TEM Journal. Volume 7, Issue 2, Pages 433-438, ISSN 2217-8309, DOI: 10.18421/TEM72-27 May 2018.

Comparing the Artifacts of Agent Methodologies

Reem Abdalla [1], Alok Mishra [2]

[1] *Department of Modeling & Design of Engineering System, Atilim University, Ankara, Turkey*
[2] *Department of Software Engineering, Atilim University, Ankara, Turkey*

***Abstract* - Agent-oriented software engineering (AOSE) systems are often distinguished from each other based on intelligence, autonomy, and reasoning. In this recpect, agent methodologies aim to covering all important components needed to build software systems.**

The success of agent methodologies depends on the development of all the necessary processes within different software stages, such as analysis, design, implementation, testing and maintenance. This work introduces a comparative study of agent approches with the purpose to asseses and compare the development lifecycle processes of four different well-known AOSE methodologies, ADELFE, PASSI, Gaia and O-MaSE to define the similarities and differences between them. We can conclude that, the development of standard agent approach that supports Multi-Agent Systems (MAS) based has become an urgent need.

***Keywords* – Agent-Oriented Software Engineering; ADELFE; Gaia; PASSI; O-MaSE.**

1. Introduction

In the past couple of decades, Software Engineering (SE) has sought to create new software development phases and modeling languages in order to establish certain steps for constructing efficient software systems.This agent- oriented area has helped to reduce complexity and increase the quality of distributed software [1]. Nevertheless, traditional SE methods require that they adapt to the complexity and distribution of Multi-Agent Systems (MAS).

These modifications are essential because the basic methods of conventional SE do not provide activities to cover the complexity of MAS [2]. However, to assign the relationship between multiple roles and one agent is not trivial. Many design quality factors need to be considered [17]. Software quality has long been a critical issue for software developers [18],[19],[20]. Agent-Oriented Software Engineering (AOSE), has been suggested in recent years to integrate the fundamentals of software engineering and artificial intelligence.

One of the objectives of agent technology is to suggest methods and processes to construct agent systems [3]. The AOSE approach has its own charactreristics to present more elastic and strong projects for complex applications.

As a result, a big number of AOSE methods and processes have been suggested along with their methods and tools [4]. The main purposes of developing AOSE is to facilitate the process of developing software lifecycles and increase the software quality [7]. In order to do so, a standard approach is required. For the time being, there are no agreements on the construct agents or their modeling in the lifecycle development process, and different proposals for building agent-based systems are being presented in order to discover which properties are important in producing a good-designed approaches.

In this study, we select four prominent agent-oriented methodologies, ADELFE, Gaia, PASSI and O-MaS, which cover a large domain of software processes than the other methodologies, and provide methods to develop agent-oriented software systems.

2. Adelfe Methodology

There are six stages within ADELFE [5]: preliminary requirements, final requirements, analysis, design, implementation and tests. There are

DOI: 10.18421/TEM72-27
https://dx.doi.org/10.18421/TEM72-27

Corresponding author: Reem Abdalla,
Department of Modeling & Design of Engineering System, Atilim University, Ankara, Turkey
Email: *reemelsaetii@yahoo.com*

Received: 29 March 2018.
Accepted: 11 May 2018.
Published: 25 May 2018.

four definitions of work that need modifications based on Adaptive Multi-Agent System (AMAS) theory [13] [14]. The activities of the other steps in the Rational Unified Process (RUP) are the same [6].

2.1 ADELFE methodology and other three methodologies

A comparison among the ADELFE models with other selected approaches is shown in Table 1. In the following part, there is a list of ADELFE artifacts to determine the similarities between the ADELFE diagrams and the others.

• Use Case Diagram: In ADELFE, this activity which ends the requirements workflow consists of three steps: the first is design use cases, the second one is clarification of the linked sequence diagrams and the third one is to determine cooperation failures [5].

Table1. Similarity of ADELFE processes with the other three methodologies

ADELFE	PASSI	Gaia	O-MaSE
Use Case Diag.	Domain Requirements Description Diag.	×	Organizati -onal Diag.
Collaboration Diag.	×	×	×
Protocol Diag.	Protocol Description Diag.	Protocol Definition Diag.	Protocol Diag.
Class Diag.	Domain Ontology Diag.	×	×

- Domain Requirements Description Diagrams (PASSI): PASSI uses the conventional use case diagram to provide an overview of the system function.

- Organizational Diagram (O-MaSE): O-MaSE uses this diagram to model the relations and interactions between the organization and the actors of the system.

- Gaia does not have a diagram to represent this type of concept. As a result, there is no other diagram in Gaia to give a general view about the systems and their external components.

• Collaboration Diagrams: These are used to study the context of the system and describe the flow of data between the system and the actors.

-Gaia, PASSI and O-MaSE do not use this kind of diagram and, instead, use other diagrams to describe the system data flow.

• Protocol Diagram: It uses sequence diagram to describe the interactions among different agents in the target system.

-Protocol Description Diagram (PASSI): PASSI methodology has protocol description diagram, which is similar to the protocol diagram of ADELFE. Each of these has the same use. Therefore, they are considered as equivalents.

-O-MaSE protocol diagram offers different conversations among the system components[15],[16].

-Gaia has this diagram for interactions at the level of the former, but it cannot be considered equivalent to the ADELFE protocol diagram.

• Class Diagram: The determined agents are described in the initial class diagram. This diagram delivers various class diagrams in order to take the database designs into account.

- Domain ontology diagram (PASSI): This diagram is used in PASSI to display the knowledge related to individual agents and their relations.

- O-MaSE dose not have a diagram to demonstrate this kind of vision. It uses the class diagram for a different purpose.

- Gaia does not have the same type of agent classes.

3. Gaia Methodology

Gaia [10] separates the of software developing process into two various stages, analysis and design, the first phase intended to gather and organize the specifications as the foundation for the second phase of the computational organization.

3.1 Gaia methodology and the other three methodologies

This section shows the resemblance between Gaia and the other three methodologies. The base models and processes of the Gaia are compared with the concepts of other entrant approaches listed in Table 2.

Table2. resemblance of Gaia processes with the other three methodologies

Gaia	ADELFE	PASSI	O-MaSE
Protocol Definitions Diag.	Protocol Diag.	Protocol Description Diag.	Protocol Diag.
Role schema Diag.	×	Roles Identificati on Diag.	Role Diag.
Service Diag.	×	Task Specificatio n diag.	×
Agent Diag.	×	Agents Identificati on Diag.	×
Acquaintance Diag.	×	×	×

• **Protocol Definitions Diagram** (Gaia): The relations among the roles are depicted in the interaction model which is composed of a collection of protocol definitions, one for each interaction role [10].

-In ADELFE, PASSI and O-MaSE the concept of protocol is different than in Gaia, they represent the system interactions by models extracted from UML sequence diagrams. Each methodology has different interaction diagrams; for example, ADELFE depicts interactions between agents while O-MaSE shows those among roles. PASSI uses this concept to explore the tasks of each agent through role-particular scenarios.

•**Role schema Diagram**: The role schema diagram is used to determine role permissions and responsibilities in addition to the protocols and events in which they share.

- Roles Identification Diagram (PASSI): It demonstrates the paths of the agents' identification diagram including agent interactions [11]. Even though both diagrams have the same purpose, they offer different views of the system.

- Role Diagram (O-MaSE): It presents this concept to guarantee that all system roles are determined to fulfill the objectives determined in the goal model [15]. The external actors' concept is not present in the role schema diagram of Gaia.

-ADELFE does not offer the models included in the role schema diagram of Gaia.

•**Agent Diagram**: In Gaia this diagram is used to define the agents and specifies their types to create the target system.

- Agents Identification Diagram (PASSI): The objective of this process is to model the agents' responsibilities by UML standard packages.

- In ADELFE, the concept of agent definition is different than in Gaia, it specifies another type of diagrams to describe the agent of the system.

- O-MaSE, there is not a particular model to depict the artifact involved in the agent diagram of Gaia, because O-MaSE provides different diagrams.

• **Service Diagram**: This diagram shows the services that are offered by each kind of agent. The agent tasks are extracted from the processes of the agent roles.

- Task specification diagram (PASSI): There is a specific step in the PASSI methodology to represent agents' tasks. Agent's tasks depend on their own capacity.

-ADELFE and O-MaSE do not have a similar kind of service diagram and, instead, deliver different models.

• **Acquaintance Diagram**: Based on [10], this model permits the developer to know the type of relations among different kinds of agent systems. The other three selected methodologies do not use a similar concept.

4. Passi Methodology

This methodology [11] [12] consists of four models: system requirements, agent society, agent implementation, and code models, with each stage having different activities.

4.1 PASSI methodology and the other three methodologies

Table 3. shows the commonalities between PASSI and the other three competitive approaches.

•**Domain Description Diagram**: PASSI uses traditional use-case diagrams to provide holistic view on the execution environment and the domain knowledge

Table 3. resemblance of PASSI processes with the other three methodologies

PASSI	ADELFE	Gaia	O-MaSE
Domain Description Diag.	Use Case Diag.	×	Organizational Diag.
Agent Identification Diag.	×	Agent Diag.	×
Roles Identification Diag.	×	Role schema Diag.	Role Diag.
Protocol Description Diag.	Protocol Diag.	Protocol Definitions Diag.	Protocol Diag.
Task Specification Diag.	×	Service Diag.	×
Roles Description Diag.	×	×	Agent Class Diag.
Structure Definition Diag.	×	×	×
Domain Ontology Diag.	Class Diag.	×	×

-Organizational Diagram (O-MaSE): O-MaSE methodology has organizational diagram which is similar to the domain description diagram of PASSI.

Each one of them have the same function, which means that they are equivalent.

– Use case diagram (ADELFE): This diagram in ADELFE is a little more accurate than the domain description diagram of PASSI, because it shows the cooperation failures among the system and its environment to assist developers to determine problems and none cooperative cases.

– Gaia: This approach does not address the domain description concept.

•**Agent Identification Diagram:** This model specifies responsibilities and interactions between agents and their environments.

- Agent diagram (Gaia): This model in Gaia has a different objective, it just used to show the agents in the system and their types without going into details of their relations.

-ADELFE and O-MaSE do not deliver such kind of agent models.

•**Roles Identification Diagram**: The importance of this model lies in discovering and displaying the tasks assigned to each agent in the system[12].

-Role schema diagram (Gaia): As mentioned before, Gaia uses this concept only to show the agents' role and their properties.

-Role Diagram (O-MaSE): Although this diagram uses different notations, it has the same objectives as the roles identification diagram.

-ADELFE does not address the concept of roles.

•**Protocol Description Diagram**: This diagram specifies the interactions among system compnents by using sequence diagrams to display all agent’s functions [8].

-Protocol Diagram (ADELFE): In this approach the main purpose of this process is to represent the system agents' conversations, thus making them equal to the protocol description diagram of PASSI.

- Protocol Diagram (O-MaSE): This methodology uses this type of diagrams to depict the internal specifics of each protocol defined in the previous steps. Therefore, we can say that both have the same purpose with different processes and notions.

- Gaia supports this kind of concept, but in a different diagram.

•**Task Specification Diagram**: PASSI used this model to illustrate agent capability, while ignoring information about the roles played by the agent when conducting certain functions.

- Service diagram (Gaia): Despite service diagram has the same purpose of task specification diagram, each one of them present different views of the system.

- ADELFE works with different artifacts.

- O-MaSE does not have a similar diagram.

•**Roles Description Diagram**: In PASSI this diagram demonstrates the roles of agents and the tasks include communication capabilities and agent relations.

- Agent Class Diagram (O-MaSE): This diagram models the agents and their roles. It describes interactions between the agents and the organization. The roles description diagram in PASSI represents agents and roles without taking into account its organization.

- ADELFE and Gaia do not specify any diagram to show the roles description concepts.

• **Structure Definition Diagram**: This model in PASSI delivers the comprehensive view of the system and discovers all the properties of the agent class together with internal task classes by using two different types of class diagrams. ADELFE, Gaia and O-MaSE do not cope with such type of concepts.

• **Agent Behavior Description Diagram:** There is a specific stage in PASSI to describe the flow of events and agent connections through calls and exchange of messages. The other three methodologies do not provide such diagrams.

• **Domain Ontology Diagram**: The goal of domain ontology diagram is to characterize the ascribed knowledge of individual agents and their relations according to class models.

- Class diagram in ADELFE has exactly the same goal of the domain ontology diagram, however it can not only depict inter-agent relationships, but also relationships between all components of the system.

- Gaia and O-MaSE do not present this type of diagram.

5. O-MaSE Methodology

O-MaSE [15][16] developed from a collection of fragments which offered by designers to achieve the requirements required for their target system. O-MaSE clearly defines processes and tasks and enables method engineers to arrange processes in various methods according to project requirements. It does not make any commitments to a predefined set of stages.

5.1 O-MaSE methodology and other three methodologies

The commonalities between PASSI and the other participating methodologies are shown in Table 4. Below, the basic models of O-MaSE methodology are displayed to determine their tasks, and a comparison with the models of the other selected agent methodologies.

Table 4. Similarity of O-MaSE processes with the other three methodologies

O-MaSE	ADELFE	Gaia	PASSI
Goals Diag.	×	×	×
Domain Diag.	×	×	×
Organizational Diag.	Use Case Diag.	×	Domain Description Diag.
Role Diag.	×	Role schema Diag.	Roles Identification Diag.
Agent Class Diag.	×	×	Roles Description Diag.
Protocol Diag.	Protocol diag.	Protocol definition Diag.	Protocol description Diag.
Plan Diag.	×	×	×

•**Goal Diagram:** Based on [15]**,** the objectives of the organization are depicted as a behavioral goal hierarchy; involves the goal characteristics and the relationship of precedence.

-ADELFE, PASSI and Gaia methodologies are not as well adapted to deal with this purpose.

• **Domain Diagram:** In O-MaSE requirements analysis stage, the domain model is an important part of requirement analysis stage. It determines which language can be used by the developer to ensure that everyone is talking about the same subject. ADELFE, Gaia and PASSI do not have such a diagram.

• **Organizational Diagram:** This model is used to describe the connections among the organization and the external actors**.**
-Domain Description Diagram (PASSI): This model in PASSI has the same organizational diagram purpose in O-MaSE.

-Use Case Diagram (ADELFE): ADELFE applies use case diagram to help developers to determine problems and non-cooperative situations.

- Gaia does not deal with this type of diagrams.

•**Role Diagram:** O-MaSE uses this structural diagram to model the roles, the objectives they accomplish and relations among target system roles.

-Role schema Diagram (Gaia): Gaia uses this concept differently from the way it uses the role schema diagram to give a complete description to each role.
- PASSI has a different diagram to cover the concept of role.

- In ADELFE there is no specific model to define the agent roles.

•**Agent Class Diagram**: Based on [15] and [16] an agent class diagram depicts the classes of agents and sub-organizations that set up the organization.

-Roles Description Diagram (PASSI): It intends to guarantee that all the desired roles are determined to appear in the functions that determine the role's conduct and the interactions between agent systems [11].

-ADELFE and Gaia do not have such diagrams.

•**Protocol Diagram:** This structural diagram in O-MaSE is used to define the indoor specifics of every protocol represented in the role and agent class diagrams**.**

-ADELFE and PASSI use sequence diagram to demonstrate the interactions among different system components.

-The Protocol definition diagram in Gaia is used to describe the agents' system roles interactions [9].

• **Plan Diagram:** The plan diagram in this approach defines the plan for each kind of goal that an agent can accomplish by using a sequence of exchange messages.

-There are no such diagrams in the other three methodologies to model the same type of artifact.

6. Conclusion

This work introduces the processes and models of ADELFE, Gaia, PASSI, and O-MaSE through comparison. By studying and the analyzing the selected methodologies, we distinguish that each methodology has a different concepts and particular characteristics.

Whilst ADELFE and PASSI have diagrams that present processes to engineering and evolve multi-agent societies' merging designs, Gaia focuses on the context of the agents. O-MaSE focuses on engineering the processes for MAS. In our view, it is not easy to define a methodology for developing any project. Despite various agent methodologies, processes, models, AOSE remains at a premature phase of development. It is the time to start development of the next generation AOSE methodology, towards a standardized AOSE methodology.

References

[1]. Jayatilleke, G. B., Padgham, L., & Winikoff, M. (2006, May). Evaluating a model driven development toolkit for domain experts to modify agent based systems. In *International Workshop on Agent-Oriented Software Engineering* (pp. 190-207). Springer, Berlin, Heidelberg.

[2]. Cunha, R., Adamatti, D., & Billa, C. (2015, October). Agent oriented software engineering: a comparative study between methodologies that support the development of multi-agent systems. In *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems* (pp. 48-52). ACM.

[3]. Guedes G. T. A.(2012). Um metamodelo UML para a modelagem de requisitos em projetos de sistemas multiagentes. PhD thesis, Universidade Federal do Rio Grande do Sul.

[4]. Bergenti, F., Gleizes, M. P., & Zambonelli, F. (Eds.). (2006). *Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook* (Vol. 11). Springer Science & Business Media.

[5]. Bernon, C., Gleizes, M. P., Picard, G., & Glize, P. (2002). The Adelfe Methodology For an Intranet System Design. *AOIS@ CAiSE*, *57*.

[6]. Jacobson, I., Booch, G. & Rumbaugh, J.(1999) The Unified Software Development Process. Addison-Wesley.

[7]. Wooldridge, M., Jennings, N. R., & Kinny, D. (1999, April). A methodology for agent-oriented analysis and design. In *Proceedings of the third annual conference on Autonomous Agents* (pp. 69-76). ACM.

[8]. Cabri, G., Leonardi, L., & Puviani, M. (2007, June). Service-oriented agent methodologies. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. WETICE 2007. 16th IEEE International Workshops on* (pp. 24-29). IEEE.

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

[10]. 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.

[11]. Cossentino, M. (2005). From requirements to code with the PASSI methodology. *Agent-oriented methodologies*, *3690*, 79-106.

[12]. Cossentino, M., & Potts, C. (2002, June). A CASE tool supported methodology for the design of multi-agent systems. In *International Conference on Software Engineering Research and Practice (SERP'02).*

[13]. Capera, D., Georgé, J. P., Gleizes, M. P., & Glize, P. (2003, June). The AMAS theory for complex problem solving based on self-organizing cooperative agents. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on* (pp. 383-388). IEEE.

[14]. Capera, D., Georgé, J. P., Gleizes, M. P., & Glize, P. (2003, April). Emergence of organisations, emergence of functions. In *AISB'03 symposium on Adaptive Agents and Multi-Agent Systems* (pp. 103-108).

[15]. DeLoach, S. A., & Garcia-Ojeda, J. C. (2014). The o-mase methodology. In *Handbook on Agent-Oriented Design Processes* (pp. 253-285). Springer, Berlin, Heidelberg.

[16]. DeLoach, S. A., & Garcia-Ojeda, J. C. (2010). O-MaSE: A customizable approach to developing multiagent development processes. *International Journal of Agent-Oriented Software Engineering*, *4*, 244-280. doi:10.1504/IJAOSE.2010.036984.

[17]. Yu, L., & Mishra, A. (2013). Multiple-Role Agent Based Distributed Computing. *Technics Technologies Education Management*, *8*(1), 238-243.

[18]. Mishra, D., & Mishra, A. (2010, October). A software inspection process for globally distributed teams. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 289-296). Springer, Berlin, Heidelberg.

[19]. Mishra, D., & Mishra, A. (2009). Simplified software inspection process in compliance with international standards. *Computer Standards & Interfaces*, *31*(4), 763-771.

[20]. Mishra, A., & Mishra, D. (2006). Software quality assurance models in small and medium organisations: a comparison. *International Journal of Information Technology and Management, 5*(1), 4-20.