DOI: https://doi.org/10.14201/0AQ03153547

A BRIEF REVIEW ON MULTI-AGENT SYSTEM APPROACHES AND METHODOLOGIES

BREVE REPASO A LOS ENFOQUES Y METODOLOGÍAS DE LOS SISTEMAS MULTIAGENTE

María-Eugenia Pérez-Pons[1], Javier Parra-Dominguez[1], Camilo Anzola-Rojas[2], Ramón J. Durán Barroso[2], Ignacio de Miguel[2], Jonas Queiroz[3], and Paulo Leitao[3]

[1] *BISITE Research Group, University of Salamanca, Edificio Multiusos I+D+i, Calle Espejo, 2, 37007, Salamanca, Spain*

{eugenia.perez, javierparra, corchado}@usal.es

[2] *University of Valladolid, E.T.S.I. Telecommunication, Campus Miguel Delibes, 47011 Valladolid, Spain*

{rduran}@uva.es

[3] *Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolonia, 5300-253 Braganca, Portugal*

{jqueiroz, pleitao}@ipb.pt

Abstract: Agents are understood by definition as entities that interact with their environment and also with other agents. Thus, when two or more agents are able to work together in order to solve a common problem, they form a multi-agent system (MAS). MAS are systems that integrate a set of agents that interact, communicate and coordinate to achieve the established objectives. MAS are designed to meet a set of objectives according to a set of rules and standards. This article briefly summarizes different MAS classifications and methodologies that have been extensively used in the literature.

Keywords: multi-agent systems; MAS methodologies; MAS classification.

Ediciones Universidad de Salamanca / cc by-nc-nd Proceedings of the IV Workshop on Disruptive Information and Communication Technologies for Innovation and Digital Transformation, pp. 35-47

Resumen: Los agentes se entienden por definición como entidades que interactúan con su entorno y también con otros agentes. Así, cuando dos o más agentes son capaces de trabajar juntos para resolver un problema común, forman un sistema multiagente (SMA). Los SMA son sistemas que integran un conjunto de agentes que interactúan, se comunican y se coordinan para alcanzar los objetivos establecidos. Los SMA se diseñan para cumplir una serie de objetivos de acuerdo con un conjunto de reglas y normas. Este artículo resume brevemente las diferentes clasificaciones y metodologías de los SMA que se han utilizado ampliamente en la literatura.

Palabras clave: sistemas multiagente, metodologías SMA, clasificación SMA.

1 Introduction

An agent is a computer system located in some environment that is capable of acting autonomously in an environment to meet its design goals. Autonomy is a difficult concept to define as it is very broad but in this context, it refers to a system's ability to act without the direct intervention of humans (or other agents) and to control its own actions and internal state [20]. Agents can be defined as intelligent entities with social skills (interaction, collaboration, communication, coordination, competence, negotiation, intelligence) that encapsulate a functionality to solve a problem [32, 19].

The agent paradigm emerged to satisfy the deficiencies that classical software engineering had when modelling complex software systems. In this sense, according to [9, 18, 32], the agent paradigm can be applied to three main types of systems:

- **Open systems**: These systems are able to change dynamically, because their components are not known a priori and are highly heterogeneous (different entities, implementations and techniques, etc.). Negotiation and cooperation techniques are necessary to solve this type of systems, the basis of which is part of multi-agent systems [1].
- **Complex systems**: They are related to large and unpredictable systems, whose approach is the use of abstraction and modularity techniques. Both characteristics are an intrinsic part of MAS, since

the notion of an autonomous agent is itself an abstraction of a module that encapsulates procedures and data (objects) that allow to solve a problem autonomously [17].

– **Ubiquitous systems**: This type of systems enhance the use of a computer system by utilizing the computational power of a physical environment, usually distributed, but somehow abstracting its complexity away from the end user. The system must cooperate with the user to achieve the desired objectives.

In sum, agents must be able to interact, negotiate and coordinate to achieve the common goal.

The remainder of this study is structured as follows: Section 2 presents the most widely used MAS classifications found in the literature, Section 3 presents MAS methodologies, Section 4 the communication between agents and finally, the last section presents the conclusions drawn from the review.

2 MAS Classifications

The other major focus of interest has been the classification of agents according to different criteria [26, 23, 4, 11, 30]. Nowadays, in the literature there are different classifications of agents, highlighting mainly the following taxonomies:

– Classification of the type of agent implementation [26]. According to this classification, the following four types of agents are distinguished, which are presented in order of complexity: simple reflex agent, reflex agent with internal state, goal-based agents and utility-based agents.
– Classification according to the type of agent attributes [23]. For instance, [23] makes a complex classification of agents, distinguishing agents according to different characteristics. Thus, according to :
  • Mobility, there are static and mobile agents.
  • According to internal reasoning model, there are deliberative and reactive agents.
  • Attributes such as autonomy, cooperation and learning, there are collaborative, interface, collaborative learning and smart agents.

    - The union of characteristics of these types of agents gives rise to a new type of agent, called a hybrid agent.
- Classification according to the conceptual design. Taking into account the conceptual design of the agents, there are interface, search and information agents.
- Classification according to the architecture [30]. The last classification to be presented in this paper is based on the internal architecture of the agent. Following this criterion, reactive, deliberative and hybrid architectures are distinguished.

Within these classifications of agents, [30] classify according to the architecture of the agent, since it determines what the agent's main components are, and how they interact with each other to achieve the final mission. If an agent is considered to be a complex system, the architecture should describe the internal structure of the agent, explaining how the agent is decomposed into independent modules that interact to achieve the required functionality. In this sense [30] proposes three classic architectures.

- Reactive architectures in which the agent lacks both reasoning and the ability to represent its environment, and its actions are modelled by basic rules [22].
- Deliberative architectures where the agent is able to maintain a symbolic representation of knowledge and plan the set of actions to be taken to achieve its goals.
- Hybrid architectures, which is a model of architecture that combines the characteristics of the two aforementioned architectures.

The most widely studied and extended deliberative architecture is the BDI (Belief, Desire, Intentions) model [13], which has been the most widespread among reasoning models, since it combines a philosophical model associated with human reasoning and a considerable number of implementations [12]. Intelligent agents have different roles, as outlined below [3]:

- **Beliefs**. Related to the set of propositions that the agent accepts as true. That is, the agent's view of the environment and the state of the other agents.

- **Desires or goals or objectives**. Refer to the set of properties that the agent tries to make true.
- **Intentions**. Associated with the set of planned actions that allow it to reach a desired state.

These agents provide solutions in dynamic and uncertain environments. Moreover, they are capable of dealing with real-world problems, even when they have only a partial view of the problem and a limited number of resources.

In addition, most software systems existing today are highly complex (such as Cloud Computing environments), since they are usually concurrent systems, interacting with each other and with other external systems [33].

1.1 Multi-Agent Systems and Optimization

An optimization problem consists in finding the best solution, according to a set of criteria, within a set of possible solutions. Optimization algorithms are general procedures that solve a problem by producing feasible solutions through problem contextualization. Optimization algorithms can be seen from two different viewpoints, they are exact if they find the optimal solution or heuristic if the solution is not necessarily optimal.

The ideal would be to obtain an optimal solution, however, in problems where the human factor exists, it is necessary to apply heuristic methods to adapt them to the needs or characteristics that the user desires. To apply these methods, it is necessary to:

1. Identify the problem.
2. Define and present the problem.
3. Explore viable strategies.
4. Advance the strategies.
5. Achieve solutions and evaluate the effects of the application of these solutions.

3 MAS Methodologies

In recent years, the rise of organizational concepts has undergone a process of rapid development. Thus, a great variety of models and methodologies for their development can be found in the state of the art, some of them based on artificial societies. The development of these methodologies, whose objective is to facilitate agent-based software design tasks, has also led to the evolution of Agent Theory itself [32]. Some of the most common methodologies used for multi-agent systems are described in the Table 1.

Table 1. Multi-agent methodologies

Name	Methodology Description		Authors
GAIA	Provides a set of models that are used in the analysis and design stages of multi-agent system development and evolve throughout that process	Environmental Preliminary role Preliminary interaction Organizational rules	[31, 34]
INGENIAS	Is based on the well-known and established software development process; the unified process. It is based on the definition of a set of meta-models that describe, from various points of view, the elements that make up a multi-agent system	Organization environment task/goals agent interacion	[25]
MASE	Uses a series of graphic models to describe system objectives, behaviors, agent types and agent communication interfaces. It uses most of the Unified Modelling Language (UML) diagrams and makes some improvements to fit the MAS domain	Capturing Goals Applying the use cases Refining roles	[8]
TROPOS	Agent-oriented software development methodology, based on two key features: (1) the notion of agent and the associated mentalist notions (e.g. objectives and tasks) (2) the analysis of the requirements and the specification of the system to be analyzed with respect to its intended environment	Early requirements analysis late requirements analysis Architectural Design Detailed design Implementation	[5, 14]

Name	Methodology Description		Authors
PROMETHEUS	Specifically aimed at the construction of intelligent agents	System Specification Architectural Design Detailed design	[24]
PASSI	Step-by-step methodology for designing and developing multi-agent partnerships. Integrating design models and concepts from software engineering approaches using the UML notation	Domain Description Agent Identification Role Identification Task specification	[6]
DECAF	Flexible multi-agent system, which is a set of software tools for rapid design, development and execution of intelligent agents for complex software systems	Distributed Environment Centered Agent Framework	[15]
RETSINA	The agents of the system must form a community of peers who are committed to peer-to-peer relationships. The structure of the community of actors must emerge from the relationship between the actors rather than be imposed by the infrastructure. It does not employ any centralised control, implements distributed infrastructure services that facilitate the relationships between agents	operating environment communication infrastructure ACL infrastructures MAS management services Performance Services security	[27, 28]

As seen in the Table 1, GAIA [31] is undoubtedly one of the main MAS development methodologies and has been widely used in different studies and contexts. After its emergence, the methodology was revised to include organizational concepts, the notion of environment, and a set of tools and techniques to facilitate its use in open environments; this revision has been called GAIA II or GAIAexOA [34]. In recent years, organizational concepts have undergone a process of rapid development. Thus, a great variety of models and methodologies for their development can be found in the state of the art, some of them based on artificial societies. The development of these methodologies, whose objective is to facilitate agent-based software design tasks, has also led to the evolution of Agent Theory itself [32, 18].

The design process of a networked system or a Cloud Computing (CC) environment, requires the study of the design methodologies of the Virtual Organizations (VO) themselves. This is so because each methodology includes its own tools, techniques and models for modelling, but in most cases these methodologies have numerous common features; different authors have defined them as the meta-model of the multi-agent paradigm [7, 2].

3.1 MAS methodologies analysis

Once the main methodologies for the design and (in some cases) development of MAS have been presented, it can be seen at a glance that all of them have a set of common features; features that have evolved as they have been revised over time. Thus, the classical methodologies have a clear design approach focused on the agent itself, while revisions of these methodologies have evolved to an organizational design model.

In the organizational systems' meta-model, there are two key concepts, the role and the organization (or group), in addition to the agent concept itself. It is common to divide the study of organizational systems into two levels of abstraction [31]; the structural (or macro) that takes into account the dynamic and organizational aspects (roles and groups); and, the concrete (or micro) level that shapes the low-level definition of the agents (tasks, plans, etc.).

4 Communication between agents

Over the years, the different types of communication between agents have been studied. Some of them are defined below:

- Speech act: An agent can act as a speaker (S) who produces an utterance to change the beliefs of the hearer (H) [21].
- Message: Agents use point-to-point or broadcast communication to talk to other agents. In the former, agent A can talk directly to agent B if it knows the agent's address.
- Blackboard: In this communication method, agents can collaboratively share data with each other using a central repository called Blackboard.

Nevertheless, the definition of the concept of organization is much more complex. First, the organization has to describe the objectives for which it has been designed, bearing in mind that these objectives have to be aligned with those of its members (roles or suborganizations) and that to some extent they provide rationality to the system as a whole. This design, although it has different nomenclatures depending on the methodology, it tends to include social, communicative, interactive and normative aspects [9]. Each of these is described below:

1. Social aspects refer to the description of the set of roles, groups (role associations) and the relationship between them. Regarding the existing relationships between roles and groups (recursively), some authors have defined a set of social structures that allow to model the interactions between members. Among the main structures, the following stand out: hierarchies, coalitions, teams, congregations, societies, federations, markets, matrixes and composite organizations. Some studies have simply defined possible relationships between members [9] such as dependency, hierarchy, use, etc.
2. The communication aspects refer to the means or language that makes the exchange of information possible. That is, a knowledge representation language (usually represented by an ontology) and a communication language. The communication sequence between two agents is called illocution [10], communication act [9] or link [16], and can have different purposes depending on the philosophy of the message [9]: representation, prohibition, permissions, declaration, expressive, commitment or directive.
3. Interaction aspects refer to how roles collaborate to achieve common goals. That is, given that there may be objectives that cannot be achieved individually, and that require the combination of several agents for achievement, it is necessary to describe an interaction structure that allows to articulate or regulate the achievement of individual sub-objectives that in turn make the achievement of higher-level objectives possible. It should be emphasized that these interaction sequences evolve as the relationships between roles and groups progress.
4. Normative aspects, it should be noted that this is one of the main pillars of organizational MAS, and there is even a methodology based on this concept HARMONIA [29]. Norms (or institutional patterns

[34] make it possible to establish a relationship of trust between the members of an organization, since they limit the free will of individual agents. Any external agent, who wishes to be part of an organization, must comply with the standards. In short, they formally define the obligations, prohibitions and permissions of the members and of the communications between them.

In addition to the concepts that have just been presented (role, organization, norms and social structures), organizational MAS routinely include another key concept; environment. Agent theory traditionally conceives the agent as an entity that plans its actions based on its perception of the environment. However, the increasing complexity of the environment itself in the context of open systems (dynamic, heterogeneous and unpredictable) can not only make the MAS unpredictable, but also difficult to interact with.

5 Conclusions

Despite their broad applicability, MAS still face a number of challenges, such as coordination between agents, security and tasking. This study provides a comprehensive analysis of all aspects of MAS, from definitions, characteristics, applications, challenges, and communications to evaluation. A classification of MAS applications and challenges is provided along with references for further study. We hope that this paper will serve as an insightful and comprehensive resource on MAS for researchers and practitioners in the field.

6 Acknowledgements

This work has been partially supported by the European Regional Development Fund (ERDF) through the Interreg Spain-Portugal V-A Program (POCTEP) under grant 0677 DISRUPTIVE 2 E (Intensifying the activity of Digital Innovation Hubs within the PocTep region to boost the development of disruptive and last generation ICTs through cross-border cooperation).

References

1. Bond, A.H., Gasser, L.: Readings in distributed artificial intelligence. Morgan Kaufmann (2014)
2. Bonjean, N., Gleizes, M.P., Chella, A., Migeon, F., Cossentino, M., Seidita, V.: Metamodel-based metrics for agent-oriented methodologies. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems Volume 2. pp. 1065–1072. Citeseer (2012)
3. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational intelligence 4(3), 349–355 (1988)
4. Brenner, W., Zarnekow, R., Wittig, H.: Intelligent software agents: foundations and applications. Springer Science & Business Media (2012)
5. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and MultiAgent Systems 8(3), 203–236 (2004)
6. Chella, A., Cossentino, M., Sabatucci, L., Seidita, V.: Agile passi: An agile process for designing agents. International Journal of Computer Systems Science & Engineering 21(2), 133–144 (2006)
7. DeLoach, S.A.: Omacs: A framework for adaptive, complex systems. In: Handbook of research on multi-agent systems: Semantics and dynamics of organizational models, pp. 76–104. IGI Global (2009)
8. DeLoach, S.A., Wood, M.F., Sparkman, C.H.: Multiagent systems engineering. International Journal of Software Engineering and Knowledge Engineering 11(03), 231–258 (2001)
9. Dignum, M.: A model for organizational interaction: based on agents, founded in logic. SIKS (2004)
10. Esteva, M., Rodriguez-Aguilar, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specification of electronic institutions. In: Agent mediated electronic commerce, pp. 126–147. Springer (2001)
11. Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: International workshop on agent theories, architectures, and languages. pp. 21–35. Springer (1996)
12. Georgeff, M.P., Lansky, A.L.: Reactive reasoning and planning. In: AAAI. vol. 87, pp. 677–682 (1987)
13. Georgeff, M.P., Rao, A.: An abstract architecture for rational agents. In: Proc. of the Third International Conference on Principles of Knowledge Representation and Reasoning. pp. 439–449 (1992)

14. Giunchiglia, F., Mylopoulos, J., Perini, A.: The tropos software development methodology: processes, models and diagrams. In: International Workshop on Agent-Oriented Software Engineering. pp. 162–173. Springer (2002)
15. Graham, J.R., Decker, K.S., Mersic, M.: Decaf-a flexible multi agent system architecture. Autonomous Agents and Multi-Agent Systems 7(1-2), 7–27 (2003)
16. Hübner, J.F., Sichman, J.S., Boissier, O.: A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: Brazilian Symposium on Artificial Intelligence. pp. 118–128. Springer (2002)
17. Jennings, N.R.: An agent-based approach for building complex software systems. Communications of the ACM 44(4), 35–41 (2001)
18. Jennings, N.R., Corera, J.M., Laresgoiti, I.: Developing industrial multi-agent systems. In: ICMAS. vol. 95, pp. 423–430 (1995)
19. Jennings, N.R., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Autonomous agents and multi-agent systems 1(1), 7–38 (1998)
20. Jennings, N.R., Wooldridge, M.: Applications of intelligent agents. In: Agent technology, pp. 3–28. Springer (1998)
21. Kibble, R.: Speech acts, commitment and multi-agent communication. Computational & mathematical organization theory 12(2), 127–145 (2006)
22. Maes, P.: Designing autonomous agents: Theory and practice from biology to engineering and back. MIT press (1990)
23. Nwana, H.S., et al.: Software agents: An overview. Knowledge engineering review 11(3), 205–244 (1996)
24. Padgham, L., Winikoff, M.: Prometheus: A methodology for developing intelligent agents. In: International Workshop on Agent-Oriented Software Engineering. pp. 174–185. Springer (2002)
25. Pavón, J., Gómez-Sanz, J.J., Fuentes, R.: The ingenias methodology and tools. In: Agent-oriented methodologies, pp. 236–276. IGI Global (2005)
26. Russell, S., Norvig, P.: Artificial intelligence: a modern approach (2002)
27. Sycara, K., Pannu, A., Willamson, M., Zeng, D., Decker, K.: Distributed intelligent agents. IEEE expert 11(6), 36–46 (1996)
28. Sycara, K., Paolucci, M., Van Velsen, M., Giampapa, J.: The retsina mas infrastructure. Autonomous agents and multi-agent systems 7(1-2), 29–48 (2003)
29. Vázquez-Salceda, J.: The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework. Ai Communications 16(3), 209–212 (2003)
30. Wooldridge, M.: An introduction to multiagent systems. John wiley & sons (2009)

31. Wooldridge, M., Jennings, N.R., Kinny, D.: The gaia methodology for agento-riented analysis and design. Autonomous Agents and multi-agent systems **3**(3), 285–312 (2000)
32. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: Theory and practice. The knowledge engineering review **10**(2), 115–152 (1995)
33. Zambonelli, F., Gleizes, M.P., Mamei, M., Tolksdorf, R.: Spray computers: Frontiers of self-organization for pervasive computing. In: 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. pp. 403–408. IEEE (2004)
34. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The gaia methodology. ACM Transactions on Software Engineering and Methodology (TOSEM) **12**(3), 317–370 (2003)