"2002 AI, Simulation and Planning in High Autonomous Systems - AIS'2002", 7-10 April 2002, Lisbon, Portugal

TOWARDS IMPROVING MULTI-AGENT SIMULATION IN SAFETY MANAGEMENT AND HAZARD CONTROL ENVIRONMENTS

Dionisis Kechagias
Andreas L. Symeonidis

Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
& Intelligent Systems and Data Engineering Lab,
Informatics and Telematics Institute/
Center for Research and Technology Hellas (ITI/CERTH)
Thessaloniki, Greece

Pericles A. Mitkas

Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
& Intelligent Systems and Data Engineering Lab,
Informatics and Telematics Institute/
Center for Research and Technology Hellas (ITI/CERTH)
Thessaloniki, Greece

Miguel Alborg

IDI EIKON, Parque Tecnológico de Valencia,
C/Benjamín Franklin, 27,46980 Paterna
Valencia, Spain

ABSTRACT

This paper introduces the capabilities of *Agent Academy* in the area of Safety Management and Hazard Control Systems. Agent Academy is a framework under development, which uses data mining techniques for training intelligent agents. This framework generates software agents with an initial degree of intelligence and trains them to manipulate complex tasks. The agents, are further integrated into a simulation multi-agent environment capable of managing issues in a hazardous environment, as well as regulating the parameters of the safety management strategy to be deployed in order to control the hazards. The initially created agents take part in long agent-to-agent transactions and their activities are formed into behavioural data, which are stored in a database. As soon as the amount of collected data increases sufficiently, a data mining process is initiated, in order to extract specific trends adapted by agents and improve their intelligence. The result of the overall procedure aims to improve the simulation environment of safety management. The communication of agents as well as the architectural characteristics of the simulation environment adheres to the set of specifications imposed by the Foundation for Intelligent Physical Agents (FIPA).
Keywords: *agents, multi-agent systems, hazard control, data mining, Agent Academy*

1 INTRODUCTION

Intelligent agent (IA) technology promises to enable an enormous explosion of brand new computer-based services. The use of IAs introduces referencing capabilities to software, transforming computers into personal collaborators that can provide active assistance and even take the initiative in decision-making processes. What makes an intelligent agent different from a typical computational procedure is the fact that intelligent agents posses among other characteristics, the ability to satisfy their goals and create new ones, based upon their former interaction with other agents. To accomplish this IAs must be capable of learning [MAE94], in order to modify their behaviour and make new decisions based on the certain goals, which they try to satisfy. For the purpose of learning, some techniques arisen from the area of Artificial Intelligence can be used. Keeping this in mind, we designed Agent Academy (AA), a framework for training agents in order to make them "smarter", using data mining (DM) techniques.

The data mining approach introduces a new set of tools and methodologies for discovering knowledge and patterns in large databases [FAY98]. Mining information and knowledge has been recognized by many researchers as a key research topic in database systems and machine learning. Collecting, therefore, the content of messages exchanged between agents in a large data repository, has given us the ability to "mine" this data, in order to discover new agent behavioral patterns, and to apply these patterns to existing agents. The aforementioned procedure of monitoring agents, discovering knowledge on their behavior and embedding it back to them, is realized through Agent Academy.

The main goal of AA is to provide a platform for enhancing agents' functionality and intelligence through the use of DM techniques. This particular framework is concerned with two major issues: the first one focuses on the way the AA trains the agents, in order to provide them with the necessary skills to successfully accomplish their assigned tasks. This mechanism combines the use of DM techniques to improve the decision making process in IA, the training of the newly "born" agents, as well as the way the AA deals with different types of agents. The second major issue involves techniques by which the AA obtains information on the way it should train its own agents.

This new approach can facilitate "smart" solutions in Small and Medium Enterprises concerning a wide range of services provision including management, resource allocation and remote administration in both "safe" and "hazardous" environments.

At first in this paper we describe the overall functionality of Agent Academy. Then we briefly discuss the standards of agent platform architecture and agent communication implied by FIPA [URL1]. Later on, we discuss in detail an application scenario of AA, which is based on the creation of a multi-agent community that handles the parameters of a Safety Management system. This application paradigm demonstrates in general the use of AA trained agents and the contribution to Hazard Control cases in particular. Finally, after presenting a set of already-developed tools that we integrated in our platform, we conclude this paper and foresee all possible extensions, which may fit in the case of safety control, as well as complementary future work.

2 FUNCTIONAL DESCRIPTION OF AGENT ACADEMY

Agent Academy forms an integrated framework that receives input from its users and the Web. The main block diagram of AA is illustrated in Figure 1 and shows the communication of its internal components. AA operates as a multi agent system, which can train new agents or retrain its own agents in a recursive mode. A user issues a request for a new agent as a set of functional specifications. The Agent Factory, a module responsible for selecting the most appropriate agent type and supplying the base code for it, handles the request. A newly created untrained agent (UA) comprises a minimal degree of intelligence, defined by quite primitive behavioural characteristics, such as the awareness of learning. This agent enters the *Agent-Training Module* (ATM), and improves its referencing mechanism by adopting a set of decision rules in the form of a decision-tree, which is efficiently formulated. This process may include modifications in the agent's decision path traversal and application of augmented adaptivity in real transaction environments.

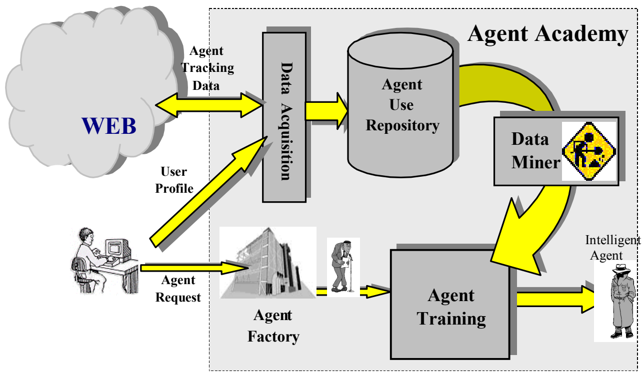

Figure 1: The main Agent Academy architecture

The core of AA is the *Agent Use Repository* (AUR), which is a collection of data on prior agent behavior and experience. It is on the contents of AUR, where data mining techniques, such as extraction of association rules for the decision making process, are applied in order to augment the intelligence of the AM in the training module. Building AUR is a continuous process performed by a large number of mobile agents that are controlled by the *Data Acquisition Module*. A large part of an agent's intelligence handles the knowledge acquired by the agent since the beginning of its social life through the interaction with the environment it acts upon.

2.1 Basic Components of Agent Academy

The main components of the system can be described as follows:

Agent Factory (AF): The Agent Factory is responsible for producing "untrained agents". There can be several types of untrained agents distinguished not by their behaviors or beliefs but by their hard-coded abilities such as the processing functions/power they have, the external applications or devices they can access, their ability to mobilize or lack thereof. The agent factory selects and assembles the correct type of agent according to the requests of the user.

Agent Training Module (ATM): Also called the 'agent master' this module is responsible for training untrained agents with the necessary behaviors, knowledge base and goals which it derives from preferences of the user requesting the agent. The training module acquires the relevant relationships, facts and knowledge from the Data Miner. The ATM also has the ability to re-train already active agents.

Data Mining Module (DMM): The Data-Mining Module, also known as "Data Miner", is charged with the task of generating associations, categorizations and sequences in the data collected so far. These are used by the ATM during the agent-training procedure.

Agent Use Repository (AUR): Another core component of the system is the agent use repository, which is a collection of statistical data, as well as meta-data on prior agent behavior, experience and facts about the working domain or environment. AUR is the structure holding body of data mentioned above. The AUR is constantly updated and appended with information form the *data acquisition module* that controls data coming in from the operating agents as well as from the environment.

3 THE HAZARD CONTROL SCENARIO

In order to measure the efficiency of Agent Academy trained agents we developed a test case scenario of a real-time system, which receives environmental data through sensors and notifies humans about forthcoming meteorological hazards. Our effort is concentrated on the creation of a multi-agent software system, which will simulate the real time application described above. For this purpose, we first identify the critical environmental parameters, which have to be measured, and then we propose an architecture for modeling the real time system in terms of a multi-agent environment. In this environment each agent receives some input from the environment through sensors, or from another agent, exchanging FIPA-SL0 messages. In our experiment we will be based on real data collected on a time-fashion basis, provided by the Mediterranean Centre For Environmental Studies Foundation (CEAM), which is located in Valencia, Spain. Finally, we describe the procedure of passing data as input to the DM process which will produce decision rules.

3.1 The Real Time System

This experiment was based on a real time application, which was developed without the use of intelligent agents. This system collects periodically atmospheric data, detects the implosion and pressure of certain gases on the stratosphere (nitric monoxide, nitric dioxide, carbon dioxide, etc.), analyses this data and finally triggers ozone alarms. This procedure can facilitate a safety management mechanism, which provide environmental alerts to both individuals (asthma patients, allergic people etc.) and public authorities (hospitals, civil protection etc.) depending on the levels of certain atmospheric variables, which are collected by CEAM. The system is supposed to monitor the incoming data from the environmental sensors and make predictions as to the effects that a modification of a certain input parameter may cause in the environment. Some changes will lead to certain kind of alerts, which then need to be transferred effectively to parties interested in this particular alert.

We noticed that this real time monitoring application receives multiple values of equal in number parameters as an input and a decision on triggering an alert is made as an output. Therefore, it is theoretically correct for the DM procedure to assume that a correlation exists between input and output parameters. This correlation can only be identified or even probabilistically evaluated by the application of DM algorithms. Since there is a very large amount of data, the DM procedure is able to be executed in order to identify some hidden relationships between input and output data. Integrating the DM procedure on environmental data into the AA framework, we are able to produce decision trees on ozone alarm and express them in form of ACL messages. These messages can then be used in the agent training process in order to improve the initial intelligence of an agent. Eventually, we will produce specific referencing rules and apply them to newly trained agents.

3.2 The Multi-Agent Simulation Environment

In order to simulate the real time system we have to provide an agent-based architectural design. First, we track the whole process from receiving input data through sensor devices to triggering an alert. This procedure helps us to break the overall decision support system into smaller components, which collaborate with each other in order to lead the system to a final decision. Each component can be considered as a black box that receives some inputs and provides output to another component. From this perspective, we can model every decision-support component as a single agent. The multi-agent simulation environment is depicted in Figure 2.

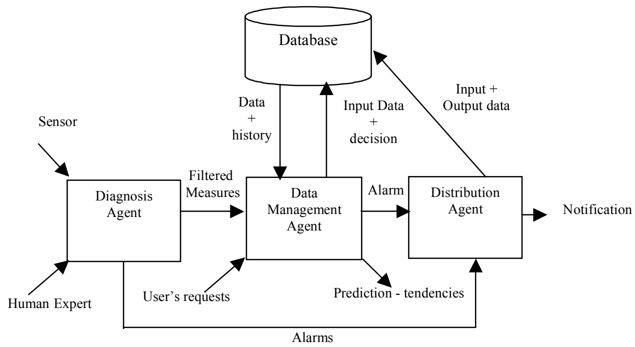

Figure 2: The multi agent hazard control environment

In this reference model we identify three types of agents: a) Diagnosis Agents, b) Data Management Agents, and c) Distribution Agents, which communicate in the same environment and exchange data through a database. These types of agents are described in more detail as follows:

3.2.1 Diagnosis Agents

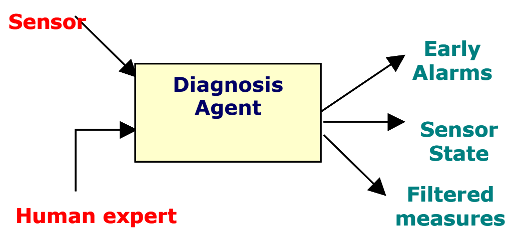

Figure 3: Input and output of Diagnosis agents

Agents of this type receive input from sensors, which monitor the environmental variables, or from a human expert who enters values into system manually. These agents have the responsibility to filter incoming data, accepting only valid values and discarding all other forms of irregular or corrupted data. In Figure 3 the input and output of this type of agents are shown.

Diagnosis agents act as a first filter of the overall system separating nuggets of critical information from data dirt. However, they are given the opportunity to post early alarms in rather hazardous situations, which can be predicted at the first stage of data input. They also can provide information about sensor status.

3.2.2 The Data Management Agents

Agents which fall into this category, are also called Predictor Agents, because they study continuously the incoming data, together with the historical data stored in the database and try to identify common patterns, in order to predict and fire an early alarm to the distribution agent. Agents of this type communicate with the main database system and handle information, which is constantly augmented.

Figure 4 depicts the possible input values that this type of agent receives and the output that it produces.

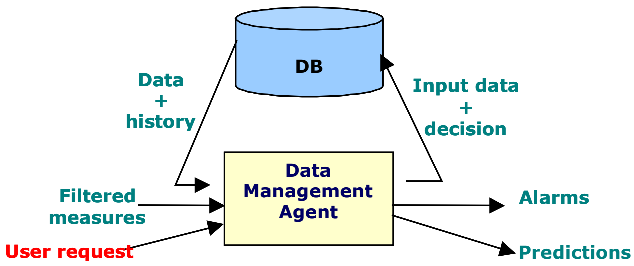

Figure 4: Input and output of Data Management agents

3.2.3 The Distribution agents

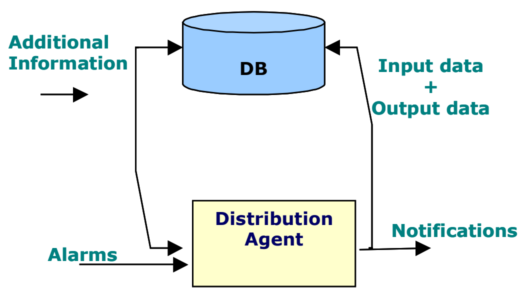

Figure 5: Input and output of Distribution agents

When an alarm is fired by the prediction component, it is sent to the distribution component, which decides who will be affected by the alarm based on the profiles of the users and their current locations. The functionality of a distribution agent is illustrated in Figure 5. This agent is the one that finally handles the notification of end users and decides to send an alert or not. In case of alert, the distribution agent looks up the database and creates a list of users to inform. Once this list is generated, the alarm is sent to each user over the communication channel preferred by him/her (e-mail, html, voice, SMS etc.).

4 UNDERLYING TOOLS AND TECHNOLOGIES

Agent Academy, adheres to FIPA Specifications and also integrates a set of already developed tools in order to realize a framework with combined functionality.

4.1 FIPA Specifications

AA operates as a multi-agent platform, which is compliant to FIPA specifications and uses FIPA-SL0 [FIPA00008] as an agent communication language. The Foundation for Physical Intelligent Agents (FIPA) [URL1] is an international non-profit association of many companies and organizations, which share the common effort to produce and provide standards on agent technology. Specifically, FIPA proposes, among others, an agent management reference model [FIPA00023] and a set of specifications on message transport between agents in general and in communication protocols, Agent Communication Languages (ACL) [FIPA00008], transport protocols and message envelops in particular. Two of FIPA specifications, which are of great importance are: a) the specification of agent management architecture [FIPA00023] and b) the definition of an ACL [FIPA00061], which provides a formal representation of message content.

4.2 JADE

For the realization of the main agent platform of AA, the JADE [URL3], [BEL00], [BEL99] platform was chosen. This provides an application programming interface (API) for Java programmers together with a set of GUIs for monitoring agent behaviour in an AP. JADE fully conforms to the FIPA specification standards and it natively supports the FIPA-SL0 communication language.

4.3 WEKA

The Waikato Environment for Knowledge Analysis (WEKA) [URL4] provides an API of Java classes and a set of algorithms that implement several known DM techniques.

4.4 JESS

Java Expert Systems Shell (JESS) [URL5] realizes a rule engine that AA makes use of in order to apply inferencing capabilities to its agents. The beliefs of the agents are represented as JESS rules, and when a new fact coming from the environment is asserted to the knowledge database of the agent, these JESS rules are executed automatically and the action that is defined to be the result of this condition is executed. The agents created by Agent Factory, gain the ability to execute JESS rules, after an initial training by the Agent Training Module.

5 CONCLUSIONS AND FUTURE WORK

In this paper we introduced Agent Academy and described the fundamental operations that supports. After a discussion on FIPA standards we presented a hazard control application scenario of an environmental alert system. We developed a modeling of this system as a multi-agent system and identified three types of agents which can be efficiently trained by the AA framework based on a-priori data provided by CEAM, a Spanish national organization acting in the environmental affairs. Using DM techniques we can train our agents in the simulated multi-agent environment. These trained agents have encapsulated a certain degree of knowledge, which becomes greater as the amount of the provided data increases.

The next step of our prototype is to use all the already trained agents in a real-time environment of ozone alarm. This implies the implementation of a suitable software interface to allow the perception of real time data by the diagnosis agents. No further modification of agent code is needed since the already trained agents are aware of handling real-time data and this is something that they “learned” during the process of their training.

ACKNOWLEDGEMENTS

The Agent Academy framework is supported by the European Commission as an IST-funded project under the identifier IST-2000-31050. AA is still under development.

Authors would like to especially thank IDI-EIKON for their active assistance on the creation of safety management and hazard control scenario and the Mediterranean Centre For Environmental Studies Foundation (CEAM) for providing us environmental data.

REFERENCES

[BEL00] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, Paola Turci , “An Object-Oriented Framework to realize Agent Systems”, *Proceedings of WOA 2000 Workshop,* Parma, May 2000*,* pp. 52-57.

[BEL99] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, “JADE – A FIPA-compliant agent framework”, *CSELT internal technical report.* Part of this report has been also published in Proceedings of PAAM'99*,* London, April 1999, pp.97-108

[FAY98] Usama Fayyat, “Mining Databases: Towards Algorithms for Knowledge Discovery”, *Bulletin of the Technical Committee on Data Engineering,* Vol 21, No 1, March 1998, pp39-48.

[FIPA00008] FIPA SL Content Language Specification, Foundation for Physical Agents, 2000 `http://www/fipa.org/specs/fipa00008`

[FIPA00023] FIPA Agent Management Specification, Foundation for Physical Agents, 2000, `http://www/fipa.org/specs/fipa00023`

[FIPA00061] FIPA ACL Message Structure Specification, Foundation for Physical Agents, 2000, `http://www/fipa.org/specs/fipa00061`

[HAN01] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, May 2001

[MAE94] Pattie Maes, “Agents that Reduce Work and Information Overload”, *Communication of the ACM* July 1994/Vol. 37, No. 7, 31-40. (p. 811)

[URL1] The Foundation of Physical Intelligent Agents, `http://www.fipa.org`

[URL2] The Mediterranean Center for Environmental Studies Foundation (CEAM) `http://www.gva.es/ceam`

[URL3] Java Agent Development Environment - JADE, `http://jade.cselt.it`

[URL4] WEKA Set of tools for data mining, `http://www.cs.waikato.ac.nz/ml/weka/`

[URL5] Java Expert System Shell – JESS, `http://herzberg.ca.sandia.gov/jess/`

AUTHOR BIOGRAPHIES

Dionisis Kechagias is a software engineer and researcher with Informatics and Telematics Institute of the Center for Research and Technolgy Hellas, with active involvement in the Agent Academy IST project. His areas of expertise are Intelligent Agents, Distributed Artificial Intelligence, Multi-Agent Systems and Mobile Agents. Mr. Kechagias has a Diploma in Electrical and Computer Engineering from Aristotle University of Thessaloniki and he is currently a PhD candidate in the same department. He is a student member of IEEE and the Computer Society and a member of the Technical Chamber of Greece. His email and web addresses are <diok@iti.gr> and <danae.ee.auth.gr/en/people/diok.htm> respectively.

Andreas L. Symeonidis is a software engineer and researcher with ITI with active involvement in the Agent Academy IST project. His areas of expertise are data mining, knowledge discovery and their applications on e-commerce. Mr. Symeonidis has a Diploma in Electrical and Computer Engineering from Aristotle University of Thessaloniki and he is currently a PhD candidate in the same department. He is a student member of IEEE and the Computer Society and a member of the Technical Chamber of Greece. His email and web addresses are <asymeon@ee.auth.gr> and <danae.ee.auth.gr/en/people/asymeon.htm> respectively

Dr. Pericles A. Mitkas is an Associate Professor with the Department of Electrical and Computer Engineering at the Aristotle University of Thessaloniki and a faculty affiliate of the Informatics and Telematics Institute in Thessaloniki, Greece. His research interests include parallel architectures for large multimedia databases, data mining and data warehousing, mobile agents and e-commerce, and bioinformatics. Between 1990 and 1999 he was an Assistant and Associate Professor with the Department of Electrical and Computer Engineering at Colorado State University where he performed research on parallel storage and processing systems for large databases. Dr. Mitkas' work has been reported in over 100 journal and conference publications, several invited lectures and five book chapters. He has also organized four conferences. Dr Mitkas is a senior member of IEEE and the Computer Society. His email and web addresses are `<mitkas@eng.auth.gr>` and `<danae.ee.auth.gr/en/people/mitkas.htm>` respectively

Miguel Alborg is a founding member and Managing director of IDI Eikon. He is also the head of the R&D department of IDI Eikon. He has a comprehensive understanding of business drivers, critical success factors and key issues in the media industry, gained through working in this sector for twenty years. He is mainly devoted to make smoothly integration of human and social issues in research and development projects. Mr. Alborg is especially interested in environmental matters and follows very closely all related topics. His email and web addresses are `<alborg@terra.es>` and `<www.idieikon.com>` respectively