MAS Ontology: Ontology for Multiagent Systems

Felipe Cordeiro[1], Vera Maria B. Werneck[1], Neide dos Santos[1] and Luiz Marcio Cysneiros[2]
[1]*Universidade do Estado do Rio de Janeiro, Master Program in Computational Science, Rio de Janeiro, Brazil*
[2]*York University, School of Information Technology, Toronto, Canada*

Keywords: Agent-based Software Engineering, Domain Ontology, Comparison of Agent-oriented Methodologies.

Abstract: This work describes the Multiagent Systems (MAS) Ontology to assist in the development of multi-agent system using different methodologies. The MAS Ontology consists of fragmenting agent-oriented methodologies following an ontology approach based on the best aspects of four prominent AOSE methodologies and Guardian Angel exemplar that identify the strengths, weaknesses, commonalities and differences. In this paper, we present a brief explanation of Multiagent methodologies and the step-by-step process to describe the agent-based systems domain and how it can be represented. Given the numerous works in the literature about MAS methodologies, our aim is to help select the best and more appropriate properties to be used in Multiagent Systems development.

1 INTRODUCTION

In the past two decades, the agent technology approach has been considered as a new paradigm for developing complex systems. This approach has attracted an increasing amount of interest from the research community and has demonstrated its potential in many fields, such as: (i) working with different types of distributed devices (e.g., sensor networks, mobile phones, and personal computers), (ii) enabling various types of communication and data exchange (e.g., audio and video), and (iii) ability to dynamically adapt to the ever changing requirements and dynamic operating environment (Munroe et al., 2006), (Pěchouček and Mařík, 2008), (Dam and Winikoff, 2013).

Agent-oriented systems must be built in terms of autonomous task-oriented entities. They need to be organized to interact (cooperate, coordinate and negotiate) with one another. To adopt the agents' perspective requires a new set of tools to support software development (Cernuzzi, Cossentino and Zambonelli, 2005).

Currently, we are faced with a multitude of different frameworks, some of them even supported by tools. However, very few methodologies are broad enough to support the whole software development life cycle or to support the complexity of developing such systems. Years ago, Luck, Mcburney and Preist (2003) stated: "One of the most fundamental obstacles to large-scale take-up of agent technology is the lack of mature software development methodologies for agent-based system".

In this work, we assume "methodology" as a set of phases that a practitioner must go through to design an agent-based system. We see a methodology as being composed of general concepts (deals with the question of whether a methodology adheres to the basic notions of agents and multiagent systems), specific concepts (underlying one particular capability or a characteristic), notation (symbols used to represent elements), modeling techniques (set of models that depict a system at different levels of abstraction and different aspects of the system), process (development aspect) and pragmatics (practical implementation aspects) (Sturm and Shehory, 2004).

The main goal of this paper is to provide an ontology structure for selecting the best and more appropriate artefacts to be used to develop one particular Multiagent Systems, the MAS Ontology. It is motivated by a large number of existing approaches and supported by our experience in using some of them. It is important to notice that this work does not claim nor intends to be complete. It is expected to be a first approach that will be perfected overtime but that will yet be of importance to help developers to use the best each of the current four (Gaia, MaSE, Prometheus and Tropos) methodologies covered in this work has to offer.

The main goal of the ontology proposed in this work is to capture and facilitate the reuse of

Cordeiro, F., Werneck, V., Santos, N. and Cysneiros, L.
MAS Ontology: Ontology for Multiagent Systems.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 536-543
ISBN: 978-989-758-187-8

knowledge gained through the evaluation of several MAS methodologies based on more than 20 projects developed with different methodologies using Guardian Angel (GA) Exemplar proposed by Yu and Cysneiros. However, to complement the ontology we added experiences extracted from other well know evaluation studies from the literature. We populated the ontology with the four methodologies because they were evaluated by GA, Sturm and Shehory (2004, 2014) and Dam and Winikoff (2004, 2013).

2 AGENT-ORIENTED METHODOLOGIES

Cernuzzi et.al. (2005) suggests a clean and disciplined approach to analyzing, designing and developing multiagent systems, using specific methodologies and techniques by means of notations, diagrams and tools to support the development.

We assume that each method has strengths and weakness, and these characteristics may influence the use of one methodology over another for one specific project. To validate the MAS Ontology we used four methodologies, namely, Gaia (Zambonelli, Jennings and Wooldridge, 2003), (Wooldridge, Jennings and Kinny, 2000), MaSE (Deloach, 2001), (Deloach, 2004), Prometheus (Padgham and Winikoff, 2002), (Padgham and Winikoff, 2003) and Tropos (Bresciani et al, 2004).

Jennings and Wooldridge proposed Gaia in 1999. It was extended and modified by Zambonelli in 2000 (Wooldridge, Jennings and Kinny, 2000), finally Zambonelli, Jennings and Wooldridge presented a stable version in 2003 (Zambonelli, Jennings and Wooldridge, 2003). Unlike many other methodologies, Gaia starts from modelling requirements. Later it guides developers to a well-defined design for the multiagent system, that way programmers can easily model and implement it, while dealing with the characteristics of complex and open multiagent systems.

MaSE methodology is heavily based on UML and RUP. It is divided into seven phases: capturing goals, applying use cases, refining roles, creating agent classes, constructing conversations, assembling agent classes and system design (Deloach, 2001), (Deloach, 2004).

Prometheus is an iterative methodology covering the complete software engineering process while aiming at the development of intelligent agents (in particular BDI agents). The concepts applied are goals, beliefs, plans, and events, resulting in a specification that can be implemented with JACK (Coburn, 2000). Prometheus covers three phases: the system specification, architectural design phase, detailed design phase (Padgham and Winikoff, 2002), (Padgham and Winikoff, 2003).

Tropos relies on the notion that an agent is based on goals and tasks adopted by the i* framework (Yu, 2009) and offers supports to applications, particularly for the development of BDI agents and the agent platform JACK. (Coburn, 2000). Tropos consists of four phases: early requirements, late requirements, architecture design, detailed design and implementation (Bresciani et al, 2004), (Tropos, 2014), (Coburn, 2000).

3 EVALUATION OF AGENT ORIENTED METHODOLOGIES

Several evaluations of agent orientated methodologies have been published (Dam and Winikoff, 2014), (Sturm and Shehory, 2014), (Dam, 2003), (Dam and Winikoff, 2004), (Tran and Low, 2005), (Elamy and Far, 2008), (Iglesias, Garijo and González, 1999), (Cernuzzi, Rossi and Plata, 2002), (Sure, Staab and Studer, 2002). Sturm and Shehory (2004, 2014), and Dam and Winikoff (2004, 2013), (Dam, 2003) were the most cited works in the MAS area.

Sturm and Shehory (2004), proposed a framework for quantitative and qualitative evaluation of MAS methodologies (Gaia, MaSE and Tropos). It explores the following aspects: concepts, properties, notations and modeling techniques, process and pragmatics. Dam and Winikoff (2004, 2013), (Dam, 2003) illustrate the strengths and weaknesses of MaSE, Prometheus and Tropos methodologies through an attribute-based evaluation process.

The Guardian Angel (GA) Exemplar proposed by Yu and Cysneiros (Yu and Cysneiros, 2002) defines a set of questions to evaluate the behaviour of MAS methodologies and is expressed in terms of a set of numbered scenarios. The GA is an easily comprehended open system that provides automated support to assess patients with chronic diseases through a set of "guardian angel" software agents.

The GA exemplar is a complete solution, with a practical, real and significant enough example, to test and verify how the methodology behaves in close-to-real situations. The primary concern of the exemplar is to highlight the strengths, weaknesses and potentials of each methodology justified by the artefacts (work products) that can be used to answer the methodology questions.

We chose to use the GA exemplar as it was the only one we found in the literature that proposes complex situations that can be used empirically to evaluate different methodologies that go beyond toy problems.

4 DOMAIN THEORY: AGENT-ORIENTED METHODOLOGIES

In order to define a Domain Theory for Agent-Oriented Methodologies, we have compiled the knowledge gathered from papers on AOSE methodologies listed in section 2 (Dam and Winikoff, 2013), (Luck, Mcburney and Preist, 2003), (Sturm and Shehory, 2014), (Tran and Low, 2005), (Elamy and Far, 2008) together with the results from our experience using the Guardian Angel exemplar over the past 10 years.

While building the MAS Ontology, we tried to answer the following research questions: (i) in what situations is a methodology or method fragment best applied?; (ii) which instruments are used to define the methodological questions from GA and from the works of (Dam and Winikoff, 2013), (Luck, McBurney and Preist, 2003), (Sturm and Shehory, 2014), (Tran and Low, 2005), (Elamy and Far, 2008) (iii) what are the general concepts of agents that a MAS methodology should support?; (iv) what are the specific concepts of agents that a MAS methodology can support?; (v) what are the notations and modeling techniques found in the methodology?; (vi) what are the support resources offered by the methodology?

4.1 Approach

In the ontology, we assembled the knowledge generated by using the GA exemplar pertinent to four different methodologies (Gaia, MaSE, Prometheus and Tropos). We organized the knowledge and experiences gained by signaling which work product is responsible for a certain task when answering the questions listed above while applying the exemplar to each of the aforementioned methodologies.

4.2 MethodBase GA: Experience Modeling with GA

The MethodBase GA is the knowledge base that compiles the work done over many years by MSc and last year undergrad Computer Science students. During this time, these students modeled multiagent systems using methodologies such as Gaia, MaSE, Prometheus and Tropos and relying on scenarios proposed in Guardian Angel exemplar. After modeling the solutions, the students answered the methodological issues in accordance with strengths, neutral or weakness, as seen in figure 1.

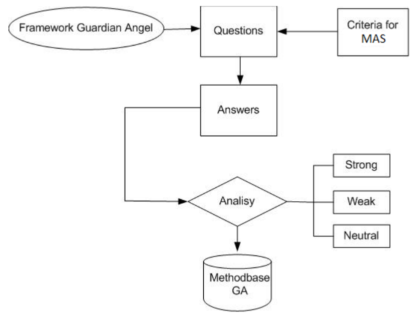

Figure 1: Methodbase GA.

4.3 Evaluated MethodBase: Evaluating Methodologies

The Evaluated MethodBase is the knowledge base built based on the work of Sturm and Dam (Sturm and Shehory, 2014), (Dam and Winikoff, 2013), as illustrated in figure 2. The proposed Evaluated MethodBase includes the following concepts: General Concepts of MAS, Specific Concepts of MAS, Notations, Modeling Techniques, Process and Pragmatics (practical aspects).

The ranking of values ranges from 0 to 6, where 0 represents cases where a certain characteristic is not applicable, 1 Refers to but not detailed, 2 Limited, 3 Neutral, 4 Small issues, 5 Minor deficiencies and 6 is the ideal efficiency. This was an adaptation of (Sturm and Shehory, 2004), (Dam and Winikoff, 2002) using the databases Methodbased GA and Evaluated Methodbase.

5 ONTOLOGY LEARNING FROM EXPERIENCE

Many definitions of ontology can be found in the literature. However, Sure (Sure, Staab and Studer, 2002) provides a simple and comprehensive definition: "An ontology is a formal and explicit specification of a shared conceptualization". In this definition "formal" means readable by computers; "explicit specification" refers to concepts, properties, relations, functions, constraints, axioms, explicitly defined; "shared" means consensual knowledge, and "conceptualization" refers to an abstract model of some phenomenon in the world real. The ontology built in this work was based on a middle-out strategy (Uschold and King, 1995), in which concepts were generalized and specialized.

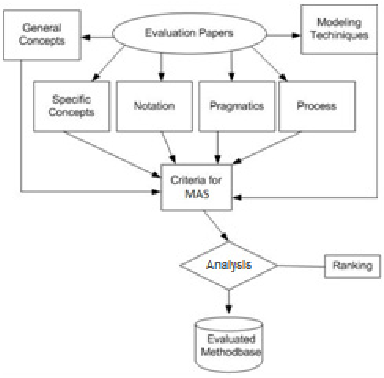

Figure 2: Evaluated Methodbase.

Building on identify concepts (terms) that can provide short assertive sentences, we developed an ontology based on the knowledge acquired from both databases Methodbased GA and Evaluated Methodbase.

The ontology development is defined through seven stages: *Ontology Specification, Knowledge Acquisition, Conceptualization, Formalization, Integration, Implementation* and *Evaluation*.

The *Ontology Specification* is used to prepare a document using natural language, containing information such as the primary ontology goal and its other purposes.

Knowledge Acquisition focuses on possible sources of knowledge. In this survey the GA experiences were used in order to manage the data collected, analyzed and categorized according to their degree of strength, weakness or neutrality.

Conceptualization focuses on structuring the domain knowledge into a conceptual model and was based on the acquired vocabulary in the previous phases, in order to describe the problems and their possible solutions

In the *Formalization*, the concepts are now formally written through OWL. The Protégé tool version 4.3 (Protege, 2000) was used, and the first preliminary version of the ontology was generated. At this stage, a taxonomy that shows the processes of a multiagent system is available.

The *Integration* stage obtains the representative experimental ontology from the Guardian Angel exemplar and is re-evaluated to better address the domain of multiagent systems.

At this stage, other studies on the comparison of methodologies are integrated. (Sturm and Shehory, 2004).

The *Implementation* used the Pellet, a Protégé plugin to automatically check the ontology consistency and also takes into account the experience of validating the data, as well as establishing the comparable relationship between the Methodbase GA and Evaluated Methodbase values, classes and attributes. Each phase of the ontology is related to models, tables or charts, which serve to guide the building process of the MAS Ontology, here defined as products, as seen in figure 3.

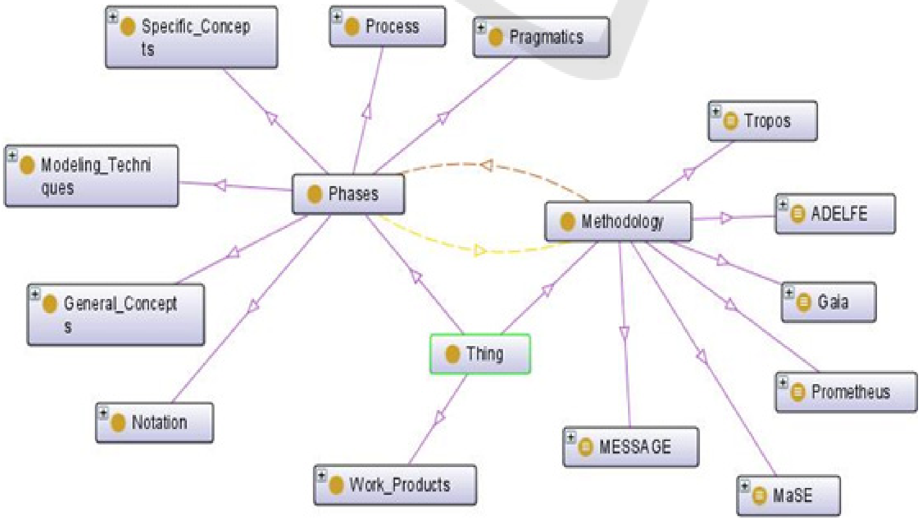

Figure 3: MAS Ontology.

6 MAS ONTOLOGY AND RESULTS

The MAS ontology has three main classes: Methodology, Phases and Work_Products.

The class Methodology focuses on the methodologies that are the study objects (Tropos, ADELFE, Gaia, Prometheus, MaSE, and MESSAGE).

The Phases class combines the characteristics essential to multiagent systems (General Concepts, Modeling Techniques, Notation, Pragmatics, Process and Specific Concepts). Each class has a set of attributes associated with it. (e.g. General Concepts attributes such as: Autonomy, Reactiveness, Sociality, Proactiveness, Reasoning, Mobility).

The class Work_Products lists the necessary artifacts to build a multiagent system.

Figure 3 shows a simplified MAS ontology. The Phase class is associated with the Methodology class. In this relationship, subclasses of Phase are related to subclasses of Methodology. The class Work_Products is also listed to illustrate the artefacts in Figure 4. It is important to determine which attributes from the Phases class might be associated with corresponding work products. For example, in Figure 5 the subclass Tropos_Products has two phases: Tropos_Analysis and Tropos_Design. Each subclass has its own subclasses. Tropos analysis is composed of Actors Diagram and Reasoning Diagram. Tropos Design consist of Extended Actors Diagram, Table of Actors and Capabilities, Table of Agents, Agents Interaction Diagram, Tasks or Plans Diagram and Capabilities Diagram.

6.1 Schematic Model

In order to facilitate understanding the domain of multiagent systems by ontological representation, a Schematic model of MAS Ontology (Figure 6) was developed to illustrate the relationship between classes, attributes and expected values.

In Figure 6 the schematic forms are described as follows:

- Ellipse Form - Classes or Subclasses
- Rectangle Form - Attributes
- Dotted Ellipse - Value types for attributes
- Dotted Rectangle Form - Class properties

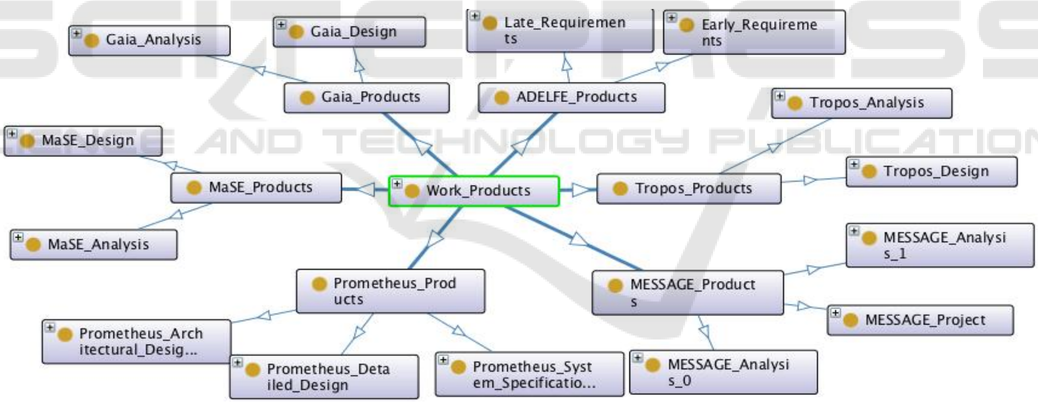

Figure 4: Work Products Detailed.

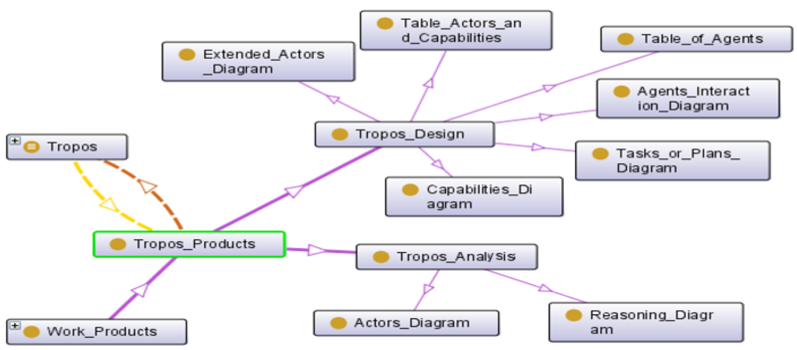

Figure 5: Tropos Work Products.

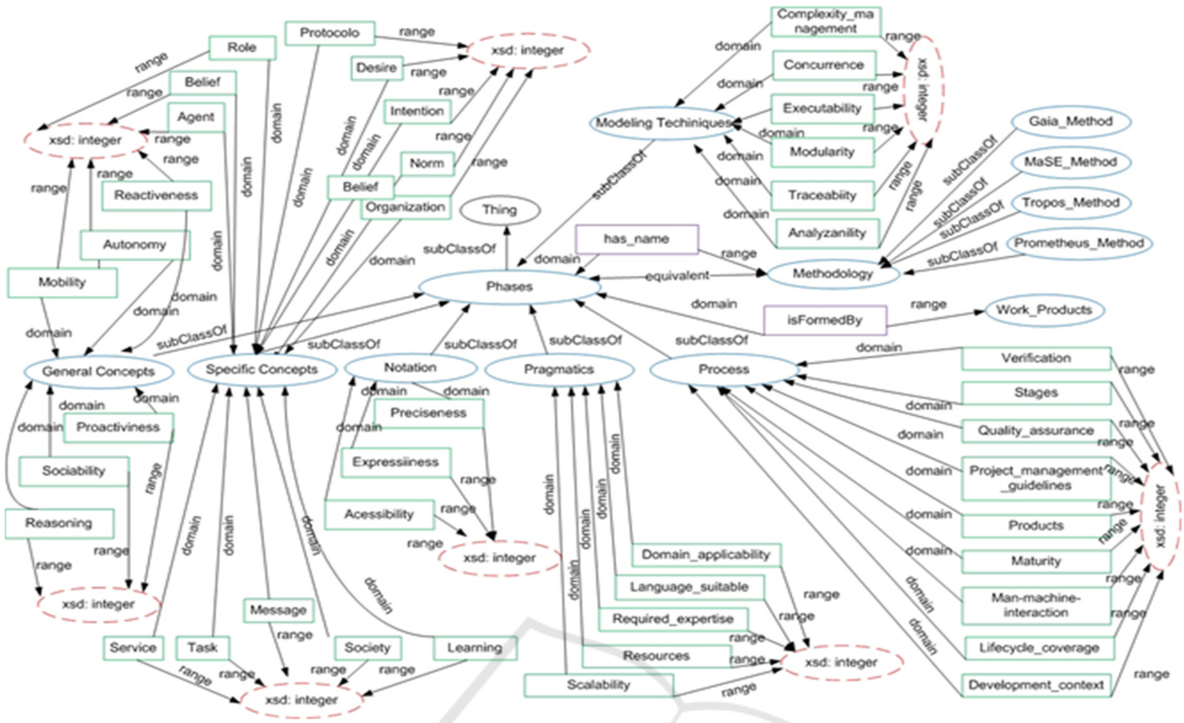

Figure 6: Schematic model of MAS Ontology.

6.2 Examples

For MAS Ontology population the individuals were separated into two groups: (i) representing specific values of GA attributes (Yu and Cysneiros, 2002) and (ii) representing the set of attributes that make up a methodology in the evaluation comparison papers (Sturm and Shehory, 2004, 2014), (Dam and Winikoff, 2002, 2013). Thereby the query may return a particular specific situation or a methodology.

Figure 7a represents an unsuccessful search carried out in plugin DL Query (Protege, 2000), where the attribute Mobility (The quality or state of being mobile) was defined as value 4, and no individual was found. Figure 8b represents a successful search done in plugin DL Query, where the attribute Mobility was defined as value 3

In this scenario, three methodologies were found (figure 7b). Figure 8 shows the associated work products (e.g. Mobility Tropos and Mobility Prometheus) obtained from a refined search for Tropos and Prometheus.

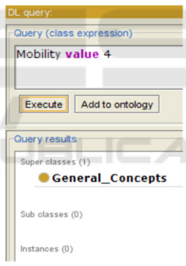

Figure 7.a: Unsuccessful Search.

Figure 7.b: Successful Search.

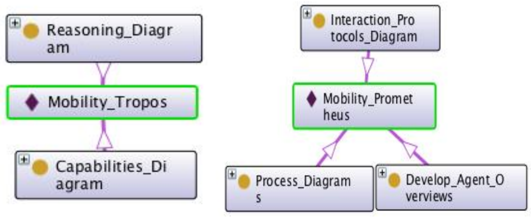

Figure 8: Associated Work Products.

7 CORRELATED WORKS

Several works addressing the evaluation of agent orientation methodologies have been published (Dam and Winikoff, 2002), (Sturm and Shehory, 2004), (Dam, 2003), (Dam and Winikoff, 2004), (Tran and Low, 2005), (Elamy and Far, 2008), (Iglesias, Garijo and González, 1999), (Cernuzzi, Rossi and Plata, 2002), (Sure, Staab and Studer, 2002), (Casare et al, 2014). They consist of quantitative and qualitative evaluation framework based on checklists of certain properties, qualities, attributes or characteristics of the methodology and some simple problems.

Tran and Low (2005, 2005a) compared ten methodologies (Gaia, Tropos MAS CommonKADS, Prometheus, Passi, ADELFE, MaSE, RAP, MESSAGE, Ingenius). They used a criteria checklist that was developed to assess the resources of the chosen methodologies, covering the process, techniques and model stages.

Cernuzzi and Zambonelli (2011) used the multivalued statistical method for quantitative evaluation of profiles. The goal was to present the potential profile analysis in the comparison process for the evaluation of methodologies, searching for similar evaluations to confirm the results.

Our study differs from similar works by proposing the use of a knowledge base where the knowledge is expressed and organized as an ontology. The ontology can guide the developer to select fragments of methodologies that best fit the multiagent system under development. It allows for queries to be made that can help developers to customize their development process. It helps them to search for where the methodologies best fit their needs considering the specific project at hand.

On another level, it also helps researchers further developing these methodologies to easily compare where their approaches fall behind when compared to other existing methodologies and therefore, where they should invest more effort to develop furthere their methodologies.

8 CONCLUSIONS

As a result of the fast dissemination of MAS methodologies, deciding what methodology to use in a project is a complex task. Many frameworks and toolkits are provided, but they do not always offer support to assist developers in choosing the best or most appropriate methodology to handle the project at hand. This paper proposes an ontology-based support to help developers faced with the need to use agent-oriented properties to develop software. The ontology was created based on the experience gathered by applying the Guardian Angel exemplar in four agent-oriented software engineering methodologies, as well as adding the knowledge obtained from the results from Sturn and Dam (2004) and Dam (2003). The knowledge base provided in this ontology can assist developers to use these methodologies and also to choose better the adequate artifacts for a particular domain.

The MAS Ontology approach focuses on being a facilitator for developing a MAS process, as it concentrates on relationships between the principles of software engineering evaluation and experience. Furthermore, it can be extended to suit the particularities of other AOSE methodologies and other studies based on statistics, as in (Iglesias, Garijo and González, 1999).

Future works will address a systematic validation of the Ontology using case studies where different groups of randomly selected students will be asked to develop solutions to a specific problem. Some students will use the ontology, and another set of students will have to develop the solution using pre-determined methodology. Final results will then be compared.

REFERENCES

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), 203-236.

Casare, S. J., Brandão, A. A., Guessoum, Z., & Sichman, J. S. (2014). Method Framework: a situational approach for organization-centered. *MAS. Autonomous agents and multi*-agent systems, 28(3), 430-473.

Cernuzzi, L., & Zambonelli, F. (2011). Improving comparative analysis for the evaluation of AOSE methodologies. International *Journal of Agent-Oriented Software Engineering*, 4(4), 331-352.

Cernuzzi, L., Cossentino, M., & Zambonelli, F. (2005). Process models for agent-based development.

Engineering Applications of Artificial Intelligence, 18(2), 205-222.

Cernuzzi, L., Rossi, G., & Plata, L. (2002, November). On the evaluation of agent oriented modeling methods. In *Proceedings of Agent Oriented Methodology Workshop*, Seattle, (Vol. 29).

Coburn, M. (2000). *Jack intelligent agents: User guide version 2.0*. AOS Pty Ltd..

Cossentino, M., Fortino, G., Garro, A., Mascillaro, S., & Russo, W. (2008). PASSIM: a simulation-based process for the development of multi-agent systems. *International Journal of Agent-Oriented Software Engineering*, 2(2), 132-170.

Cossentino, M., Gaglio, S., Galland, S., Gaud, N., Hilaire, V., Koukam, A., & Seidita, V. (2009). A MAS metamodel-driven approach to process fragments selection. In *Agent-Oriented Software Engineering IX* (pp. 86-100). Springer Berlin Heidelberg.

Dam, K. H. (2003). *Evaluating and comparing agent-oriented software engineering methodologies* (Doctoral dissertation, School of Computer Science and Information Technology, RMIT University, Australia).

Dam H. K., Winikoff M., 2013. Towards a next-generation AOSE methodology. *Science of Computer Programming*, v. 78, n. 6, p. 684-694.

Dam, K. H., & Winikoff, M. (2004, January). Comparing agent-oriented methodologies. In *Agent-Oriented Information Systems* (pp. 78-93). Springer Berlin Heidelberg.

Deloach, S. (2004). The MaSE methodology. *Methodologies and software engineering for agent systems*, 107-125..

DeLoach, S. A. (2001). *Analysis and Design using MaSE and agentTool*. Air force inst of tech wright-patterson afb oh school of engineering and management.

Elamy, A. H. H., & Far, B. (2008). On the evaluation of agent-oriented software engineering methodologies: a statistical approach. In *Agent-Oriented Information Systems IV* (pp. 105-122). Springer Berlin Heidelberg.

Iglesias, C. A., Garijo, M., & González, J. C. (1999). A survey of agent-oriented methodologies. In *Intelligent Agents V: Agents Theories, Architectures, and Languages* (pp. 317-330). Springer Berlin Heidelberg.

Luck, M., McBurney, P., & Preist, C. (2003). *Agent technology: enabling next generation computing (a roadmap for agent based computing)*. AgentLink/University of Southampton.

Munroe S. et al., 2006. Crossing the agent technology chasm: Lessons, experiences and challenges in commercial applications of agents. *The Knowledge Engineering Review*, v. 21, n. 04, p. 345-392.

OMG Group. (2008). *Software & Systems Process Engineering Meta-Model Specification*, at http://www.omg.org/spec/SPEM/2.0/

Padgham, L., & Winikoff, M. (2002, November). Prometheus: A pragmatic methodology for engineering intelligent agents. In *Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies* (pp. 97-108).

Padgham, L., & Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents. In *Agent-oriented software engineering III* (pp. 174-185). Springer Berlin Heidelberg.

Pěchouček M., Mařík V., 2008. Industrial deployment of multiagent technologies: review and selected case studies. *Autonomous Agents and Multiagent Systems*, v. 17, n. 3, p. 397-431.

PROTEGE (2000). The Protege Project. at http://protege.stanford.edu.

Sturm, A., & Shehory, O. (2004, January). A framework for evaluating agent-oriented methodologies. In *Agent-Oriented Information Systems* (pp. 94-109). Springer Berlin Heidelberg.

Sturm, A., & Shehory, O. (2014, January). The landscape of agent-oriented methodologies. In *Agent-Oriented Software Engineering* (pp. 137-154). Springer Berlin Heidelberg.

Sure, Y., Staab, S., & Studer, R. (2002). Methodology for development and employment of ontology based knowledge management applications. *ACM SIGMOD Record*, 31(4), 18-23.

Tran, Q. N. N., & Low, G. C. (2005). Comparison of ten agent-oriented methodologies. *Agent-oriented methodologies*, 341-367.

Tran, Q. N. N., Low, G., & Williams, M. A. (2005). A preliminary comparative feature analysis of multi-agent systems development methodologies. In *Agent-Oriented Information Systems* II (pp. 157-168). Springer Berlin Heidelberg.

Uschold, M., & King, M. (1995). *Towards a methodology for building ontologies* (pp. 15-30). Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh.

Wooldridge, M., Jennings, N. R., & Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and multi-agent systems*, 3(3), 285-312.

Yu, Eric (2009). Social Modeling and i*. In *Conceptual Modeling: Foundations and Applications* (pp. 99-121). Springer Berlin Heidelberg.

Yu, E., & Cysneiros, L. M. (2002, May). *Agent-oriented methodologies-towards a challenge exemplar*. In Proc of the 4 Intl. Bi-Conference Workshop on AOIS, Toronto (Vol. 151).

Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2003). Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology* (TOSEM), 12(3), 317-370.