In **Second International ICSC Symposium on Information Technologies in Environmental Engineering (ITEE 2005)**, (Leal-Filho, W. and others, Eds.), Magdeburg, Germany, Shaker Verlag, September 2005, pp.253-267.

An Agent-based Middleware for Environmental Information Management

Ioannis N. Athanasiadis

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Switzerland*

Andreas Solsbach

Technische Universität Clausthal, Germany

Pericles A. Mitkas

Aristotle University of Thessaloniki & Centre for Research and Technology – Hellas, Greece

Jorge Marx-Gómez

Otto-von-Guericke-Universität Magdeburg,Germany

Abstract: Managing environmental information is a demanding task, which engages serious amounts of efforts by environmental scientists working in public institutes or the industrial sector. In order to reduce the workload required and disengage environmental scientists from trivial tasks, as data transformation and reviewing, we developed an Agent-based Middleware for Environmental Information Management (AMEIM for short). Our approach, presented in this paper, utilizes software agents that undertake environmental data management tasks. Software agents in AMEIM are capable to fuse and preprocess environmental data. Taking under account the requirements of the application domain, AMEIM's core functionalities, the agent-based architecture and the platform developed in Java are detailed. The AMEIM system is fully customizable and follows an extendable architecture. Also, reasoning capabilities can be incorporated into AMEIM agents for supporting decision-support features.

Keywords: Environmental information management, software agents, data fusion, preprocessing, decision support.

* This work has been conducted, while the author was with the Electrical and Computer Engineering Dept. of the Aristotle University of Thessaloniki and the Informatics and Telematics Institute of the Centre for Research and Technology - Hellas.
Contact email address: ioannis@athanasiadis.info

1 Introduction

In the last decades, there has been a remarkable transition of society's values related with the natural environment and sustainability. As a consequence, the public, non-governmental organizations, local authorities and the government ask for access to environmental data and information. In this respect, Environmental Information Systems are called to expand their target user groups with the above-mentioned "new users", and incorporate sophisticated services for satisfying accordingly the letter's needs. As a result, data management activities of public environmental monitoring centers or private industries are required to be strengthened by supporting advanced information services. The common practice in both public and industrial sector, for many years, was the use of environmental data for in-house purposes and studies. Typically, environmental data are stored in several diverse formats and media in local repositories, not easily reusable and in no case accessible by the third parties, as the new users.

The advances of the Information Technology (IT) sector can be proven valuable in the specific problem-space. The key characteristics of the environmental data management domain are mounted on two pillars. This first concerns with data availability and quality. Swayne recently underlined that "*the problems of data quality and availability in environmental systems are areas of research that continue to require support*" and that *"the advances in database technology are not uniformly available in the environmental domain"* [Sway03]. The second pillar is related to the ever-increasing needs for environmental data. Current policies both in Europe and in the United States impose that environmental data, as public sector information should be disseminated to everyone [EC98].

In this background, it is acknowledged that significant efforts are required for providing environmental information services to wide audiences, through the exploitation of IT methods. Information Technology is not simply required to overcome the obstacles of legacy systems, data noise and redundancy, lack of data standardization and variety of data formatting and aggregation, but to automate the environmental data review processes by incorporating decision support capabilities. Towards this direction drives the work presented in this paper that utilizes software agent technology for managing environmental data, stored in various formats and diverse sources, and providing ultimately advanced information services to a wider user group.

2. Environmental Information Systems with agents

2.1 Environmental information systems

There are several definitions of Environmental Information Systems (EIS) in the literature. One of the most generic ones was given by Günter: "*EIS are concerned with the management of data about the soil, the water, the air, and the species in the world around us*" [Günt98]. Günter also proposes a conceptual framework for EIS that structures the data flow into 4 phases: *data capture*, *storage*, *analysis*, and *metadata management*. This flow corresponds to a complex aggregation process gradually transforming the incoming raw data into concise documents suitable for high-level decision support [Günt98]. It is commonly accepted that a straightforward data storage procedure is not sufficient for environmental data. Rather the filtering of meaningful and up-to-date information on the state of the environment is required to support administrative and planning tasks related to environmental protection [PaRa01]. Hence, EIS goal is to process environmental data in order to increase both data (re)usability and quality. In addition, an EIS is required to deal with both domain uncertainties encapsulated in data and data overflow of environmental information. Various user groups have to be served by an EIS, including the environmental scientists and institutes, as well as the public, NGOs, industry and the government. In this paper, we present how EIS objectives can be satisfied using software agents, which mediate between diverse data sources and the final users.

2.2 Agent technology

In the above-mentioned background, we utilized software agent technology for achieving the struggling task of managing environmental information. The last decades, agent technology has emerged as a powerful metaphor for developing software, which builds upon (and takes advantage of) several disciplines of computer science, including artificial intelligence, human-computer interaction, computer networks, and software engineering. As agent technology is still on the rise, there is not a commonly accredited definition of agent and its characteristics. However, one of the most generic definitions is that of Franklin and Graesser: *"An autonomous agent is a system situated within and a part of an environment, that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future"* [FrGr96]. In that respect, key characteristics of agency are the abilities of sensing the environment, acting autonomously and taking decisions. However, an agent can be characterized by some of the following attributes: temporal continuity, responsiveness, proactiveness, social ability, mobility, veracity, benevolence, rationality, cooperation, character, and adaptiveness [Etsi95, WoJe95].

2.3 Related work

Although an agent can be considered as a powerful metaphor that can be utilized as a common abstraction for the entire software development process, including modeling and development stages, agent technology has been exploited to a limited extent in the environmental domain. Various agent-based practices for environmental software have been applied in diverse fields, varying from simulation platforms to distributed problem solving. In this paper, we focus on agent-based applications related to environmental data management issues.

In this type of EIS, there are just a few agent-based attempts reported in the literature. Among them is the EDEN-IW (Environmental Data Exchange Network for Inland Water) platform. EDEN-IW [Fell+03] uses *"independent agents"* for supplying scientists, citizens and decision-makers with environmental-related information, through an intelligent interface that operates as a "one-stop-shop". Another application is the NZDIS System (New Zealand Distributed Information System) [Purv+03]. In NZDIS, software agents are used for submitting queries to environmental databases in a seamless way. A third application reported in the literature is the Forecast Streamlining and Enhancement Project, developed for the Australian Bureau of Meteorology. In this approach, agents are used for identifying and using data and services in an open, distributed environment [Dan+03]. The common ground among these applications is that in all them, software agents are used as interfaces for accessing information stored in distributed databases. However, this is only a part of an environmental data management system's functionalities, which is required to incorporate customizable information services for a variety of users.

In this paper, we build upon our experiences in developing environmental data management systems with agents. In previous works, we have used software agents, situated in a layered architecture, for providing advanced information services, in different application fields. In particular, the O_3RTAA system employs software agents for assessing air quality data [AtMi04a, Atha+03], while the ABACUS system uses an analogous architecture for managing meteorological radar data [Atha+05]. Based on these two applications, we proposed (elsewhere) a generic methodology for applying agent technology in environmental data management systems [AtMi04c]. Taking a step ahead, this paper presents a reusable platform, which realizes a generic architecture for developing agent-based systems, operating as a middleware application between environmental data pools and the final users of environmental information.

2.4 Problem description

As mentioned above both environmental institutes and private companies require EIS, for managing environmental information. The key issue in such systems is to capture, preprocess and consequently diffuse environmental information in custom presentation formats in an automated, transparent way that will require minor interference of environmental scientists. In a generic view of such system, data could be stored in remote locations and in diverse formats, while preprocessing activities, data reviewing and transformation for homogeneous presentation would require significant time and effort by the environmental experts. The main objective of AMEIM is to intervene between environmental data fountains and the final users, for assisting the environmental experts with their work, by:

a. Providing transparent access to diverse environmental data pools,
b. Reviewing and validating environmental data (as uncertainties are typically involved in the environmental domain),
c. Supplying the end users with customized, ready-to-use information, in different abstraction and aggregation levels.

In order to realize these goals, AMEIM exploits agent technology for providing the infrastructure required for delivering critical tasks involved in environmental information processing, which typically require significant efforts by human experts. These tasks are undertaken by AMEIM's agents who are responsible for accessing environmental data sources, pre-processing, validating and fusing raw data into environmental information schemes, and disseminating and reporting environmental information.

The rest of this paper is structured as follows: Section 3 presents AMEIM's software requirements specification, Section 4 the agent-based design of the platform and Section 5 presents the implantation details. Finally, the developed platform is demonstrated.

3 AMEIM Key Requirements

3.1 AMEIM's scope and operational environment

AMEIM has been designed to intervene between environmental data pools and the end-user applications. An abstract view of the system is shown in Figure 1. AMEIM interweaves multiple *data streams* produced by (or lying in) the Environmental Data Pools and supplies the End-User Applications with pre-processed, ready-to-use *information*. Sensor devices, remotely accessed documents (via FTP or HTTP), or locally stored databases and files, operate as the data fountains of the system. Even if both the topology and the procedure may vary for each installation, we use the generic term *data streams* to describe the data perceived by sensor devices or lying in a local or remote data storage medium. These data streams are the inputs of the AMEIM system and inherit the critical characteristics of environmental data, such as low reliability, redundancy, and poor semantics. In most cases, data streams are incompatible with the needs of the majority of the end-user applications, including database systems, domain specific software, and reporting services over the web. In such environments, AISLE operates as an intermediate layer supporting data fusion and dissemination.

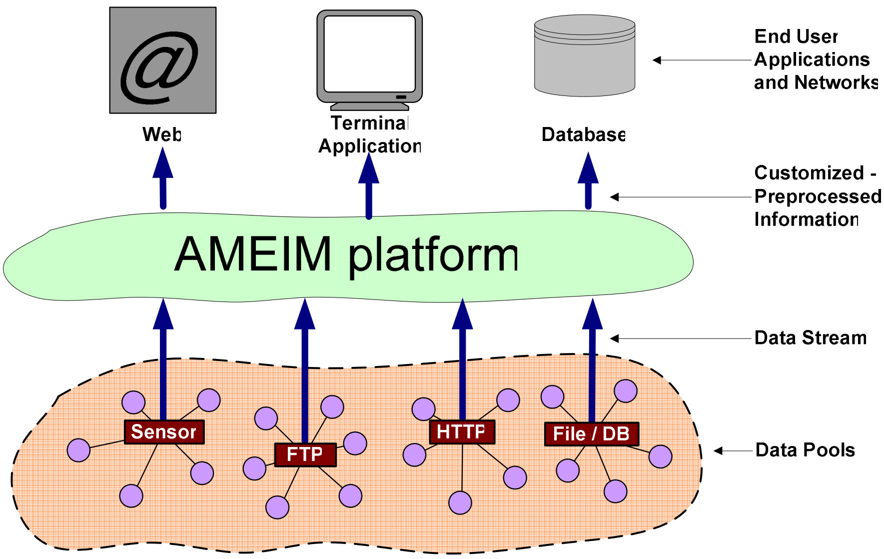

Figure 1. AMEIM operational environment

3.2 Abstract architecture

In order to realize AMEIM's objectives, its key functionalities have been grouped in three distinct, yet cooperative clusters of services:

a. The Contribution Cluster, which is responsible for data collection and validation.

b. The Management Cluster that focuses on data management and pre-processing activities.

c. The Distribution Cluster, which provides the appropriate interfaces to end-user applications.

The AMEIM mechanism, as a synergy of clusters, is shown in Figure 2. Incoming data streams traverse among the three clusters, which form a hierarchy. The outcome of the system is customized information to the end users.

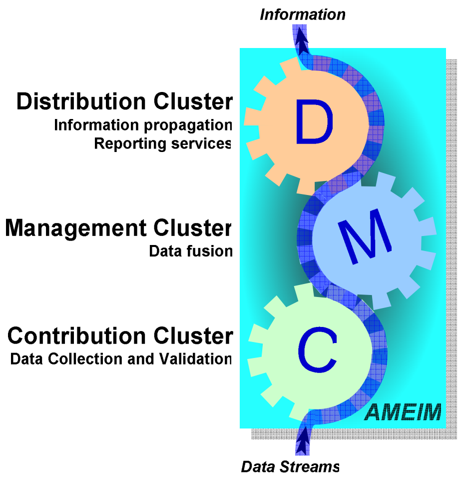

Figure 2. The AMEIM mechanism, as a synergy of three co-operative clusters of services.

Software agents, which are the building blocks of AMEIM, undertake the key responsibilities of each cluster. There are four distinct generic agent types:

1. Contribution Agents (CA), acting as the implicit data receptors of AISLE.
2. Data Management Agents (DMA), responsible for data fusion and pre-processing.
3. Distribution Agents (DA), acting as an interface to the end-user applications.
4. The Graphical User Interface Agent, which interacts with the platform administrator and orchestrates the platform agents.

3.2 User Classes and Characteristics

AMEIM's users are separated into two groups: *direct* and *indirect* users. "Direct" are characterized these users who actually use the platform, while "indirect" are those who will benefit from its services. In this respect, direct users are the environmental scientists, the platform administrators and the computer scientists/developers. Indirect users include the government, the industry and the public.

Environmental scientists work on a frequent basis with environmental data fused by the AMEIM platform. In principal, there are no programming skills required by them for using AMEIM's graphical user interface. Using the GUI they will be enabled to operate the platform, and to add new (or customize existing) services and data pools by activating the appropriate codes written by the software engineers.

Software Engineers – Developers will add new functions to the system in order to extend its functionality. The code of these functions can be re-used by the environmental scientists through the GUI. The frequency of their use of the system is determined by the necessity of getting new features and releases to the system.

Administrators are responsible for updating the system when a new release or new features are available. Typically, they will work rarely on the system.

The indirect users are those who benefit from the information services supplied by the system. Typically, they do not need any programming skills, as they do not work directly on the system. In this category fall:

- The **Governmental institutes** that acquire information, through the platform, on a regular basis, according to legal guidelines or institutional practices.
- The **Private companies** and **industries**; these get access to environmental information, custom presentations and reports through the system. Their frequency of system use may vary according to their business activity domain.
- The **public** is informed by custom handouts and web services for the conditions of the living environment.

3.4 System Features

In an abstract level, data are manipulated by agents as follows: In the Contribution layer data are captured and validated, in the Management layer all the processing and transformation/aggregation activities take place and finally, through the Distribution layer they are delivered to the direct and indirect users of the system. In the followings, the main features of AMEIM platform are detailed.

In the **Contribution layer,** CAs acquire data originally in different format types and in diverse sources, as files on internet servers, databases or sensors. Each CA is responsible to review and validate incoming data streams and ultimately to deliver preprocessed data to DMAs. CAs is required to differentiate their behavior for the various input sources, following "pull" or "push" strategies. Input sources that push data in the system (as for example sensors), typically have no local storage, so CAs need to monitor them continuously and capture data as they are produced (i.e. in an online way). On the other, data stored in files or databases have to pulled by CAs. In this respect, CAs is required to query these mediums for extracting the appropriate data.

In the **Management layer**, DMAs fuse data coming from diverse sources in a common data scheme, creating joint presentations (views on data). According to the direct user inquires, DMAs prepare presentations, files or data output streams, which are further disseminated through the Distribution layer, via web services, email, or other forms. Direct users have full control of the data manipulation process, and are enabled to load preprocessing functions in the DMAs.

In the **Distribution layer**, DAs deliver the processed information to their final destination (indirect or direct users). DAs can use several ways of diffusing information as web messages, email and handouts, via SMPT, FTP or HTTP protocols. Direct Users are enabled to define email distribution lists, and to customize these services.

4. AMEIM design

4.1 Architecture Design

Figure 3 illustrates the conceptual design of the AMEIM system, as a layered agent architecture. A set of CAs, DMAs and DAs cooperate in a structured way for delivering information services to the end users.

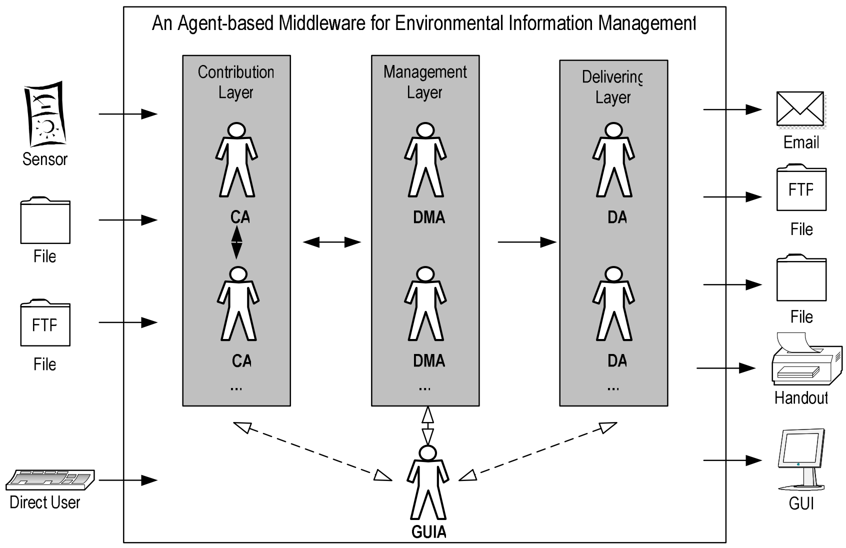

Figure 3: Architecture design of AMEIM system

Data flows through the system and pass through all three layers. The contribution agents acquire and review raw data from sensor devices or from local or remote files and databases. CAs push validated data to the processing layer. In the processing layer, each DMA receives information from several CAs (i.e. from different original sources), preprocess it appropriately, and ultimately sends customized data views to the delivering layer. There, DAs take care for delivering the output of the system to the end users. The output of the system could be a file, an email, a handout or a notification on the GUI. All agents in the system have a small memory to save their operational parameters.

4.2 Agent roles in AMEIM

The Gaia methodology [Wool+00,Zamb+03] for agent-oriented analysis and design has been used to specify in detail the functionalities and the communication of AMEIM platform. Agent behavioral patterns in Gaia are specified using roles. A role can be viewed as an abstract description of an entity's expected function. In Figure 4 the CA's role is specified in Gaia.

Role Schema: Contribution Agent (CA)

Description:

CA is responsible for acquiring raw data or files from external sources, as sensors, local or remote databases and directories. It can realize both pull and push strategies for this function. Also, it is responsible for reviewing and validating the captured data, exploiting decision making capabilities. CA transforms the data acquired in a common, context-rich format.

Protocols and Activities:

MessageReceiver, ReceiveParameters, CaptureData, ContributeData, DataValidation, ReceiveParameters, SendData

Permissions :

Reads input measurements, files, sensor data
Reads BehaviourParameters

Responsibilities:

Liveness: CA = MessageReceiver, ReceiveParameters, [[CaptureData || ContributeData]*, DataValidation, SendData]*

Safety: BehaviourParameters ≠ null

Figure 4. The Contribution Agent Role using Gaia

In a similar way, the Roles of DMA, DA, and GUIA have been specified. Using the Gaia specifications for agent system analysis and design, the agent protocols, interactions and services have been defined. In Figure 5, the Agent Model and the Acquaintance Model are presented.

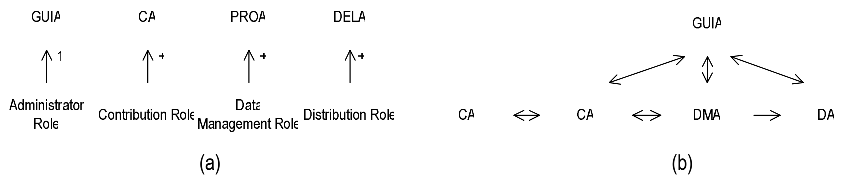

Figure 5. (a) The Gaia Agent Model, and (b) the Acquaintance Model of the AMEIM system

4.3 Agent communication for information flow

As defined in Section 3, agent communication in AMEIM ensures the propagation of information through the system's clusters. Agent communication in AMEIM has been specified in detail using the Agent-Object-Relationship Modelling Language (AORML) [Wagn03]. Figure 6 presents the external AORML diagram of the system. On start up, only the GUIA is created and active. GUIA launches the system's user interface, gets the system parameters from the user, launches at run-time all CAs, DMAs and DAs and, ultimately, starts the system process.

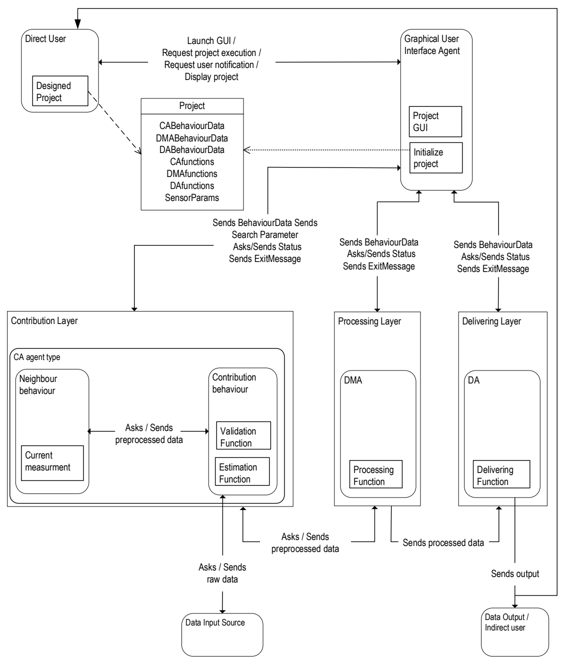

Figure 6: External AORML model for AMEIM

5. AMEIM software development

AMEIM has been developed using the Java Agent Development Environment (JADE). JADE is a framework supporting agent-based development in Java, in a seamless way [Bell+03a, Bell+03b]. In addition, an ontology has been developed using Protégé-2000, for describing all agent concepts and predicates of the system. AMEIM's implementation confronts to the FIPA specifications for agent development and communication [FIPA02].

Figure 7 presents a typical message communicated by AMEIM agents for exchanging environmental information. It is structured using FIPA's Agent Communication Language. The system's GUI is depicted in Figure 8. Through the GUI, the user may customize agent behavior and add (or terminate) AMEIM agents.

```
(inform
  :sender (agent-identifier :CA1)
  :receiver (set (agent-identifier :DMA1))
  :content
    "((SendPREADataToPROAMessage
     (PREADataToPROA
       :TypeOfData Sensor
       :Data "<value>28.0</value>
              <validationTag>VALID</validationTag>
              <level>Low</level>")))
  :language fipa-sl
  :ontology AgentSystem)
```

Figure 7: A typical agent message communicated by a CA to a DMA

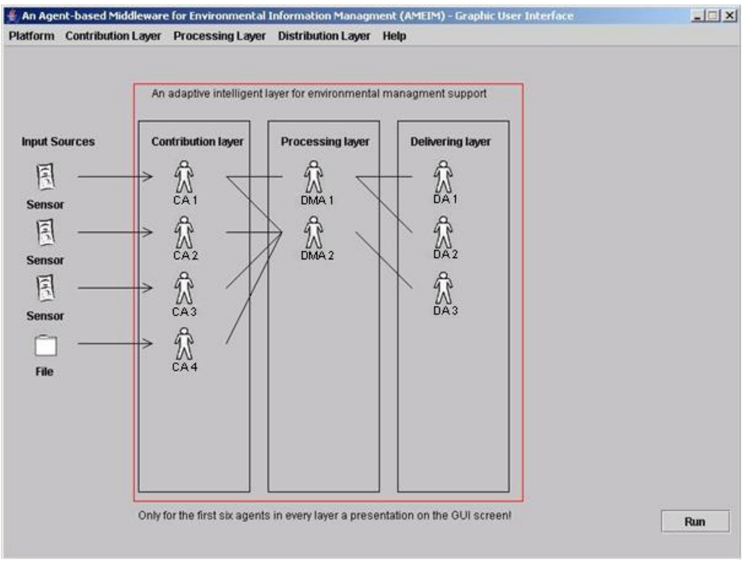

Figure 8: AMEIM's graphical user interface.

6. Discussion

The designed MAS is able to capture data from several external sources and to validate the incoming data. Invalid data can be estimated by the AMEIM and for this purpose the PREA is able to use the data from other external sources supported by other PREAs specified as neighbours.

The MAS supports two modes of operation for capturing different kinds of external sources. The message passing in the system is asynchronous, but with the help of behaviours the system is able to simulate a behaviour which works in cycles and thus it can simulate a synchronous behaviour if needed.

The AMEIM supports the use of several kinds of formats except for formats that use special characters not supported by strings in Java.

The AMEIM ver. 1.0 shows that a MAS can be used as a middleware to support the capturing of environmental information and to present the data in the desired way so that the direct user can implement all kind of transformations and use all kind of data through the function classes. The AMEIM handles all the data without knowing the exact data format which shows the functionality of the system that is able to reach its goals without such information.

The AMEIM shows that agents can handle such tasks and explains why agents are a good choice for such a design. Even if one agent crashes, his crash does not affect the system. This structure of agents shows why a MAS has advantages against a centralized architecture or single agent system which would crash if one part is missing. So in further version of the AMEIM it is important to implement a routine which guarantees that a crashing agent has no effect on the whole MAS. But this implementation can easily be done and needs only some changes in the ReceiverBehaviour and the function classes to implement such functionality.

In general, the AMEIM shows its ability to support environmental institutes in the autonomous capturing, validating and presenting of data in the desired way. The AMEIM needs improvements to support a broader field of applications but the AMEIM and the diploma thesis show that a MAS can handle such tasks and is able to support environmental institutes in further versions which are more user-friendly and have more functionality than the version 1.0.

References

[Atha+03] I. N. Athanasiadis, P. A. Mitkas, G. B. Laleci, and Y. Kabak. Embedding data-driven decision strategies on software agents: The case of a multi-agent system for monitoring air-quality indexes, In R. Jardim-Goncalves, J. Cha, and A. Steiger-Garcao (eds.), Concurrent Engineering: The Vision for the Future Generation in Research and

Applications, vol. 1: Enhanced Interoperable Systems, Madeira, Portugal, Balkema Publishers, July 2003, pp. 23-30.

[Atha+05] Athanasiadis, I.N.; Milis, M.; Mitkas, P.A.; Michaelides, S. Abacus: A multi-agent system for meteorological radar data management and decision support, International Symposium on Environmental Software Systems (ISESS-05), Sesimbra, Portugal May, 2005.

[AtMi04a] I. N. Athanasiadis and P. A. Mitkas, An agent-based intelligent environmental monitoring system. Management of Environmental Quality, 15 (3), pp. 238-249, 2004.

[AtMi04b] Athanasiadis, I. N.; Mitkas, P.A. Supporting the Decision-Making Process in Environmental Monitoring Systems with Knowledge Discovery Techniques, In H. Voss, M. Wachowicz, S. Dzeroski A. Lanza (eds), Proc. of the Knowledge-based Services for the Public Sector Symposium (Workshop III: Knowledge Discovery for Environmental Management), Bonn, Germany, KDnet, June 2004, pp.1-12.

[AtMi04c] I. N. Athanasiadis and P. A. Mitkas. Applying agent technology in environmental management systems under real-time constraints, In C. Pahl, S. Schmidt, A. E. Rizzoli, and A. Jakeman (eds), International Environmental Modelling and Software Society 2004 Congress: Complexity and Integrated Resources Management, Osnabrueck, Germany, June 2004, vol.2, pp. 531-536, iEMSs: Manno, Switzerland. [Best student paper and presentation prize].

[Bell+03a] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, JADE Programmer's Guide, ver. JADE 3.0, 2003.

[Bell+03b] Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. JADE - A white paper. EXP in search of innovation 3 (3), 2003), pp. 6-19.

[Dan+03] Dance, S., Gorman, M., Padgham, L., and Winikoff, M. An evolving multi agent system for meteorological alerts. In Proc. of the 2nd international joint conference on Autonomous Agents and Multiagent Systems, AAMAS-03 (2003), ACM Press, pp. 966-967.

[EC99] European Commission. Public sector information: a key resource for Europe - Green Paper on public sector information in the information society, COM(98)585, January 1999. Available online at: ftp://ftp.cordis.lu/ pub/econtent/docs/gp_en.pdf.

[Etsi95] Etzioni, O., and Weld, D. S. Intelligent agents on the internet: Fact, Fiction, and Forecast. IEEE Expert: Intelligent Systems and Their Applications 10, 4 (1995), 44-49.

[FIPA02] FIPA. ACL Message Structure Specification. Doc. No. SC00061 G, Foundation of Physical Intelligent Agents, Geneva, Switzerland, 2002.

[Fell+03] Felluga, B., Gauthier, T., Genesh, A., Haastrup, P., Neophytou, C., Poslad, S., Preux, D., Plini, P., Santouridis, I., Stjernholm, M., and Wuertz, J. Environmental data exchange for inland waters using independed software agents. Report 20549 EN, Institute for Environment and Sustainability, European Joint Research Centre, Ispra, Italy, April 2003.

[FrGr96] Franklin, S., and Graesser, A. Is it an agent, or just a program?: A taxonomy for autonomous agents. In Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (1996), Springer-Verlag, pp.21-35.

[Günt98] Günter, O. Environmental Information Systems. Springer, 1998.

[PaRa01] Page, B.; Rautenstrauch, C. Environmental Informatics – Methods, Tools and Applications in Environmental Information Processing, In Rautenstrauch, C., and Patig, S., (Eds.) Environmental Information Systems in Industry and Public Administration. Idea Group Publishing, 2001, pp. 2-12.

[Purv+03] Purvis, M., Cranefield, S., Ward, R., Nowostawski, M., Carter, D., and Bush, G. A multi-agent system for the integration of distributed environmental information. Environmental Modelling & Software 18 (2003), 565-572.

[Sway03] Swayne, D. Applying computer research to environmental problems. Environmental Modelling & Software 18 (2003), 485–486.

[Wagn03] G. Wagner: The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behaviour , Informations Systems 28:5 (2003), pp. 475-504.

[WoJe95] Wooldridge, M., and Jennings, N. R. Intelligent agents: Theory and practice. The Knowledge Engineering Review 10, 2 (1995), 115-152.

[Wool+00] M. Wooldridge, N. R. Jennings, and D. Kinny: The Gaia Methodology for Agent-Oriented Analysis and Design, Journal of Autonomous Agents and Multi-Agent Systems pp.285-312, 2000.

[Zamb+03] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering Methodology, 12(3):317-370, 2003.