following guidelines:

- Dam and bedrock form a unity, which is embedded in a natural environment.
- An anomalous structure behaviour can occur either gradually or quickly.
- When an anomalous behaviour occurs, the origin should be rapidly identifiable by an analysis of the measured parameters.
- Inspection by qualified personnel is indispensable.

In addition, the monitoring system must be adapted to the characteristics of the dam structure and has to take into consideration the corresponding measuring categories. At curved retaining walls it is important to monitor displacements while at gravity dams pore pressures are of particular importance in addition to displacements.
An automatic monitoring system rests on extensive electronic measuring equipment. This equipment consists of two essential components: transmitters (sensors) and data recorders (data loggers). Recommended transmitters are:

- temperature sensors,
- ultrasonic sensors for measuring seepage water,
- laser for measuring displacements,
- vibrating wire piezometers for measuring pore pressure.

The sensors are installed at specified positions (figure 1) inside the structure and they are controlled by electronic equipment (e.g. data loggers) sending electronic impulses. After having received an impulse, the sensors return a signal which can be a measurement of voltage, resistance or frequency. The electronic equipment scales the signal into a value, and either stores it in an internal memory or transfers it to a local database. Data stored in an internal memory can normally be received via a COMMS port (RS232). This interface also allows that the electronic equipment can be programmed from a host computer. An automatic monitoring system is customarily completed with a local computer, usually placed in a control room near the dam. As the redundant data storage is essential in dam monitoring, measured data are stored in a local database and additionally transferred to a central database [5].

CONCEPTUAL DESIGN OF THE DAM MONITORING SYSTEM

The analysis of the existing work flow applied by the Ruhrverband indicated that there is a chain of five tasks regularly performed by the responsible experts or in collaboration with other experts. Thus, the basic principle of the conceptual design is to map the regularly performed tasks, the individual experts and the interaction between themselves onto a multi-agent system. By that, the software agents can be divided into two categories: task-oriented agents mapping the routine tasks and personal agents mapping the experts involved in dam monitoring. The conceptual design for the organisation of the agents is shown in figure 2. On the left, the five routine tasks can be seen, in the middle the corresponding experts to be mapped onto the multi-agent system are depicted. Attention should be paid to the processing frequency of the monitoring tasks: the processing frequency decreases with every processing step, from temporal intervals of single days up to 1 year and more. Each task can be briefly described as follows:

1. Data acquisition:
   Involves the daily measurement of hydrologic parameters, in particular the seepage water, water level, water pressures, displacements and changes of temperature. The frequency of this task ranges from minute-by-minute intervals up to daily intervals, depending on the importance of the data.
2. Check of plausibility:
   Herein, the measured data are checked with respect to its plausibility. The checking must provide mechanisms to identify measuring errors, and to notify the concerned expert(s) in case of anomalous measurements. This task is also performed in minute-by-minute intervals up to daily intervals, depending on the importance of the data.
3. Check of short-time behaviour:
   The parameters measured within every week are checked. The concerned experts are notified in case of an anomalous behaviour. This task is performed weekly.
4. Check of long-time behaviour:
   This step involves checking the measured data for its long-time behaviour and is carried out annually.
5. Safety assessment:
   In this processing step the safety of the considered structure is verified officially. Measured parameters have to be scrutinized and the results of the analysis are to be summarized in terms of an annual report.

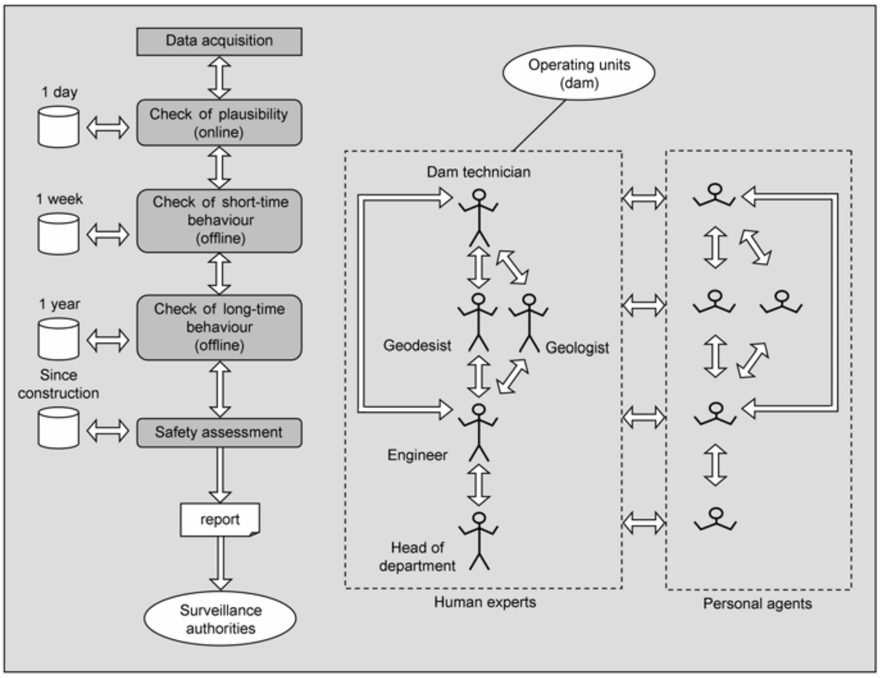

Figure 2 Conceptual design of the organisation of the agents

In order to provide smooth communication between the human experts and the multi-agent system, each human expert involved in dam monitoring is assisted by a personal agent; this software agent represents the interface between the MAS and the human experts and has to be proactive. The corresponding agents are organised using the same relationships as used by the human experts (see figure 2).

IMPLEMENTATION OF THE DAM MONITORING SYSTEM

General remarks

The implementation of the multi-agent system is still in progress and only established in parts, at present. Nevertheless, a brief report on the present work already achieved is to be given. In particular, the implementation concept used for the MAS is elucidated and the basic functionalities of the agents applied in the dam monitoring system are described. By that, the capabilities of the agents in the presented monitoring system can be demonstrated.
Specifically, two (dam monitoring) modules are represented, which are designed such that they can be incorporated into agents: the first module allows the control of automatic measuring devices and it will be implemented into an agent which is responsible for controlling all of the measuring devices installed within a dam wall. The second module, called evaluation module, provides processing and visualization of measured parameters. It will be integrated into the personal agents in order to process and visualize measured data according to the user's preferences by means of GIF, ASCII, HTML, XML or XLS formats.

Basic functionalities of the agents to be applied in dam monitoring

Considering the agents applied in the dam monitoring system, there are some basic requirements that have to be satisfied by the conceptual design of the agent architecture. In the following the chosen agent architecture is developed.
Interaction, and in particular its basis, **communication**, is the most essential element of networked and collaborative systems. Powerful solutions, if they are to be accepted by engineers, must provide several communication protocols for different requirements. For example, in some environments the connection to CORBA-based applications is used, whereas in other environments the HTTP protocol is required in order to avoid firewall problems. Considering such requirements, an architecture providing a replaceable communication layer is of particular importance. This is managed by the application of XML, since then XML has become the lingua franca for data exchange. Unfortunately, XML cannot solve the problem of the correct interpretation of exchanged data. Thus, a layer for **encoding** and **decoding** purposes is added (see figure 3). The inter-agent communication within the MAS is to be realised according to FIPA specifications [12], because FIPA is one of the central standards in the agent world. Furthermore, this approach allows inter-platform communication with other FIPA-compliant agents on various platforms.

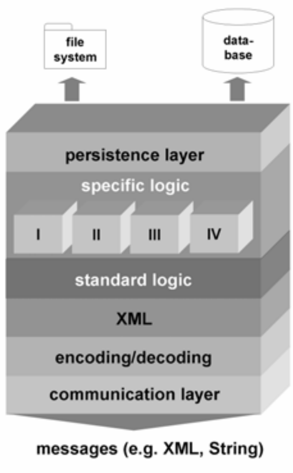

Figure 3 Conceptual design of the agent architecture

When dealing with complex monitoring problems, the agents have to be provided with logic. In the chosen architecture, the logic elements are divided into two categories: **standard logic** and **specific logic** (knowledge). In this particular case the standard logic contains the ontology of the domain “dam monitoring”, by what the agents possess the required vocabulary and basic knowledge in order to communicate and to execute simple tasks. The specific logic layer consists of several modules which are developed for a particular task or goal (in figure 3 named as I, II, III, IV). Via these modules the individual knowledge of the involved experts is integrated easily. This approach enables the user to adapt the agent to new tasks, goals or environments, too. In other words, the agent becomes more “intelligent” [8,9,10].
The last layer of the chosen architecture is a **persistence layer** by which the state of the agent is kept persistent. In case of a system crash this layer helps to identify the actual state of the agent and to continue the work without any loss of time.

Control of automatic measuring devices

As an important factor, a capable computer-based monitoring system must cover the applied electronic equipment. In order to control the measuring devices installed within the structure, there are two popular solutions: systems based on process control systems and systems based on data loggers. In the following, only data loggers are discussed. From an objective point of view, data loggers are the better and more transparent solution for dam operators in planning, use and maintenance.
Since the control of data loggers depends on the specific communication protocol and the instruction set predetermined by the specific manufacturer, a Java-based programming interface, called Logger API, has been developed to encapsulate specific loggers. Further specific loggers can be added to the developed library without expenditure.
The main benefit of the implemented library is demonstrated in figure 4. When integrating the API into Java applications, all of the measuring devices connected to a data logger can be controlled in a comfortable way: the functionalities provided by a data logger are accessed via simple, textual commands (*connect(), getStatus(), getMeasure()*).

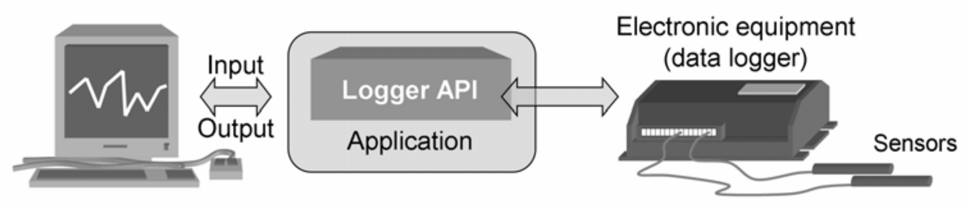

Figure 4 Control of a data logger via the Logger API

The applicability of the implemented API has already been verified in a test run at the Ennepe-dam. In this test, the installed data loggers associated with connected vibrating wire piezometers and temperature sensors could optimally be controlled via the Logger API [2].

Data processing and visualization – the evaluation module

Data processing and visualization are provided by an evaluation module which has been constructed as a web-based frontend in a first step. The web-based paradigm has been chosen such that an acceptance test could be performed in practice in a simple manner and so that no further client-sided software would be necessary.
In particular, the task of this web-based evaluation module is to read the acquired data of the dam monitoring from the database and to evaluate, edit and prepare the data in a user-oriented way, graphically and/or tabularly [3].
For this, the implementation is based on the Model View Controller paradigm (MVC), leading to a tripartite division composing the visualization component (i), the controller component (ii) and the database adapter (iii). Currently, the conventional MVC design pattern is changed towards a modular layer structure to allow a non-complex exchangeability of each component (figure 5). The functionality of each of these three components is briefly examined below.

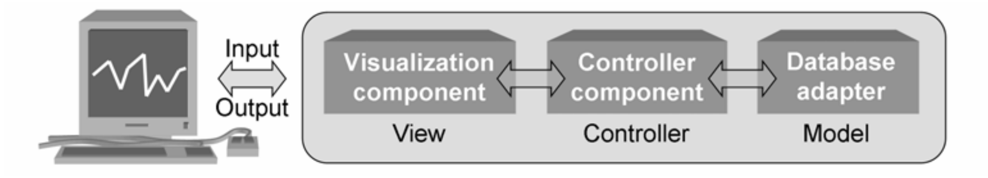

Figure 5 Pattern of the web-based evaluation module

(i) Visualization component

The visualization component (**view**) acts as a graphical user interface which allows database inquiries, administration of users, etc. (*inputs*) on the one hand and visualizes the requested result quantities in different data formats (*outputs*) on the other hand (figure 6).
Due to the demand for a web-based frontend and to accomplish sufficient security, the visualization component is realised using the Java Servlet Technology [11]. As an alternative, this component could be realised as a CGI program. An implemented Servlet class - the nucleus of this component - provides a GUI by using different helper classes. In order to follow the requirement for a differentiation of the user rights, user-specific and administrator-specific subclasses of these helper classes are distinguished.
The second task of the visualization component is the representation of the requested data in the format indicated by the user. This is ensured by an abstract class and its inheriting format-specific subclasses, for example ASCIIResult, GIFResult or XLSResult. By that, output objects are instantiated in order to produce the appropriate outputs depending on the desired format. In the case of an additional extension of the visualization component, regarding visualization in a non-implemented format, a new subclass has simply to be implemented and be announced in the central Servlet class. The contact to the controller component is managed by the Servlet class. Therefore, only one connection exists between the visualization component and the controller component. Consequently, other visualization components can be used without further significant expenditure.

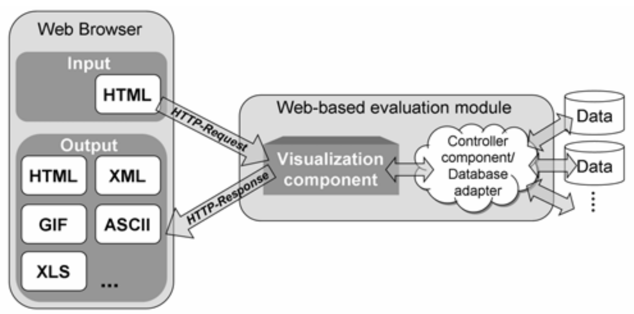

Figure 6 Function of the visualization component

(ii) Controller component

The controller component (**controller**) essentially consists of a controller module which takes over the user identification, and a request handler module which provides the structuring, evaluation and forwarding of the measured data requested by the user to the visualization component (figure 7).
A prime task of the controller is to read out the configuration file *config.xml*. This file contains the relevant data concerning all files involved, such as *user.xml* and *usergroup.xml* which contain information about characteristics of the existing users. These files are addressed by means of adapters. Thus, it is possible - instead of using XML files (used currently) - to apply, for example, a database for the administration of the user data. If a user and its usergroup is identified (as an administrator or as a „standard“ user), the controller instantiates a user object (with an associated usergroup object) which is associated with the controller and remains active throughout the entire session.
A further important aspect of the controller component is formatting the measured data as requested by the user and received via the database adapter. This is ensured by an abstract class *RequestHandler* and its format-specific subclasses. Each *RequestHandler* subclass implements format-specific methods for addressing the database adapter, requesting the desired measured data

and formatting the received data. The result will be returned to the *Result* subclasses of the visualization component described above.

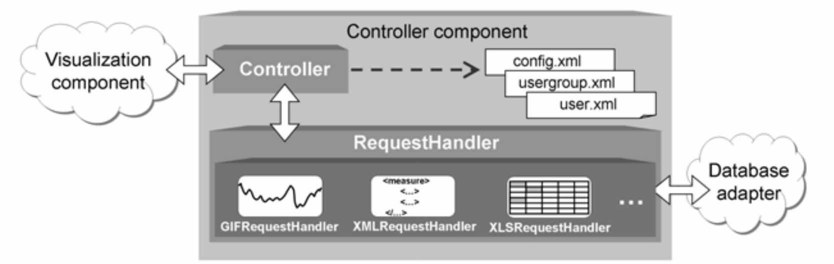

Figure 7 Function of the controller component

(iii) Database adapter

In order to be able to attach several (replaceable) databases, the **model** is realised as an exchangeable database adapter. It generates a connection between the database(s) and the controller component, passes on inquiries which concern measured data to the database and, finally, returns the received results to the controller component (figure 8).
Basically, the database adapter consists of an abstract class *DBAdapter* and its subclasses which are aligned to special databases. The class *DBAdapter* defines attributes and methods which are necessary for a connection with a database (as for instance *connect(), disconnect(), getStatus()*), as well as methods for the selection of the measured data, e.g. *getMeasures(…)*. As a subclass of *DBAdapter* the class *XindiceDBAdapter* has been implemented in order to encapsulate the native XML database "Xindice 1.0" utilized in this project.

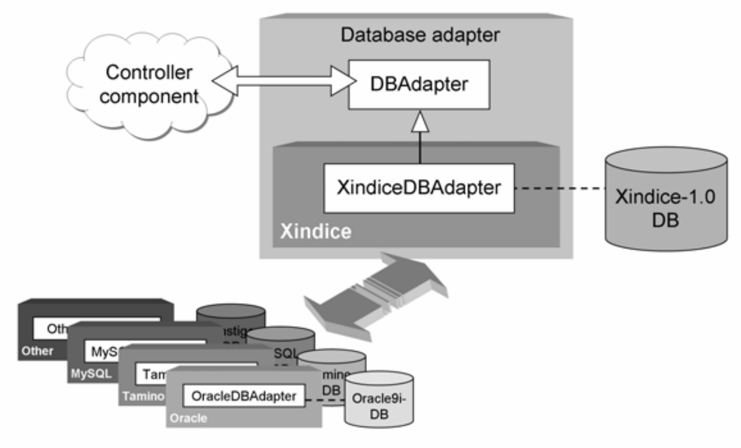

Figure 8 Illustration of the database adapter using the native XML database "Xindice 1.0" of the Apache Software Foundation

Due to the modularity and expandability of the evaluation module developed, it can be used in a multi-agent system, for example as a wrapper agent, which reads measured data from a database. A further possibility is the application of the module as an interface agent.

CONCLUSIONS

The present paper is a report on the enduring development of a multi-agent system for dam monitoring purposes. In particular, the conceptual design of the organisation and the architecture of the agents to be applied in the multi-agent system have been demonstrated. Furthermore, the implementation of two important modules - the logger API and the evaluation module - has been elucidated.

Actually, these two modules represent a conventional, web-based monitoring system. Accordingly, the measuring devices installed within a dam wall can be controlled online, and measured data can be read out of the databases and processed according to user preferences.
The multi-agent system presented is designed to map the distributed collaborative work of the concerned experts and to integrate their specific knowledge. It should be emphasized that the selected solution differs significantly from conventional monitoring systems and represents an innovative approach which is capable of demonstrating the enormous potential of agent-based applications.

ACKNOWLEDGEMENTS

The authors would like to thank the DFG (Deutsche Forschungsgemeinschaft) for its financial support.

REFERENCES

[1] German Association for Water Resources and Land Improvement (DVWK): Measuring devices for checking the stability of gravity dams and embankment dams, *DVWK-Merkblätter zur Wasserwirtschaft (DVWK-Guidelines)*, Volume 222, 1991.

[2] Mittrup, I.: Design of a web-based facility monitoring application for dam monitoring. Diploma thesis, Institute of Computational Engineering, Ruhr-Universität Bochum. Bochum, 2002. http://www.inf.bi.rub.de

[3] Smarsly, K.: Development of a web-based framework for evaluation, processing and visualization of measured data of a dam monitoring system. Diploma thesis, Institute of Computational Engineering, Ruhr-Universität Bochum. Bochum, 2002. http://www.inf.bi.rub.de

[4] Mittrup, I., Smarsly, K.: Development of a dam monitoring system by the example of the Ennepe-dam. Forum Bauinformatik 2002. Fortschritt-Berichte VDI, Reihe 4, Nr. 181.

[5] Bettzieche, V.: Experiences with the monitoring of dams. Scientific Reports. Journal of The Mittweida University of Technology and Economics. Volume III, Mittweida, 1/1997.

[6] Bettzieche, V.: In-depth analysis of measured data of dam monitoring. Scientific Reports. Journal of The Mittweida University of Technology and Economics. Mittweida, April 2002.

[7] Bettzieche, V., Heitefuss, C.: Monitoring as a basis of cost-effective rehabilitation of an old masonry dam; European ICOLD Symposium, Norway, 2001.

[8] Wooldridge, J.M., Jennings, N.R.: Agent Theories, Architectures, and Languages: A Survey. In: Intelligent Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages. Springer-Verlag, Lecture Notes in Artificial Intelligence, Vol. 890, S. 1-39, 1995. http://citeseer.nj.nec.com/wooldridge94agent.html

[9] Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley, Munich, 2001.

[10] Maes, P.: Modeling Adaptive Autonomous Agents. In: Artificial Life Journal, An Overview, edited by Christopher G. Langton, MIT Press, Cambridge, 1995.

[11] Sun Microsystems, Inc.: Java Servlet API Specification – Version 2.2, 1999. http://java.sun.com/products/servlet/2.2/

[12] Foundation for Intelligent Physical Agents (FIPA): FIPA Specifications, 2002. http://www.fipa.org