International Conference on Automation, Control Engineering and Computer Science (ACECS)
Proceedings of Engineering and Technology – PET
Vol.20 pp.29-34

Comparative study between Multi Agents Systems methodologies according to intelligent embedded systems requirements

MECIBAH Zina[#1], BOUTEKKOUK Fateh[#2]

[#]*Research Laboratory on Computer Science's Complex Systems ReLa(CS)[2], University of Oum El Bouaghi Oum El Bouaghi, Algeria*
[1]mecibah.zina@hotmail.fr
[2] Fateh_boutekkouk@yahoo.fr

***Abstract*— Recently, Intelligent Embedded Systems (IES) are becoming more and more complex and their importance is increasing in our daily life. At present, IES are everywhere: smart vender machine, smart card, security door, router, hub, alarm system, satellite, automobile, IPhone, etc. Despite the ever growing of IES, we can remark a scarcity of design methodologies covering all aspects related to specificities of such systems like autonomy, predictable behavior under hard real time constraints, self-adaptation, energy consumption management, reliability, hardware implementation, etc. On the other hand and in the last years, many Multi Agent Systems (MAS) design methodologies have been developed (more than 80 methodologies). In fact, MAS are mainly used to solve complex problems and have been used successfully in many academic and industrial projects. Thus, MAS may be a good alternative for developing IES. In order to apply MAS in the field of IES efficiently, tuning existing MAS methodologies is a must. The objectives of this paper are twofold, we try firstly to compare between six MAS methodologies that are Roadmap, Passi, Prometheus, RT-Message, O-Mase and ADELFE 3.0 according to some pertinent IES requirements and secondly to decide which is the most appropriate MAS methodology to design IES.**

***Keywords*— Embedded systems, intelligent systems, MAS methodologies, design, comparative.**

I. Introduction

In contrast to traditional Embedded Systems (ESs), nowadays ESs are becoming more complex, more autonomous, more open, more networked and more adaptable (submerged in a dynamical environment which is incompletely specified i.e. the necessary algorithm for solving the problem does not exist). For instance, most of the ESs are connected to internet, use wireless communication, function in dynamic possibly unreachable environments, execute very complex intelligent tasks to help invalid and aged persons in their daily activities. These new features have pushed researchers and ESs specialists to tune some well known methods and paradigms. Consequently a new class of ES called Intelligent Embedded Systems (IES) is occurred.

W.Elmenreich [1] presented a variety of intelligent methods for IES design, among them: bio-inspired computing (Neural Networks, Genetic Algorithms and Neuro-Fuzzy Systems), Soft Computing, Model Checking and Multi-Agent Systems.

On the other hand, it should be noted that the application of MAS to model, simulate and even to synthesis IES is an attractive tendency. Experience from both academia and industry has proved that MAS have been used successfully to design complex, self adaptable and even real time systems.

Currently, there are more than 80 MAS design methodologies. We think that most existing MAS methodologies in their current state are not able to deal with IES specificities, however with some tuning and enhancement, MAS can be very efficient to develop IES.

In all cases we see that we must create a bridge between MAS models and existing well practiced ESs Codesign methodologies and associated tools for hardware/software partitioning.

The progress in hardware technologies will certainly contribute in efficient implementations of IES notably those targeting multicores and reconfigurable architectures like FPGA. Reconfigurable architectures match well dynamic and adaptable IES.

According to literature, four interesting IES design methodologies were reported:

- In 2001: approach proposed by C.Rust et al [2]. This approach allows only the modeling of Intelligent Embedded Real-Time Systems using High-Level Petri Nets. The key objective of this approach is formal verification because it is based on predicate/transition Nets (Pr/T-Nets).
- In 2005: DIAMOND methodology [3] which is an acronym of Decentralized Iterative Approach for Multi agent Open Networks Design, it is mainly used to design embedded MAS. The process is arranged into four different stages. These phases are combined into a spiral life cycle to allow the iterations at the stage of analysis and generic design. The methodology is used to design several projects as the EnvSys project (embedded sensor network for the instrumentation of an underground hydrographic system) and supported by the MASK tools (MultiAgent System Codesign). However, the tool is not available for the moment

ISSN 2356-5608

which makes the use of this methodology very difficult. Also, although the methodology seems robust and complete, we have not enough experimental results to judge its efficiency for IES.

- In 2009: E. Kazanavicius et al [4] proposed a new approach to design real time embedded systems based on multi agent paradigm. The approach proposes a system architecture for embedded systems, it is composed of four layers built on the basis of Jade agent platform. In order to hide the implementation complexity, the authors handled agents as reusable software components. Although, Jade platform provides a good background which facilitates the development of MAS, the approach was not adjusted to design real time embedded systems and indeed we do not well understand this approach.
- In 2011: E. Kazanavicius et al [5] proposed a methodology to design IES covering analysis, design, implementation and deployment phases. Unfortunately, we are not able to use this methodology because there is a scarcity of references.

Throughout this paper, we try to compare between six MAS methodologies which are ROADMAP, Passi, Prometheus, RT-Message, O-Mase and ADELFE 3.0, according to the most pertinent IES requirements and then to choose the most appropriate MAS methodology for IES design. Note that the selection of these methodologies is based on a preliminary analysis of the features of several MAS methodologies. The rest of the paper is structured as follows: section one describes eight criteria of comparison which are used later to compare between the six MAS methodologies. Section two presents an overview of agent oriented methodologies. Section three presents a general description of ROADMAP, Passi, Prometheus, RT-Message, O-Mase and ADELFE 3.0. Section four presents a synthesis. Finally, we end this survey with some conclusions.

II. COMPARISON CRITERIA

In this section, we describe in table 1 eight criteria which are used later to compare between six MAS methodologies. It should be note that we propose these criteria according to our objective (choose the most appropriate methodology to design IES). So, for each criterion, we define a set of values and each value possesses a rate. We use three rates which are:

Rate	Used for
0	Non-interesting values.
50	Interesting value. The MAS methodology which has this value for any criterion can be used provided that improves this criterion. It should be noted that, the design of IES with MAS methodology becomes more complicated if many criteria possesses this value.
100	Very Interesting value. The MAS methodology which has this value for most criteria can be used with success to design IES.

TABLE 1 : DIFFERENT CRITERIA USED TO COMPARE BETWEEN MAS METHODOLOGIES

N°	Criterion name	Criterion description	Possible values	Values rates
1	**Autonomy**	The IES operates without direct human or other intervention. So, we must choose a MAS methodology which allows the development of autonomous agent.	V1= autonomous.	Rate of V1= 100.
			V2 = not autonomous.	Rate of V2= 0
2	**Take into account Non functional Requirements (NFRs)**	There are two types of requirements, functional and non-functional requirements. The second type is an attribute/constraint on a system and in fact receives less attention in the design of many types of systems but in the IES they play a very important role because the design of these systems is a complex job (their resource is limited) which incurs handling a big range of Non-Functional Requirements (NFRs) such as Reliability, determinism, security, timing etc. Therefore, satisfaction of NFRs plays an important role in the correctness of the design of these systems (system design which cannot satisfy its NFRs can mean failure of the end product).	V1 = MAS methodology take into account this criterion.	Rate of V1 = 100.
			V2 = MAS methodology not take into account this criterion.	Rate of V2 = 50
3	**Cover All phases**	We must choose a methodology which covers all phases from the requirements analysis to implementation, because our work is adapted the methodology according to requirements of the	V1 = MAS methodology cover all phases.	Rate of V1 = 100.

ISSN 2356-5608

		IES and not improve the MAS methodology.	V2 = MAS not cover all phases.	ate of V2 = 0.
4	**Availability of tools**	Is the tool exist, free and available or not? Because the tools simplify the development and the availability of the code source we allow integrating new plug-ins if necessary to develop IES.	V1 = tool free and available.	Rate of V1 = 100.
			V2= tool free and not available.	Rate of V2 = 50.
			V3 = tool not free or does not exist a tool.	Rate of V3 = 0.
5	**Used for real time systems**	The most of IES must react to stimuli within the time interval dictated by the environment (hard/soft real time).	V1 = MAS methodology used for real time application.	Rate of V1 = 100.
			V2 = MAS methodology not take into account the real time.	Rate of V2 = 0.
6	**HW/SW Partitioning**	It is a crucial steps in the design of IES because in this step we must deciding which components of the system should be implemented in hardware and which ones in software.	V1 = MAS methodology take into account this criterion.	Rate of V1 = 100.
			V2 = MAS methodology not take into account this criterion.	Rate of V2 = 0.
7	**open system**	Currently, most of the problems are more complex (there is not a known algorithm for solving a given problem). Therefore, the internal organization of the systems is not known a priori which makes these systems more complexes, opens and must evolve in a dynamic environment.	V1 = MAS methodology used for open systems.	Rate of V1 = 100.
			V2 = MAS methodology used for closed systems.	Rate of V2 = 0.
8	**Used for self-adaptive systems**	Currently, Most of the IES are autonomous and in the same time worked in the dynamic environment i.e. impossible to specify all the situations they will face. So, we must choose a MAS methodology which guaranties this criterion.	V1 = MAS methodology used for self adaptive systems.	Rate of V1 = 100.
			V2= MAS methodology used for adaptive systems.	Rate of V2 = 50.
			V3 = MAS methodology used for not adaptive systems.	Rate of V3 = 0

ISSN 2356-5608

III. Overview of Agent Oriented Methodologies

Diversity in MAS attributes like autonomy, social ability, reactivity, real-time capabilities and self adaptation…etc led to the emergence of a large number of MAS methodologies (More than 80 methodologies). Figure 1 presents in a curve the number of Agent Oriented methodologies designed each year, since 1990 till present.

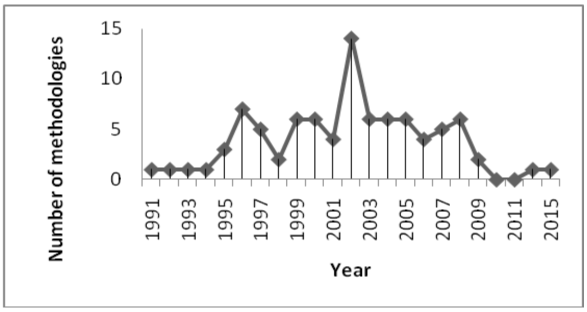

Fig. 1: Number of methodologies designed each year

Some of these methodologies are general in the sense they can be used in any context such as MASE methodology or GAIA methodology, however the rest are domain specific as ELDAMeth [6] which is used for simulation-based prototyping of distributed agent systems. Some methodologies cover all design steps from needs analysis to implementation like Prometheus but others do not cover all the steps like MESSAGE. Indeed, there are three main families of MAS methodologies: The first one derived from object oriented (OO) paradigm like ingenias methodology, The second one based on the idea of artificial intelligence coming from the knowledge engineering (KE) like CoMoMas methodology or DESIRE methodology, The last one is not really a methodology rather than it is a simulation platform like Promotheus. Note that there are other methodologies using a mix of concepts like: Tropos methodology or MAS_KommonKADS methodology. Finally, it is necessary to mention that recently many researchers like Lichen Zhang [7] developed some preexisting methodologies to take into account the real time. In this section, we try to provide some MAS methodologies that are recaptured in figure 2.

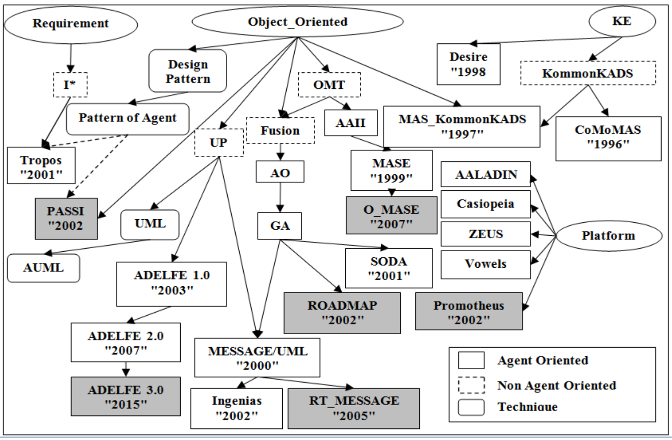

Fig. 2: Genealogy of MAS Methodology

IV. General description of ROADMAP, Passi, Prometheus, RT-Message, O-Mase and ADELFE 3.0

A. *Roadmap methodology (2002) :*

ROADMAP methodology which is proposed by T.Juan el al [8] at the University of Melbourne. It is an extension of the GAIA Methodology for Complex Open Systems. There are mainly two phases in ROADMAP [8]: Specification and analysis phase (ROADMAP extends the analysis phase of the GAIA in the sake of cover also the specification phase) and Design phase.

Until now, ROADMAP is not supported by any tools and not exist a full case study to evaluate this methodology. So, the use of this methodology in a real project is very difficult and creates a big risk.

B. *Passi methodology (2002):*

PASSI [10],[11] or "steps" in Italian is a step-by-step requirement-to-code method for designing and developing multiagent software. It is a generic methodology that can be applied to any field. Nevertheless, PASSI has the particularity to take into account the modeling of mobile agents.

PASSI is an iterative-incremental process for designing MAS starting from functional requirements. It is a complete and detailed process, composed of five models including many phases. The modeling with PASSI is supported by the PTK tool (PASSI Toolkit).

PASSI focuses primarily on the functionality of the system by defining tasks and assigning roles, social skills and adaptation are not the centerpiece of the design process [9]. Also, we cannot use Passi methodology for open systems.

C. *Prometheus methodology (2002):*

The Prometheus methodology is a detailed Agent Oriented Software Engineering methodology, created by Lin Padgham and Michael Winikoff in 2002 [14]. It aims at covering all of the activities required in developing agent systems from specification requirements to detailed design. So, This methodology is composed of three design phases [10]: System specification phase, High-Level (Architectural) design phase and Detailed design phase. Also, it is intended primarily for multi-agent systems using the BDI architecture [13].

Prometheus is supported by two tools: PDT (Prometheus Design Tool) and JDE (JACK Development Environment).

D. *RT-Message (2005):*

The Real Time-MESSAGE methodology [16],[15] is specifically based on the MESSAGE methodology and on the Real Time Unified Modeling Language (RT-UML) [15]. The selection of MESSAGE is based on an analysis of the features of several methodologies. For more detail refer back to [16].

Rt-message covers the analysis, design and implementation of Real Time Multi Agent Systems. It is supported by InSiDE+ tool but this tool is not available.

ISSN 2356-5608

E. O-Mase (2007):

Scott A. DeLoach and Juan C. Garcia-Ojeda proposed a new methodology called Organization-MaSE (O-MaSE) methodology [17],[18]. It is based on the MaSE methodology. The main aim of this methodology is to enable designers to create customized agent-oriented software development processes. In O-MaSE there are three mains phases which are: Requirement Analysis, Design, and Implementation. Note that when using O-MaSE on a real project, designers can define their own set of phases and iterations and to assign Activities and Tasks to those phases and iterations as appropriate.

O-Mase is supported by AgentToolIII which is free and available tool.

F. Adelfe 3.0 (2015):

In 2003, Picard proposed ADELFE 1.0 [9],[20] which is a French acronym that signifies toolkit for designing software with emergent functionalities. ADELFE is expressed using OMG's SPEM. It is used just when the environment is unpredictable or the system is open. ADELFE methodology based on the AMAS technology but it does not assume that the designer is specialized in this domain. Therefore, in this methodology we will talk about WorkDefinitions (ADELFE proposes four workflows which are the Preliminary Requirements, final requirements, analysis and design), Activity and Step.

In 2007, S.Rougemaille [19] propose an extension of the process (ADELFE 2.0) by an implementation phase which is use UML2 for general activities, use AMASML (AMAS Modelling Language) and SpeADL (Species-based Modelling Language) for specific activities.

In 2015, W.Mefteh et al [21] propose a new version of ADELFE (ADELFE 3.0) based on a simulation based design approach in the sake of assist the designer of AMAS and make his task less difficult. ADELFE 3.0 provide 4 tools which are: Opentool, Adequation, ADELFE ToolKit and MAY.

V. SYNTHESIS

According to the following table, we can remark:

- All of methodologies guaranty the first criterion (autonomy) because according to Wooldridge and Jennings [12] among the most principal features of an agent is autonomy. So, most of methodologies guaranties this criterion.
- Between six methodologies, Only ADELFE methodology offers to designer the possibility to specify non functional requirements. Furthermore, only ADELFE can be used for self adaptive systems because it is based on the Adaptive Multi Agent Systems (AMAS).
- Only RT-Message methodology can be used for real time application.
- Any MAS methodology offers to designer the possibility to identify which part of the system is implemented with Hardware part and which part in Software part.
- We can only used Roadmap or Adelfe 3.0 methodology for open systems.

TABLE 2 : SUMMARY OF COMPARISON

Criterion / Methodologies	Autonomy	Take into account Non functional Requirements (NFRs)	Cover All phases	Availability of tools	Used for real time systems	HW/SW Partitioning	open system	Used for self-adaptive systems	Total
Roadmap	100	50	0	0	0	0	100	0	250
Passi	100	50	100	100	0	0	0	0	350
Prometheus	100	50	100	100	0	0	0	0	350
RT_Message	100	50	100	50	100	0	0	0	400
O_Mase	100	50	100	100	0	0	0	0	350
Adelfe 3.0	100	100	100	100	0	0	100	100	600

VI. CONCLUSION

In this paper we tried to compare between six methodologies according to IES specificities related criteria. According to our first comparison and under the absence of some information regarding some MAS methodologies, we can observe that ADELFE 3.0 may be a good candidate for IES since it satisfies many important IES requirements, recall for example: the methodology supports the modeling of both functional and non-functional requirements, it takes into account both open systems and self adaptive systems, The agents in this methodology are cooperative and heterogeneous. Furthermore, ADELFE 3.0 covers all phases from preliminary requirements to implementation and supported by a tools. All these features make from ADELFE 3.0 a good candidate to IES design compared to other methodologies.

ISSN 2356-5608

REFERENCES

[1] W. Elmenreich, “Intelligent Methods for Embedded Systems”, in proc. the First Workshop on Intelligent Solutions in Embedded Systems (WISES), Vienna: Austria, 2003, p.3-11.

[2] C. Rust and B. Kleinjohann, “Modeling Intelligent Embedded Real-Time Systems using High-Level Petri Nets”, in proc. the forum on Design Languages, FDL, 2001.

[3] J-P. Jamont, “DIAMOND: une approche pour la conception de systemes multi-agents embarques”, M. Occello, thesis. Institut National Polytechnique de Grenoble – INPG, French, September 2005.

[4] L. Ostaseviciute and E. Kazanavicius, “Agent-Component Design of Smart Appliances”, Solid State Phenomena, Vol.. 147-149, pp.101-106, 2009;

[5] L. Kizauskiene, E.Kazanavicius and R.Gaidys, “Agent-based methodology for developing mechatronic systems software”, vols.17-5, pp.551-556, 2011.

[6] L.Cernuzzi, A.Molesini and A.Omicini, “The Gaia Methodology Process”, Handbook on Agent-Oriented Design Processes, M.Cossentino, V.Hilaire. A.Molesini and V.Seidita, Springer-Verlag Berlin Heidelberg, 2014; p.141-172.

[7] L.Zhang, “Development Method for Multi-Agent Real-Time Systems”, International Journal of Information Technology, vol.12, pp.19-28, 2006.

[8] T.Juan, A.Pearce and L.Sterling, “ROADMAP: Extending the Gaia Methodology for Complex Open Systems”, in proc. the first international joint conference on Autonomous agents and multiagent systems: part 1, 2002, p.3-10.

[9] R.Sylvain, “Ingénierie des systèmes multi-agents adaptatifs dirigée par les modèles”, F.Migeon and Ch.Maure. thesis, Universite of Paul Sabatier:Toulouse III, 2008.

[10] M. Cossentino and C. Potts, “PASSI : a process for specifying and implementing Multi-Agent Systems Using UML”, 2001.

[11] M. Cossentino, “From requirements to code with the PASSI Methodology”, In Agent-Oriented Methodologies, B.Henderson-Sellers and P. Giorgini (Editors). Idea Group Inc., Hershey, PA, USA, 2005.

[12] M.Wooldridge and N. Jennings, “Intelligent agents: theory and practice”, The Knowledge Engineering Review, vol.10:2, p.115-15, 1995.

[13] D.Santos, M.Ribeiro and R.Bastos, “A comparative study of Multi-Agent systems development methodologies”.

[14] L.Padgham and M.Winikoff, Ed, Agent-Oriented Software Engineering III , ser. Lecture Notes in Computer Science, Springer, 2003, vol.2585.

[15] V. Julián, J. Soler, M.C. Moncho and V. Botti, “Real-Time Multi-Agent System Development and Implementation”, Recent Advances in Artificial Intelligence Research and Development, 2004.

[16] V. Julian and V. Botti, “Developing real-time multi-agent systems”, Integrated Computer-Aided Engineering journal, vol.11. Issue 2, pp.135–149. April 2004.

[17] Scott A. DeLoach and Juan C. Garcia-Ojeda, “The O-MASE Methodology”, Handbook on Agent-Oriented Design Processes, M.Cossentino. V.Hilaire, A.Molesini and V.Seidita. Springer-Verlag Berlin Heidelberg., 2014; p.253-286.

[18] Juan C. Garcia-Ojeda, Scott A. DeLoach, Robby, Walamitien H. Oyenan and Jorge Valenzuela, “O-MaSE: A Customizable Approach to Developing Multiagent Development Processes”, Agent-Oriented Software Engineering VIII: 8th International Workshop, AOSE 2007, Springer-Verlag Berlin Heidelberg, 2008.

[19] N. Bonjean, W. Mefteh, M.P. Gleizes, C. Maurel and F. Migeon, “ADELFE 2.0”, Handbook on Agent-Oriented Design Processes, M. Cossentino, V. Hilaire, A. Molesini, and V. Seidita, (Eds). Springer-Verlag Berlin Heidelberg, 2014; p.19-64.

[20] C.Bernon, M-P. Gleizes, G. Picard and P. Glize, “The ADELFE Methodology For an Intranet System Design”, in proc. Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002), Toronto (Ontario, Canada) at CAiSE'02, 2002.

[21] W. Mefteh, F. Migeon, M-P. Gleizes and F. Gargouri, Ed, Computational Collective Intelligence, ser. Lecture Notes in Computer Science,. 2015, vol.9329.

ISSN 2356-5608