Georgakarakou, C.E. & Economides, A.A. : **Intelligent Buildings from agents point of view**. *WSEAS Transactions on Computers*, Issue 10, Vol. 5, pp. 2276-2284, October 2006.

Intelligent Buildings from Agents point of view

CHRYSANTHI E. GEORGAKARAKOU AND ANASTASIOS A. ECONOMIDES
Information Systems Department
University of Macedonia
156, Egnatia Street, Thessaloniki 54006
GREECE
georgak@uom.gr, economid@uom.gr

Abstract: - The main objective of this paper is to evaluate Intelligent Buildings' (IBs) previous research activity from agents' point of view. For the above goal a meta-research is transacted over selected literature related to IBs' scientific area. A specific methodology and a statistical processing procedure are utilized in order to achieve the meta-research aims. The parameters over which the meta-research is conducted are provided through an overview of all the basic aspects of software agent technology that we judge important to include. The main objective of this process is to provide researchers with essential and relevant data promoting future theoretical investigations and stimulating their efforts.

Key-Words: - Intelligent Buildings, Agents, Meta-research Evaluation.

1 Introduction

The term "intelligent building" made its first appearance as long ago as in November 1985 in a popular trade magazine which carried an article showing how steel framing and cellular steel flooring had contributed to building intelligence [1]. Since then, a large volume of literature has been produced focusing on this term. Many definitions have been proposed about the concept of intelligent buildings (IBs). Generally speaking, intelligent building is a concept according to which a computerized scheme regulates building components, utilities, electrical circuits, and HVAC (heating, ventilating, and air-conditioning) systems so as to monitor building functions, security, energy consumption, and provide a comfortable environment to the building's residents [2]. To be more precise, intelligent buildings are composed of numerous sensors, effectors and control units interconnected in such a way as to effectively form a machine [3]. Actually, an intelligent building constantly adapts itself by learning from its users and takes actions to control the effectors of the building. By doing so it gradually learns what a user's behaviour is and adapts to it [4].

The notions of agents and multi-agent systems (MAS) are very popular in the intelligent building domain. Much of the prior work in intelligent lighting control involves building control systems that focus on HVAC (heating, ventilation, and air-conditioning), security or other aspects of building management. Several groups have examined the use of multi-agent systems (MAS) for building control [5]. As the environment of an IB is very complex (inaccessible, non-deterministic, non-episodic, dynamic and continuous) [6], a good solution for controlling it is multi-agent systems. If a problem domain is particularly a complex, large, or unpredictable, then the only way it can reasonably be addressed is to develop a number of functionally specific and (nearly) modular components (agents) that are specialized at solving a particular problem aspect [7]. In MASs, applications are designed and developed in terms of autonomous software entities (agents) that can flexibly achieve their objectives by interacting with one another in terms of high-level protocols and languages [8]. A MAS can be defined as a collection of, possibly heterogeneous, computational entities, having their own problem solving capabilities and which are able to interact among them in order to reach an overall goal [9]. More details about MAS, including its characteristics, can be found in [10].

In this paper, we conduct a meta-research evaluation of the previous research activity on agents in IBs. By performing an overview of software agents' technology, we derive a set of parameters-criteria based on which, we judge previous research activity. The judgment is accomplished in terms of percentages that depict the magnitudes that contemporary research activity studies the specific parameters of agent technology. First of all, we describe the methodology that we use. In order to select the sets of parameters-criteria we use an agent technology framework (ATF). The full presentation of the ATF that describes basic aspect of agent technology is provided in the Appendix of the paper. At the end we enumerate the results of the study. This study is based on theoretical investigation focusing on promoting theoretical work in future and constitutes a methodological exploration for synthesizing inhomogeneous literature.

2 Methodology

As discussed earlier, this study performs a meta-research of previous research activity on agents in IBs. A meta-research is a study on research and an analysis of analysis and it can be defined as the synthesis of primary research results into more general conclusions [11]. As meta-research method we use Rogers, E. M. (1981) propositional inventory with some divergence from the actual method definition. Propositional inventory is the synthesis of general conclusions from research where the original data is not available and hence where only written conclusions from each of the primary studies are available to the meta-researcher. The key methodological issue when conducting a propositional inventory is to closely define the parameters of the research to be included in the analysis prior to beginning the literature search [11]. The propositional inventory uses the categorizing of discrete elements such as variables, methods or findings in a specific study. Once similar studies have been broken down, the strengths and weaknesses of the body of the research can be understood as a whole and gaps in knowledge can be identified [12].

The body of literature over which the meta-research is conducted was identified by doing a review of articles' communication abstracts and titles firstly and articles' body secondly searching for relevant work under the key words, agent(s), multi-agent system(s) or MAS concurrently with the key words, intelligent building(s), (smart or intelligent) home(s) or house(s) or room(s) or dorm(s), ambient intelligent environment(s), ambience, inhabited intelligent environment(s), intelligent interactive home(s), human home interaction(s), interactive home(s), adaptive environment(s), home automation, ubiquitous environment(s), building automation. In addition other key words were taken into account but with low priority that are relevant with the intelligent buildings' area from a broader view, such as intelligent work space(s), elder independence, (home) appliances, etc. The reference lists from the obtained works were considered to identify additional studies in the area of the agent technology in IBs. As relevant studies were identified, their references also were used to expand the body of research that was examined. Using this approach, a total of 72 literature sources in majority conference and journal papers, were identified as potentially focused on some aspect of agent technology in IBs.

We synthesize the Agents Technology Framework (ATF) that consists of the parameters over which the meta-research was conducted. We consider the ATF as an aggregation of sets of variables. The variables that constitute each set are associative with each other under the prism of a concept that determines the specific set. The concepts that determine the specific sets of variables comprise all the different aspects of the agent technology that we judge important to include into our meta-research evaluation of IBs. An important point that we must underline is that the evaluation of a variable is performed only in comparison with the other relative variables of the same set and not taking into account any other variables of the other sets. We provide the ATF together with a brief overview of the basic aspects of multi-agent systems (MAS) and software agents' technology in the Appendix of the paper, classifying the agents' technology across 10 dimensions that correspond to 10 sets of variables.

We estimate the level of previous research activity (the selected body of literature referred above) in terms of percentages that correspond to the % percent that a specific variable is ignored by the literature body and to the % level of variable performance[1] in terms of moderate, good, very good and excellent[2]. In other words, we try to provide the level of coverage of a specific variable by the body of literature in terms of 'ignore', 'moderate', 'good', 'very good' and 'excellent'. The study provides an overall view of the literature level of relevance to specific aspects of agent technology without presenting separately in details the studies that correspond to each result or categorizing the literature body according to the results mentioning separately each distinct study. We are not so interested in providing the volume of previous research activity reasoning our results with specific explanations provided by its separate article by the literature body. We prefer to give a general point of view of previous research activity in terms of relevance percentages.

In order to carry out our research aims, we propose an explicit evaluation model. The proposed evaluation model, utilizing the above-mentioned ATF (in Appendix), puts the evaluation of previous research activity into practice in a simple but reliable way.

[1] The term variable performance is defined below.
[2] The terms 'ignore', 'moderate', 'good', 'very good' and 'excellent' are also defined below.

Evaluation model's considerations: i) it uses SPSS 11.0 for Windows in order to conduct the statistical processing of the ATF; ii) it uses qualitative variables, which registration became accordingly with the principle of registration of qualitative data in scale of order. Each answer that corresponds to the question of each variable is coded giving a number as code and then the codes are registered in the cells; iii) for the purposes of codes' definition we define the variable performance as 'the degree of positive answer and cover in the question that corresponds to the variable based on the ATF that has been defined in the requirements of the particular research'; iv) we define five codes:

Code No 1: <<Ignore>> it is reported in that the particular variable is not examined by the particular article as autonomous significance. It is not taken into consideration the fact that from the total estimate of the article it can come out some arbitrary conclusion with regard to the specific variable because this would lead to not valid and reasonably implicit estimate.

Code No 2: <<Moderate>>, Code No 3: <<Good>>, Code No 4: <<Very good>> and *Code No 5: <<Excellent>>* is reported in the fact that the 'variable performance' of the particular variable, based on the opinions, the estimate and the conclusions of researchers of the specific article as well as the opinions and conclusions of other articles that are taken into consideration in the specific article, is satisfied at moderate degree, good degree, very good degree and excellent degree correspondingly; v) each set of variables of the ATF is examined separately by a sector of Statistics that is reported in the Descriptive Statistics where the examination rates (Frequencies) of the various variables per set of variable of the framework were studied.

3 Meta-research Results

The results of the conducted meta-research are presented in terms of frequency tables. The variables (see Appendix) that are totally ignored by the previous research activity (100% percentage of ignorance) are not presented in the results.

According to our findings, some research activity considers the environment of the intelligent house as an intelligent agent that is decomposed into lower level agents which are responsible for subtasks within the home. It is rare the case that an intelligent agent is used as a centralized agent which communicates with all other devices present in home in order to achieve the intelligent home's requirements. When research is referred to a single agent, it is usually used for specific purpose and probably cooperates with other agents. Both of the cases referred above are included in the content of the single agent system parameter. Table 1 depicts the percentages of ignorance and degrees of performance for MAS and single agent system. The previous research activity prefers the use of the multi-agent system approach for the intelligent buildings. Since multi-agent systems are composed of many agents that work together to accomplish a specific goal, there is the possibility to develop very simple agents, which still achieve the goals by working together if necessary. Using multi-agent systems, complex problems can be divided into more simple sub-problems making the complexity of particular problems controllable [13].

MAS: Ignore: 32.9 %, Moderate: 13.7 %, Good: 24.7 %, Very good: 28.8 %
Single agent system: Ignore: 87.7 %, Moderate: 2.7 %, Good: 8.2 %, Very good: 1.4 %

Table 1. Percentage of ignorance and degrees of performance for the agent system parameters.

The majority of previous research activity does not advert separately to distinctive agent types such collaborative agents, interface agents, mobile agents, information/internet agents, reactive agents, hybrid agents, smart Agents, and heterogeneous agent systems. Researchers prefer to describe in detail the agent-based systems they use and focus on their properties and usually they give their own labels to the agent types that they refer to. However, we observe some research activity that referred to the above distinct agent types. Table 2 depicts the percentages of ignorance and degrees of performance of each agent type parameter. Intelligence constitutes the key property of an IB, so the smart agents' type is the most popular. A remarkable body of previous research activity gives a lot of consideration to the embedded agents[3].

Smart agents: Ignore: 63 %, Moderate: 12.3 %, Good: 16.4 % Very good: 8.2 %
Mobile agents: Ignore: 79.5 %, Moderate: 6.8 %, Good: 5.5 % Very good: 6.8 %, Excellent: 1.4 %

[3] An embedded-agent is simply an autonomous intelligent control entity built into a device [14].

Collaborative agents: Ignore: 89 %, Moderate: 4.1 %, Good: 5.5 % Very good: 1.4 %
Interface agents: Ignore: 97.3 %, Moderate: 1.4 %, Good: 1.4 %
Reactive agents: Ignore: 97.3 %, Good: 2.7 %
Information/ Internet agents: Ignore: 98.6 %, Moderate: 1.4 %
Hybrid agents: Ignore: 98.6 %, Good: 1.4 %

Table 2. Percentage of ignorance and degrees of performance for the agent typology parameters.

It was observed that most research activity focused on layered architectures with the reactive agents' architectures to follow. Table 3 depicts all the agent architectures' percentages of ignorance by previous research activity as well as their degrees of performance.

Layered architectures: Ignore: 74 %, Moderate: 5.5 %, Good: 12.3 % Very good: 8.2 %
Reactive agents: Ignore: 80.8 %, Moderate: 12.3 %, Good: 5.5 %
Logic based agents: Ignore: 97.3 %, Moderate: 2.7 %
Belief-desire-intention (BDI): Ignore: 98.6 %, Very good: 1.4 %

Table 3. Percentage of ignorance and degrees of performance for the agent architecture parameters.

Communication protocols are common in IBs. Table 4 depicts the percentages of ignorance by previous research activity as well as the degrees of performance for communication protocols. Communication protocols offer simplicity (i.e. they possess strict and formal nature so they are easy to understand), interoperability and verifiability (i.e. it is possible to formally analyze a communication protocol due to its transparency). In addition to the above advantages, "communication protocols" was the first technique that was really used in communication. The evolving languages appeared at a later stage [15].

Communication protocols: Ignore: 69.9 %, Moderate: 24.7 %, Good: 5.5 %

Table 4. Percentage of ignorance and degrees of performance for the agent communication approach parameters.

When research efforts refer to agent communication languages, in majority they do not focus on advancing or developing existing or new agent communication languages tailored to the IBs applications. Researchers are interested in existing standards without excluding the fact that there are some remarkable efforts for developing a new ACL dedicated to a specific application. KQML is the most popular ACL referred by previous research activity. Table 5 illustrates the percentages of ignorance and the degrees of performance for KQML, FIPA-ACL, KIF and ICL. KQML is the most popular ACL because of its inherent flexibility, ease of implementation, and consequent availability of tools [16].

KQML: Ignore: 80.8 %, Moderate: 13.7 %, Good: 4.1 % Very good: 1.4 %
FIPA-ACL: Ignore: 98.6 %, Moderate: 1.4 %
KIF: Ignore: 98.6 %, Moderate: 1.4 %
ICL: Ignore: 98.6 %, Good: 1.4 %

Table 5. Percentage of ignorance and degrees of performance for the ACL parameters.

CORBA constitutes a popular agent transportation mechanism for IBs. Table 6 illustrates the percentages of ignorance and the degrees of performance for CORBA. CORBA constitutes a good solution for an IB network because it allows each intelligent environment to have different devices, computer equipment and software systems without this heterogeneity being a problem. CORBA has also other features that can be useful for IBs such as software reuse and location transparency [17].

CORBA: Ignore: 90.4 %, Moderate: 6.8 %, Good: 1.4 % Very good: 1.4 %

Table 6. Percentage of ignorance and degrees of performance for the agent transportation mechanism parameters.

Regarding ontology languages and editors small concern is given to LOOM. Table 7 illustrates the percentages of ignorance and the degrees of performance for LOOM. As LOOM is almost ignored in IBs research activity, we do not separate it from the other ontology languages and editors.

LOOM:

Ignore: 98.6 %, Moderate: 1.4 %

Table 7. Percentage of ignorance and degrees of performance for the ontology language and editor parameters.

Table 8 depicts the percentages of ignorance and the degrees of performance for Java, April, C, Prolog, C++, Lisp, and Python. Java is extensively present in intelligent building research activity. This may be due to the fact it facilitates platform independent applications.

Java: Ignore: 60.9 %, Moderate: 21.9 %, Good: 13.7 % Very good: 4.1 %
April: Ignore: 91.8 %, Moderate: 5.5 %, Good: 2.7 %
C: Ignore: 93.2 %, Moderate: 5.5 %, Good: 1.4 %
Prolog: Ignore: 95.9 %, Moderate: 2.7 %, Good: 1.4 %
C++: Ignore: 97.3 %, Moderate: 2.7 %
Lisp: Ignore: 98.6 %, Moderate: 1.4 %
Python: Ignore: 98.6 %, Moderate: 1.4 %

Table 8. Percentage of ignorance and degrees of performance for the languages for constructing agent-based system parameters.

Table 9 depicts the percentages of ignorance and the degrees of performance for Jade, JAFMAS, ZEUS, RETSINA, JATLite, MADKIT, and OAA. As the differences in the percentages of ignorance are very small we can not make any distinction about the use of one platform instead of another platform in IBs applications.

Jade: Ignore: 94.5 %, Moderate: 4.1 %, Good: 1.4 %
JAFMAS: Ignore: 95.9 %, Moderate: 4.1 %
Zeus: Ignore: 97.3 %, Moderate: 2.7 %
RETSINA: Ignore: 97.3 %, Moderate: 2.7 %
JATLite: Ignore: 98.6 %, Moderate: 1.4 %
MADKIT: Ignore: 98.6 %, Moderate: 1.4 %
OAA: Ignore: 98.6 %, Moderate: 1.4 %

Table 9. Percentage of ignorance and degrees of performance for the tools and platforms parameters.

Regarding AOSE methodologies little interest is given to Gaia. Table 10 illustrates the percentages of ignorance and the degrees of performance for Gaia. The percentage that previous research activity is referred to Gaia is very small, so we can not make any conclusion about Gaia's application in IBs instead of other methodologies. We can mention that Gaia is an easy software development methodology but it only covers the phases of analysis and design.

Gaia: Ignore: 98.6 %, Moderate: 1.4 %

Table 10. Percentage of ignorance and degrees of performance for the AOSE methology parameters.

4 Conclusion

We performed an evaluation of previous research activity on agent technology applied to IBs. We utilized an agent technology framework (ATF) that provided us with the necessary set of variables which were used by an explicit evaluation model that put, through statistical processing, the evaluation activity into practice. We also described the methodology that we used in order to conduct our meta-research. We tried to estimate the level of previous research activity in terms of percentages that correspond to the % percent that a specific variable is ignored by the literature body and to the % level of "variable performance" in terms of moderate, good, very good and excellent. The term "variable performance" is the degree of positive answer and cover in the question that corresponds to the variable based on the ATF. In other words, we tried to provide the level of coverage of a specific variable by the body of literature in terms of 'ignore', 'moderate', 'good', 'very good' and 'excellent'. We hope that this may become an accessory step for promoting theoretical work in future.

References:

[1] S. Agarwal, and B. Arch., "Intelligence Vs. Automation: A Review", IT in Built Environment, International Conference at IIT Kharagpur, 2002

[2] R. Giladi, "Heterogeneous Building Automation and IP Networks Management", ICDCS Workshops 2004, pp. 636-641.

[3] S. Sharples, V. Callaghan, and G. Clarke (1999), "A multi-agent architecture for intelligent building

sensing and control", Sensor Review, Vol. 19, No. 2, 1999, pp.135-140.
[4] J. Trindler1, R. Zwiker1, U. Rutishauser, R. Douglas and J. Joller1, "Discovery of Logical Structure in a Multisensor Environment Based on Sparse Events Discovery of Logical Structure in a Multisensor Environment Based on Sparse Events", AI*IA 2003 Workshop on Ambient Intelligence, 8th National Congress of Italian Association for Artificial Intelligence.
[5] J. S. Sandhu and A. M. Agogino and A. K. Agogino, "Wireless Sensor Networks for Commercial Lighting Control: Decision Making with Multi-agent Systems", Proceedings of the AAAI-04 Workshop on Sensor Networks, San Jose, CA, 2004.
[6] U. Rutishauser, and A. Schafer, "Adaptive Building Automation. A multi-Agent approach", Research project, 7/20/02. http://www.easc.ch/aha/Docu/aha1_1_0.pdf
[7] K. P. Sycara, "Multi-agent Systems", AI magazine, Vol.19, No. 2, 1998, Intelligent Agents Summer.
[8] F. Zambonelli, N. Jennings, and M. Wooldridge, "Developing Multiagent Systems: the Gaia Methology", ACM Transactions on Software Engineering and Methodology, Vol.12, No. 3, 2003.
[9] Oliveira, E., K. Fischer, and O. Stepankova, "Multi-agent systems: which research for which applications", Robotics and Autonomous Systems, Vol. 27, 1999, pp. 91-106.
[10] N.R. Jennings, K. Sycara, and M. Wooldridge, "A Roadmap of Agent Research and development", Autonomous Agents and Multi-Agent Systems Journal (Eds.), Boston: Kluwer Academic Publ, Vol. 1, No. 1, 1998, pp. 7-38.
[11] E. H. Rogers, "Methology for meta-research", Presidential address at the conference of the International Communication Association, 1981.
[12] A. Hollifield, "Crossing borders: Media management research in a transnational market", Environment, Journal of Media Economics, Special Issue: Transnational Media Management, Vol. 14, No. 3, 2001, pp. 133-146.
[13] E. Steegmans, P. Rigole, T. Holvoet, and Y. Berbers, "Intelligent Buildings: A Multi-agent System Approach", Artificial Intelligence and Applications Proceedings (Hamza, M.H., ed.), 2003, pp. 771-776.
[14] V. Callaghan, G. Clarke, A. Pounds-Cornish, and S. Sharples, "Buildings as Intelligent Autonomous Systems: A Model for Integrating Personal and Building Agents", 6th International Conference on Intelligent Autonomous Systems, Venice, 2000.
[15] D. Bussink, "A Comparison of Language Evolution and Communication Protocols in Multi-agent Systems", 1st Twente Student Conference on IT, Track C - Intelligent_Interaction, 2004, http://referaat.ewi.utwente.nl/
[16] "ALFEBIITE: A logical framework for ethical behaviour between infohabitants in the information trading economy of the universal information ecosystem", project number: IST–1999–10298. http://alfebiite.ee.ic.ac.uk/docs/Deliverables/D0.pdf
[17] A. Reyes, A. Barba, V. Callaghan, and G. Clarke, "The Integration of Wireless, Wired Access and Embedded Agents in Intelligent Buildings", 5th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Florida, 2001.
[18] S. Franklin, and A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", Proc. Third International Workshop on Agent Theories, Architectures and Languages, Springer, 1996.
[19] Camelia Chira. Software Agents, IDIMS Report, 2/21/03,
http://pan.nuigalway.ie/code/docs/agents.pdf
[20] J. M. Bradshow. "An Introduction to Software Agents, Software Agents", Cambridge, MIT Press, 1997.
[21] H. S. Nwana, "Software Agents: An Overview", Knowledge Engineering Review, Vol. 11, No. 3, 1996, pp. 1-40.
[22] M. Wooldridge, "Intelligent Agents", The MIT Press, 1999.
[23] M. Wooldridge, and N. R. Jennings, "Intelligent agents: theory and practice", The Knowledge Engineering Review, Vol. 10, No. 2, 1995, pp. 115-152.
[24] M. Wooldridge, "Agent-based computing", Interoperable Communication Networks, Vol. 1, No. 1, 1998, pp. 71-97.
[25] V. Kostakos, and C. Taraschi, "Agents", 5/7/01. http://www.cs.bath.ac.uk/~vk/files/agents.pdf
[26] G. Weiß, "Agent orientation in software engineering", Knowledge Engineering Review, Vol. 16, No. 4, 2002, pp. 349–373.
[27] OMG Agent Working Group, Agent Technology, green paper, OMG Document ec/8/1/00, (1).
[28]
http://www.omg.org/gettingstarted/corbafaq.htm
[29] www.omg.org
[30] The CORBA Object Group Service (1998). A Service Approach to Object Groups in CORBA, THESES N° 1867. http://lsrwww.epfl.ch/OGS/thesis/
[31] A. Bracho, A. Matteo, and Ch. Metzner, "A taxonomy for comparing distributed objects

technologies”, CLEI Electronic Journal, Vol. 2, No. 2, 1999.
[32] DCOM Technical Overview (1996). Microsoft Corporation. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomtec.asp
[33] Tuukka Vartiainen. Java Beans and Enterprise Java Beans. http://www.cs.helsinki.fi/u/campa/teaching/tukka-final.pdf
[34] H. Nwana, and M. Wooldridge, “Software Agent Technologies”, BT Technology Journal, Vol. 14, No. 4, 1996, pp. 68-78.
[35] Serenko, and B. Deltor, “Agent Toolkits: A General Overview of the Market and an Assessment of Instructor Satisfaction with Utilizing in the Classroom”, Working Paper #455, 2002, http://www.business.mcmaster.ca/msis/profs/detlorb/nserc/McMaster_Working_Paper_455.pdf
[36] Agent-Oriented Software Engineering, “Research Area Examination”, 3/31/05. http://www.deg.byu.edu/proposals/ResearchAreaExamMuhammedJM.pdf
[37] C.A. Iglesias, M. Garijo, and J.C. Gonzalez, “A Survey of Agent-Oriented Methologies”, Intelligent Agents V, Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98), Lecture Notes in Artificial Intelligence. Springer-Verlag, Vol. 1555, 1999, pp. 317-330.
[38] L. Cernuzzi, M. Cossentino, and F. Zambonelli, “Process Models for Agent-Based Development”, International Journal on Engineering Applications of Artificial Intelligence (EAAI), Elsevier, 2004, http://www.pa.icar.cnr.it/~cossentino/paper/eaai_zambonelli_draft.pdf
[39] M. Wooldridge, and P. Ciancarini, “Agent-Oriented Software Engineering: The State of the Art”, First Int. Workshop on Agent-Oriented Software Engineering, P. Ciancarini and M. Wooldridge (ed.), Berlin: Springer-Verlag, Vol. 1957, 2000, pp. 1-28.
[40] AOSE Methologies http://www.science.unitn.it/~recla/aose/

APPENDIX - Agents Technology Framework (ATF)

We classify the agents’ technology across 10 dimensions that correspond to 10 sets of variables.

1 Agent Systems

Jennings, Sycara and Wooldridge (1998) state that an agent-based system is a system in which the key abstraction used is that of an agent. An agent-based system may contain one or more agents. There are cases in which a Single agent system is appropriate. However, the Multi-agent system (MAS) is arguably more general and more interesting from a software engineering standpoint [10].

In our ATF, a variable is assigned to every “Agent System”. This variable describes to which extend this “Agent System” is used by the IBs applications.

2 Agent Typology

Agents might be classified according to the tasks they perform, their control architecture, the range and sensitivity of their senses, the range and effectiveness of their actions, or how much internal state they possess [18]. There are several classification schemes or taxonomies proposed in the agent research community. The following three are well acknowledged [19]: i) Gilbert’s scope of intelligent agents [20], ii) Nwana’s primary attribute dimension typology [21] and iii) Franklin and Graesser’s agent taxonomy [18]. A typology refers to the study of types of entities and there are several dimensions to classify existing software agents [21]. We adopt Nwana’s agent typology [21], [19]: Collaborative agents, Interface agents, Mobile agents, Information/Internet agents, Reactive agents, Hybrid agents, Smart Agents, and Heterogeneous agent systems.

In our ATF, a variable is assigned to every “Agent Type”. This variable describes to which extend this “Agent Type” is used by the IBs applications.

3 Agent Architectures

‘An agent architecture is essentially a map of the internals of an agent — its data structures, the operations that may be performed on these data structures, and the control flow between these data structures’ [22]. Three classes of agent architectures can be identified [23]: i) deliberative or symbolic architectures are those designed along the lines proposed by traditional, symbolic AI; ii) reactive architectures are those that eschew central symbolic representations of the agent’s environment, and do not rely on symbolic reasoning; and iii) hybrid architectures are those that try to marry the deliberative and reactive approaches [24]. We adopt Wooldridge’s agent architecture classification [22]: Logic based agents, Reactive agents, Belief-desire-intention (BDI) agents, and Layered architectures.

In our ATF, a variable is assigned to every “Agent Architecture”. This variable describes to which extend this "Agent Architecture" is used by the IBs applications.

4 Agent Communication Approaches

Agents recurrently interact to share information and to perform tasks to achieve their goals [25]. Without communication, different agents cannot know from each other which agent is doing what and how they can cooperate [15]. The two most important Agent Communication Approaches are using: i) Communication Protocols, and ii) Evolving Languages.

In our ATF, a variable is assigned to every "Agent Communication Approach". This variable describes to which extend this "Agent Communication Approach" is used by the agents to communicate in the IBs applications.

5. Agent Communication Languages (ACLs)

A number of languages for coordination and communication have been proposed [26]. Weíβ (2002) distinguishes the most prominent examples of Agent Communication Languages (ACLs) [26]: KQML (Knowledge Query and Manipulation Language), ARCOL (ARTIMIS COmmunication Language), FIPA-ACL (FIPA Agent Communication Language), KIF (Knowledge Interchange Format), and COOL (domain independent COOrdination Language).

Apart from these, several others showing unique properties have been proposed, for instance: ICL (Interagent Communication Language), AgentTalk, CoLa (Communication and coordination Language), TuCSoN (Tuple Centres Spread over Networks), LuCe, STL++ (Simple Thread Language ++) and SDML (Strictly Declarative Modelling Language) [26].

In our ATF, a variable is assigned to every "ACL". This variable describes to which extend this "ACL" is used by the agents in the IBs applications.

6. Agent Transportation Mechanisms

In agent environments, messages should be schedulable, as well as event driven. They can be sent in synchronous or asynchronous modes. The transportation mechanism should support unique addressing as well as role-based addresses. Lastly, the transportation mechanism must support unicast, multicast, and broadcast modes and such services as broadcast behaviour, non-repudiation of messages, and logging [27]. Possible implementations of Agent Transportation Mechanism include [28], [29], [30], [31], [32] and [33]: CORBA (Common Object Request Broker Architecture), OMG (Object Management Group) Messaging Services, JAVA Messaging Service, RMI (Remote Method Innovation), DCOM (Distributed Component Object Model), and Enterprise Java Beans Events.

In our ATF, a variable is assigned to every "Agent Transportation Mechanism". This variable describes to which extend this "Agent Transportation Mechanism" is used by the agents in the IBs applications.

7. Ontology Languages and Editors

Besides an ACL, a common ontology is required for representing the knowledge from various domains of discourse [27]. The ACL remains just syntax without a shared common ontology containing the terms used in agent communication and the knowledge associated with them [34]. According to Weíβ (2002) the most elaborated examples of such languages are the following [26]: Ontolingua and Frame Logic, CLASSIC and LOOM, and CycL.

The most prominent Ontology Specification Languages that are conform to syntactic and semantic Web standards are the following [26]: SHOE (Simple HTML Ontology Extension), XOL (Ontology Exchange Language), OIL (Ontology Inference Layer), and the DAML (DARPA Agent Markup Language) languages DAML-ONT and DAML-OIL.

Three good examples of Ontology Editors for ontology creation and maintenance are the following [26]: Protégé, Webonto, and OntoEdit.

In our ATF, a variable is assigned to every "Ontology Language or Editor". This variable describes to which extend this "Ontology Language or Editor" is used in the IBs applications.

8 Languages for constructing Agent-based systems

Most agent systems are probably written in Java and C/C++. Apart from these standard languages, several prototype languages for implementing agent-based systems have been proposed [26]. Weíβ (2002) lists some of the most prominent and best understood prototype languages following the agent oriented paradigm [26]: AGENT-0, Concurrent MetateM, AgentSpeak(L), 3APL, and ConGolog.

Other examples of languages following the agent-oriented programming paradigm are April (Agent PRocess Interaction Language), MAIL/MAI2L (Multiagent Interaction and ImplementationLanguage), and VIVA.

Nwana and Wooldridge (1996) classify constructing agent application languages according to a typology that includes the following agent types [34]:

- Collaborative agents: the actor language Actors and the agent-oriented programming languages Agent-0 and Placa
- Interface, Information and mobile agents: the scripting languages TCL/Tk, Safe-TCL, Safe-Tk, Java, Telescript, Active web tools, Python, Obliq, April and Scheme-48
- Reactive agents: the reactive language RTA/ABLE

However traditional languages are still used to construct agent applications. It is possible to implement agent-based systems in languages like Pascal, C, Lisp, or Prolog. Typically, object-oriented languages such as Smalltalk, Java, or C++ lend themselves more easily for the construction of agent systems [34].

In our ATF, a variable is assigned to every "Language for Constructing Agent-based systems". This variable describes to which extend this "Language for Constructing Agent-based systems" is used for agents' construction in the IBs applications.

9 Tools and Platforms

A number of tools and platforms are available that support activities or phases of the process of agent-oriented software development. While almost all available tools and platforms have their focus on implementation support, some of them do also support analysis, design, and test/debugging activities [26].

Weíβ (2002) makes a list of such tools and platforms separating them into often sited academic and research prototypes and into commercial products for development support. The most prominent ones from the first category are: ZEUS, JADE (Java Agent DEvelopment framework), LEAP (Lightweight Extensible Agent Platform), agenTool, RETSINA, JATLite (Java Agent Template, Lite), FIPA-OS, and MADKIT. Other examples are SIM_AGENT, JAFMAS (Java-based Agent Framework for Multi-Agent Systems), ABS (Agent Building Shell), OAA (Open Agent Architecture), and Agentis. The representative examples of commercial products for developmental support are: AgentBuilder, JACK, Intelligent Agent Factory and Grasshopper [26]. Serenko and Detlor (2002) categorize the available agent toolkits on the market into four major categories [35]:

- Mobile agent toolkits: Concordia, Gossip, FarGo, and IBM Aglets.
- Multi-agent toolkits: MadKit, Zeus, JADE, JATLite, and MAST.
- General purpose toolkits: FIPA-OS, and Ascape.
- Internet agent toolkits: Microsoft Agent, Voyager, and NetStepper.

In our ATF, a variable is assigned to every "Tool or Platform". This variable describes to which extend this "Tool or Platform" is used in the IBs applications.

10 Agent-Oriented Software Engineering (AOSE) Methodologies

Agent researchers have produced methodologies to assist engineers to create agent-based systems. Some researchers have taken agent theory as their starting point while others have taken object techniques as their point of departure or knowledge engineering concepts. Researchers also have tried to assemble methodologies by combining features from different methodologies. Yet other researchers have produced methodologies based on both agent and object technologies [36]. The most popular approaches based on agent and multi-agent technology are the following [26]: Gaia (Generic Architecture for Information Availability), SODA (Societies in Open and Distributed Agent spaces), Cassiopeia, and Aalaadin.

The most popular approaches based on object-oriented technology are the following [26]: KGR, MaSE (Multiagent Systems Engineering), MASSIVE (MultiAgent SystemS Iterative View Engineering), AOAD (Agent-Oriented Analysis and Design), and MASB (Multi-Agent Scenario-Based).

The most popular approaches based on knowledge engineering technology are the following [26]: CoMoMAS (Conceptual Modelling of Multi-Agent Systems) and MAS-CommonKADS (Multi-Agent System CommonKADS).

Other Agent-Oriented Software Engineering (AOSE) methologies are the following: Tropos, Agent-Oriented Analysis and Design, Agent Modelling Technique for Systems of BDI agents, Agent Oriented Methodology for Enterprise Modelling, PASSI (a Process for Agent Societies Specification and Implementation), Prometheus , AOR, ROADMAP, OPM /MAS, Ingenias, DESIRE, AAII methodology, Cooperative Information Agents design, Adept, AUML, ADELFE, MESSAGE /UML, The Styx Agent Methology, SABPO, EXPAND (Expectation-oriented analysis and design), and ODAC [26, 36, 37, 38, 39, 40].

In our ATF, a variable is assigned to every "AOSE Methodology". This variable describes to which extend this "AOSE Methodology" is used in the IBs applications.