A Framework for the Evaluation of Agent-Oriented Methodologies

T. Abdelaziz¹, M. Elammari², R. Unland³
¹,³ICB, University of Duisburg-Essen, Essen, Germany
²IT Faculty, University of Garyounis, Benghazi, Libya
¹tawfig@informatik.uni-essen.de, ²elammari@garyounis.edu, ³unlandr@informatik.uni-essen.de

Abstract

A number of methodologies are being suggested for multi-agent systems. Yet, their application is still very limited because they still lack several aspects for the development of agent systems and require improvement. In this context, a comparative framework for the evaluation of those methodologies is needed in order to show their advantages and disadvantages. This framework is an important factor for their improvement and development. This paper presents a framework for comparison and evaluation of agent methodologies. Several known methodologies are evaluated using this framework. The framework aims at recognizing the shortcomings of each of the methodologies and should help those pursuing development and improvement of agent oriented methodologies.

1. Introduction

The most important aspect of the agent oriented technology is its ability to deal with complexity and emergent behaviour of distributed software systems. To construct such complex systems, we need a suitable methodology as a solid foundation to develop the system from the requirement to the implementation stage. Recently, several methodologies have been proposed.

In recent years, many agent-oriented methodologies have been proposed such: Gaia [14], MaSE [10], MESSAGE [2], Tropos [1], HLIM [5], Prometheus [8], AUML [7] and etc. They suffer from number of deficiencies. None of the existing agent-oriented methodologies have been accepted as a standard, an none of them is used and exploited in a wide manner. Till now there is no agent oriented standards have been established, and research that examines and compares properties of these methodologies has suggested that none of them is completely suitable for industrial development of multi-agent systems (MAS). Even though existing methodologies are based on strong agent-oriented basis, they need to support essential software engineering issues such as accessibility and expressiveness which has an adverse effect on industry acceptability and adoption of agent technology. Furthermore, up to this moment there is no well established development process to construct agent-oriented applications. Therefore, the consequences expected by the agents' paradigm cannot be fully achieved yet. Moreover, comparing and selecting agent-oriented methodologies is difficult, since they usually address different properties of software agents and methodological aspects. Therefore, a comparison framework for the evaluation of those methodologies is needed in order to show their advantages and disadvantages.

The motivation behind this framework is that existing methodologies fail to represent industrial development of MAS. Evaluating methodologies' strengths and weaknesses plays an important role in improving them and in developing new generation of methodologies.

This paper presents a Multi-agent System Analysis and Design Framework (MASADF) for comparison and evaluation of agent methodologies. Several well-known methodologies are evaluated using this framework. The framework aims at recognizing the strengths and weaknesses of methodologies and should help methodology developers. The paper is organized as follows: section 2 describes an evaluation framework called Multi-agent System Analysis and Design Framework MASADF. In section 3, three agent-oriented methodologies are evaluated and compared according to the proposed MASADF framework. In section 4, a conclusion is presented. Finally, section 6 states the future work and points out urgent and still open issues in agent-oriented software engineering.
2. MASADF Framework

Our study of the different agent methodologies indicated that the process of evaluation of the different methodologies and making comparisons between them is not an easy task as stated by Juan and et-al [6], Cernuzzi et-al [3], Odell and et-al [7], Sabas and et-al [9] and Silva and et-al [12]. It requires a large amount of important details so the evaluation and comparison can be meaningful, correct, and accurate. An evaluation framework was created and it is called the Multi-agent System Analysis and Design Framework (MASADF). The framework consists of several criteria which were deduced and inspired after the study of some important references such as Shehory and Sturm [11], Dam and Winikoff [4] and Sudeikat and et-al [13]. The study looked at common features among the different methodologies in building agents and how the agent behavior is captured. The MASADF criteria consist of a number of important factors upon which the analysis and design of agents systems depend on. The factors are: concepts, simplicity of visualizing the system, the models, agent attributes, ability to represent agent interactions, agent behavior representation, and software development life-cycle point of view.

2.1 Concepts

This factor is concerned with determining whether the evaluated methodology adheres to agent and multi-agent system concepts. The agent and multi-agent concepts cover the operational and structural concepts. Operational concepts are defined as the concepts that affect the behaviour of the agent and multi-agent system. Structural concepts are defined as the concepts that describe the building blocks of agent and multi-agent system. Agent-oriented concepts are very important for agent-oriented methodologies in general and for agent-oriented modeling languages in particular. These concepts include the definition of agents, their characteristics such as autonomy, reactivity, pro-activity, sociality, adaptability, mental notions (such as beliefs, desires and intention), the relationship and communication between agents, and other concepts such as agent roles and capabilities, as well as events.

2.2 Simplicity of Visualizing the System

This factor describes the ease, the simplicity of understanding, and means of visualizing the system to be created in its entirety. What techniques are used to understand and describe the system behavior? The different techniques and tools that are used in understanding and describing the entire system are defined in a methodical way and compared.

2.3 The Models

One of the most basic factors which affect the evaluation of a methodology is the group of models which it consists of and depends upon. It also includes the level of understanding of those models and the ease with which they can be described, imagined and visualized for a proposed system. Also the level of complexity will have a big effect on its acceptance and success in research and application. Is it simple or complex to understand? Also, is the number of models which make up the methodology so large as to complicate the connections between them?

This factor is also concerned with the level of derivation of models from others. In other words, is there a clear relationship which connects/relates the different models? Or does it lack such relations. Is it possible to extract some of the models from others with relative ease? Or do gaps and splits exist between those models to the extent that it becomes difficult to connect and follow up? This factor is also concerned with knowing how much of the models cover the development phases making up the system.

2.4 Agent Attributes

This factor is concerned with the description of the internal structure and the parts that make up the agent. The important attributes are defined and must be approved as a base to enable an agent to be independent and yet capable to adapt to and interact with the surrounding environment. It must also be capable to satisfy its need and make use of its interests and control of its beliefs.

2.5 Ability to Represent Agent Interactions

Agents in multi-agent systems need to communicate in order to plan and coordinate the work that needs to be done. This factor is concerned with the ability of the methodology in building and representing the communication between the agents in the system. It is also concerned with defining the techniques and the protocols used in performing such communication.

2.6 Agent Behavior Representation

This factor is concerned with the ability of the methodology to capture, represent and describe the
behavior of the agent. Agent behavior is realized through capturing the work or actions which must be done by the agent. It is also realized by the jobs that must be performed by the agent whenever changes take place in the surrounding environment. This factor also defines the techniques used in describing the agents’ behavior and how effective these techniques are. This factor is also concerned with the ability of the methodology in avoiding sudden and unexpected runtime actions, circumstances, and situations. This factor is also concerned with defining the method with which actions are performed or executed, i.e. sequential or simultaneous.

2.7 Software Development Life-Cycle Point of View

Software engineering remains insistence on a sequence of activities and steps which must be performed in the life cycle of a program. These activities and steps help analysts, developers and managers in system development. The ideal methodology must cover six important stages: Project Modeling, Analysis scope, Analysis requirements, Design, Implementation and Testing. The methodology which covers those stages in the development of a system is the methodology which most likely to be selected for that particular system. Reasons are its cohesiveness, synchronization and comprehensiveness. Using this factor, each of the previous methodologies is evaluated and for each methodology, the scope of comprehensiveness for each stage of the software life cycle is compared.

3. Evaluation and Comparison of Methodologies

In this section, we present a detailed evaluation of a number of methodologies and a comparison is made. The evaluation is performed using the proposed criteria mentioned above. We shall present a comparison of a three methodologies: Gaia, MaSE and HLIM.

3.1 Concepts

With regards to Gaia methodology, it strongly supports autonomy by the fact that the role encapsulates its functionality and is responsible for it. Gaia support reactivity through the liveness properties within the role's responsibilities. Gaia does not support clearly pro-activity. Gaia supports some degree of the sociality in its acquaintance model. Gaia does not support the use of mental attitudes (such as beliefs, desires, and intentions). Gaia supports communication by its own interaction protocols. Roles are supported by Gaia. Roles are used to make decisions and to represent autonomy.

With regard to MaSE methodology some agent properties are supported: some degree of autonomy and reactivity. MaSE does not support adaptability. MaSE does not support the use of mental attitudes (such as beliefs, desires, and intentions). In terms of support for communication, MaSE is probably the best with its protocol analyzer. Roles are supported by MaSE. Roles are used to identify the different types of agents the system will be composed of.

With regards to HLIM methodology also has some degree of autonomy and reactivity concepts. HILM also does not support adaptability. In terms of support the mental attitudes HLIM probably is the best in this concept. Also HLIM supports communication between agents by its own protocols. Roles are not clearly supported by HLIM.

However, all these methodologies do not specify the occurrence of events and the agent's reaction to these events.

3.2 Simplicity of Visualizing the System

Gaia lacks the existence of an initial and a primary phase which capture the Initial requirements. On the other hand, recognized techniques for describing systems behavior are not supported by Gaia. Gaia lacks tools which are known for high level visual display of system description. System description comes in the form of diagrams or notations that help in understanding systems.

In MaSE methodology, defining, understanding and visualizing the system is contained in the second stage of the analysis phase. Use-Cases are used which contain summaries of the main message exchanges in the initial flow of the system. To a certain extent, this method is not sufficient in describing the system behavior. The reason is that Use Cases are only used in the requirement analysis in order to help the customer understand the system structures from his point of view alone.

HLIM methodology makes easy to understand and visualize the system in its discovery phase. It is represented in the High Level Model which uses the Use-case maps notation. It recognizes the agents from other components in the system and their high level behavior. HLIM provides a high view of the system and provides a starting point to develop other model details in the system.
3.3 The Models

Some of the models and charts that Gaia uses are relatively good and some are insufficient. For example, the Role model is clear and easily understood. While the Interactions Model is weak and does not do the task it is supposed to do well when compared with other methodologies. Let us consider the ease with which models or diagrams are derived, and the ability of the methodology in mapping or traceability from one model to another. We found that Gaia methodology is good in this respect only in some of its models. For example, clear relations representing one-to-one are found. The transfer method between the Role diagram in the system and the Agents diagram is also good. But there are other charts that had no relations between them.

MaSE methodology is largely complex in organizing and ordering the diagrams and charts. The concurrent task diagram is derived from a number of charts: Pyramid Goal chart, Sequential chart and Role Chart. This method may cause some confusion and ambiguity to the analyst. While when compared, other methodologies have easier, more flexible and more direct ways in deriving models.

HLIM methodology is reasonably good from the point of ease of understanding and visualizing the system’s models. It uses easy to understand technologies like the use case maps to describe an agent’s behavior in a system. Also, there is no complexity in the models; they can be easily and flexibly extracted from one another. Relations can also be directly recognized. On the other hand, as far as the scope of coverage of the phases making up the system, it lacks the detailed design phase where there does exist a gap between the design and implementation phases.

3.4 Agent Attributes

Gaia methodology generally differs from all other methodologies. More specifically, it differs from the above mentioned methodologies in the internal structure of the agent. Gaia methodology depends on the Role during the analysis phase as a base for the agent. During the design phase, roles are replaced with agents. It can be stated that the internal structure of the agent in Gaia methodology is somewhat strong. The strength is due to the fact that every agent plays a specific role and is independent in making decisions. On the other hand, the agent behavior is not fully controlled but it can be controlled through responsibilities inside the organization.

In MaSE methodology, the agent’s internal structure is formed through the step of Assembling Agent Classes during the design phase. Designers have the freedom of choice in weather they wished to use the structure that they developed or to use a predefined and available structure like BDI. This indicates that this methodology did not make any suggestions related to the agent’s internal structure and its build up. Therefore, from this factor’s point of view, it is considered a weak methodology and requires a development in this point.

In HLIM methodology, the agent’s internal structure is represented and described by a number of properties like goals, plans and beliefs. This to a certain extent is considered excellent. Also, this construct supports the concept of BDI.

3.5 Ability to Represent Agent Interactions

Collaboration protocols between agents in Gaia methodology are weak, insufficient and are in need for development and improvement. Gaia only supports agent’s one-to-one interactions. It does not support simultaneous multiple agents interaction.

MaSE methodology describes conversations between agents through Finite State Machines. It is possible through this to achieve a dynamic level of message exchange between agents in a system. These finite machines lead to an algebraic description of conversations. Official mathematical proofs can be formulated from this algebraic description to describe and proof the interaction between agents in a system. For this reason this method is considered successful and acceptable in describing interactions between agents.

HLIM methodology describes conversations between agents through a conversation model which is extracted from the Agent Relationship Model and the Internal Agent Model. The model simply defines what messages are exchanged messages between agents. The exchanged messages implement dependencies and jurisdictional relationships as recognized in the agent Relationship Model. It defines the exchanged messages in general and not in details. This is one of the most important elements in the detailed design phase. Based on that, this methodology is found to lack detailed interaction protocols.

3.6 Agent Behavior Representation

With regard to this factor we can say that most discussed methodologies lack the existence of any technology or model which describe how the agent
behavior is represented, the actions that the agent is doing at the present time and also any changes in the environment around it.

3.7 Software Development Life-cycle Point of View

To a certain extent, all methodologies are acceptable at different levels when it comes to this important factor. For example, the methodologies Gaia, MaSE and HLIM cover most of the phases like the analysis and design phases, but they lack a detailed description of the implementation and testing phases. The HLIM methodology needs improvement and development in the detailed design model.

4. Conclusion

This paper presents a comparison between three known methodologies (Gaia, MaSE and HLIM) with respect to building and development of agents systems. The paper also discusses some of the basic concepts for these methodologies like agent structure, representation and visualization of the system, how an agent’s goals are obtained and controlled, and how the models are chained (or sequenced) and their extraction from each other, the ability of the methodology to represent the interaction between agents, ability to represent the agent’s behavior, the point of view of the program development life cycle through the criteria MASADF. The criteria are recognized from a study we carried out on agent oriented. Most methodologies are to a certain extent reasonable in their suitability with the MASADF criteria but they still need development and improvement.

Our future plan is to formulate a complete methodology specifically designed for developing agent systems. The new methodology should be comprehensive, derived and based on strengths and avoid weaknesses of existing methodologies.

6. References


