ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

A REVIEW ON AGENT ORIENTED SOFTWARE ENGINEERING

Aanchal Bawa[1], Sonam Bhatia[2], Varinder Kaur Attri[3]

Master of Technology, Department of Computer Science and Engineering, RC GNDU, Jalandhar, India[1,2]

Assistant Professor, Department of Computer Science and Engineering, RC GNDU, Jalandhar, India [3]

Abstract: Agent oriented software engineering is one of the important areas of software engineering which deals with agents and multi agents. These agents are the intelligent, robust modules specialized to perform work in the area it is being specialized. There exist various methodologies to design and execute them having various characteristics to support different types of agents. The paper concentrates on analyzing the limitations of different methodologies and solutions to them so as to have more reliable and efficient methodology.

Keywords: Gaia, MaSE, Message, Prometheus.

I. INTRODUCTION

Agents are powerful technological modules with many important applications. Software's required in industry are complex in nature. They are typically composed by integrating a large number of components that have many interactions. Moreover, this complexity is not accidental; it is an innate property of the software according to the task due to which it is composed and being used. Thus it can be said that the software engineering is required to identify some technique to reduce complexity of such software. The emergence of multi-agent systems has brought together many techniques in an effort to build distributed applications, intelligent, and robust.[1] In recent years, there have been several attempts of creating tools and methodologies for building such systems. Unluckily, there does not exist such efficient methodologies which focus on multi-agent architecture or give detailed view to support of complex systems.

The different key agent-oriented methodologies: Gaia, MaSE, MESSAGE, Prometheus and Tropos have been compared by performing trait analysis. This is done by studying the strengths and weaknesses of each methodology and combining their strengths for the development of new extensions. The four major areas of an agent-oriented methodology: modeling language, concepts, process and pragmatics have been checked. The Gaia, MaSE and TROPOS are the only methodologies which allow us to exploit all the flexibility provided by Agent Oriented Programming and considered as a useful approach to follow in software development.

A. Gaia

The Gaia methodology is both general, which is applicable to multi-agent systems (MAS), it covers both the macro (social) and micro (agent) aspects of systems.[3] The modified Gaia now uses the goal model, but the original Gaia does not use goal explicitly, but its responsibility and safety conditions can be seen as goals.[6] Gaia is useful to adapt methodology to be compatible with large –scale collection of standards.[1] While having these factors, Gaia has some under given limitations:

- Agents cannot share common goals.
- It does not attempt to deal with systems while acting as a actor.
- The cohesion between protocols is poor to some extent.
- Component of system may enter or leave at run time in open or dynamic systems which are unknown to designer and thus creates problem at design time while creating classes and system.
- The revised Gaia discusses goals, but checking the identification of goals is still out of scope.
- Gaia does not use case scenarios.
- There was no particular standard used to design Gaia.

B. Multiagent Systems Engineering (MaSE)

MaSE is an agent-oriented software engineering methodology that is an extension of Object-Oriented approach, thus it can be also said O- MaSE. In this methodology, each agent acts as a software process which communicates with another agent or software process to achieve a common overall goal. Although some agents are intelligent and designed efficiently while some are not, all are treated equally that is they all work in same manner to achieve desired goal.[4] The main objective of MaSE, as a software engineering methodology is to enhance the complete life cycle methodology so as , system developers can design and develop agents and multi agent system efficiently.[4] It uses case scenarios. O- MaSE can also be stated as environmental and deployment model.[6]

It has certain limitations:

- Some of the software applications are closed.
- The methodology does not support open systems, thus agents cannot be created, destroyed or moved during execution.
- Only one to one communication is allowed between agents.
- The system with is considered small.
- It does not explicitly define a use case template.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

C. MESSAGE

The methodology covers MAS (Multi Agent System) analysis and design. It is particularly used in mainstream software engineering departments. The MESSAGE notation combines the Unified Modeling Language with agent knowledge-level concepts and diagrams with notations for viewing them. The MESSAGE analysis and design process is based on the Rational Unified Process (RUP). As it is known that agent is a robust, intelligent module which requires flexibility, an interacting environment, a problem solving approach and heterogeneous and dynamic environment so agents can be created, destroyed and moved during runtime, but this methodology support these features partially which makes it inefficient to be used and does not required for practical purposes.[1] The limitations of the methodology are:

- It lacks to support various primary features of a methodology to design agents and multi agents.
- It is no longer over active development.

D. PROMETHEUS

Prometheus is used to develop intelligent agent systems.[5] Agent system can be developed by interacting modified version of Prometheus and agent oriented software which are partially implemented in JACK (visual modeling tool). This methodology uses case scenarios. It is an environmental model and it represents the environment better than other methodologies. Formation of agent types can be guided to data coupling. The agents in this methodology are not single entity, they are composed of smaller entities either roles or capabilities. The advantage of Prometheus methodology is this that it supports both dynamic and static models for individual agents.[6]

E. TROPOS

Tropos is based on two principles: First, the agents and their related notions such as goals, plans are used in all phases such as from early requirement to actual implementation in software development cycle. Secondly, Tropos covers also the early stages of requirements analysis, permitting inner understanding of the environment where software is being developed and used, and the type of communications which are required to occur between software and human agents. [5]

It has some limitations:

- This methodology is not designed to support a particular type of software.
- Sophisticated software agents designed for achieving particular goal or extended reasoning are not supported by Tropos methodology.
- It does not use case scenarios.

II. METHODS OF COMPARISON OF DIFFERENT METHODOLOGIES

Various types of comparisons have been made by researchers and engineers based on different parameters such as criteria related to the process, the steps and techniques related criteria, measures and criteria for usability, model related and "concepts" related criteria, the comparison with respect to the model related criteria and comparison with respect to the support related criteria. All these differences cover almost all the traits of these methodologies such as application development life cycle support, the coverage of life cycle, development approach, the type of application domain, the nature of the agent, ease of understanding of the stages of development, etc. It is difficult to check the best methodology among all as almost as methodologies are application oriented and there does not exist any methodology which supports all type of agent based applications.[6] However, the analysis framework is based on four criteria:

- Model Related Criteria
- Technique Related Criteria
- Process Related Criteria
- Supportive Feature Related Criteria

III. LITERATURE SURVEY

Chia-En Lin, Krishna M. Kavi(2005)[7] explored that various applications of Agent-based systems categorized into different application domains. The paper describes what properties are required to form an Agent society with the purpose of achieving system-wide goals in MAS. A baseline is developed to focus on the core of Agent concepts throughout the comparative study and to investigate both the Object-Oriented and Agent-oriented techniques that are available for constructing Agent-based systems. In each respect,address the conceptual background associated with these methodologies and how available tools can be applied to provide to specific domains.

Sukhvir Singh, Prachi, Richa Setiya[1] (2012) states that Agent oriented software engineering has numerous applications in different areas such as information management, space exploration, air traffic control, electronic commerce, business process management, defense simulation etc. A growing number of agents adopting software engineering methodologies have been proposed in recent years. The purpose of these methodologies is to provide models, methods, tools and techniques so that the development of softwares can be achieved in a systematic way. Even a large number of methodologies for agent-oriented software engineering are developed; a complete agent-oriented methodology for developing agent systems is still absent. In this paper, we explore the various applications of Agent-based systems categorized into different application domains.

Hoa Khanh Dam, Michael Winikoff(2012)[6] says that numerous methodologies for developing agent-based systems have been proposed in the paper. This proliferation creates a challenge to practitioners who need to select a methodology to adopt. This situation is analogous to that of object-oriented methodologies and notations pre-UML, and we argue that the time is ripe to begin the development of a next generation agent-oriented software engineering (AOSE) methodology, leading ultimately towards a unified AOSE methodology. This paper proposes process and models for a next generation AOSE methodology. Our proposal is based on a

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

comparative analysis of seven prominent AOSE methodologies, which identified strengths, weaknesses, commonalities and differences.

Fabiano Dalpiaz, Ambra Molesini, Mariachiara Puviani[8] explained that many different methodologies have been proposed in Agent Oriented Software Engineering (AOSE) literature, and the concepts they rely on are different from those adopted when implementing the system. This conceptual gap often creates inconsistencies between specifications and implementation. The paper proposes a meta model-based approach that aims to bridge this gap, resulting in an integrated meta-model that merges the best aspects of four relevant AOSE methodologies (GAIA, Tropos, SODA and PASSI). The meta-model assembly followed a well defined process: for each methodology to be integrated in the meta-model, we elicited the requirements, identified a set of process fragments, thoroughly compared the concepts belonging to the various fragments, and finally composed the meta-model.

Hyacinth S. Nwana, Divine T. Ndumu[9] This paper sets out, ambitiously, to present a brief reappraisal of software agents research. Evidently, software agent technology has promised much. However some five years after the word 'agent' came into vogue in the popular computing press, it is perhaps time the efforts in this fledgling area are thoroughly evaluated with a view to refocusing future efforts.. The paper contains some strong views not necessarily widely accepted by the agent community. Mehdi Dastani, Joris Hulstijn[10] Methodologies for multi agent system development should assist the developer in making decisions about those aspects of the analysis, design and implementation, that are crucial for multi agent systems, namely, social and cognitive concepts (e.g. norms and goals). In this paper, review of existing agent-oriented methodologies is done. We conclude that there is a big gap between the analysis and design models and the implementation. Some open issues for multi agent system development are identified. The paper introduces vision of researchers while a development of a methodology for multiagent systems, based on the OperA analysis models and the agent-oriented programming language.

IV. FUTURE SCOPE

As per the previous researches it has been concluded that each methodology is designed for a specific purpose or to design a particular type of agent oriented system but each methodology has its drawback which is very important to remove so as to design a perfect agent or multi agent system. There does not exist even a single methodology which supports all agent based applications. The main aim is to detect the limitations and reduce them so as to design a more intelligent, robust system and a methodology to support all agent based applications.

V. CONCLUSION

The paper concentrates on five different methodologies of AOSE (Agent Oriented Software Engineering) such as Gaia, MaSE, Message, Prometheus and Tropos and their limitations. Each methodology is application oriented and supports only particular type of softwares. Although there exists various critireas to compare these methodologies but still there is no best methodology which supports all agent based applications. These faults are required to remove so as to have reliable AOSE.

ACKNOWLEDGMENT

I am thankful to the almighty for the grace through which I am able to write this paper and no doubt the support of my teachers, family and friends is the one the best assets without whom I can't imagine to complete my paper and get it published.

REFERENCES

[1] Sukhvir Singh, Prachi, Richa Setiya, Evaluation of Agent Oriented Software Engineering (AOSE) Methodologies-A review, International Journal of Latest Research in Science and Technology Vol.1,Issue 2 :Page No.94-97 , July .August (2012)
[2] Jayatilleke, Gaya, Padgham, Lin, Winikoff, Michael, Component Agent Framework for non-Experts (CAFnE) Toolkit. (2005)
[3] Wooldridge, Michael, Jennings, Nicholas R., Kinny, David. A methodology for agent-oriented analysis and design, AGENTS '99 Proceedings of the third annual conference on Autonomous Agents (1999)
[4] Deloach, Scott A., Wood, Mark F., Sparkman, Clint H, Multi agent Systems. Engineering.(2001)
[5] Bresciani, Paolo, Perini, Anna. "Tropos: An Agent-Oriented Software .Development. Methodology,(2004)
[6] Hoa Khanh Dam; Michael Winikoff, Towards a next-generation AOSE methodology, Science of Computer Programming, 78 (2013) 684-694. doi:10.1016/j.scico.2011.12.005 (2013)
[7] Chia-En Lin, Krishna M. Kavi, A Methodology to Evaluate Agent Oriented Software Engineering Techniques(2005)
[8] [8]Fabiano Dalpiaz, Ambra Molesini, Mariachiara Puviani : Towards filling the gap between AOSE methodologies and infrastructures: requirements and meta-model
[9] Hyacinth S. Nwana, Divine T. Ndumu, A Perspective on Software Agents Research
[10] Mehdi Dastani, Joris Hulstijn, Issues in Multiagent System Development.

BIOGRAPHY

Aanchal Bawa received Bachelor's degree in Computer Science from Punjab Technical University, Jalandhar, India. She is pursuing her Master degree from Guru Nanak Dev University, Amritsar, India under the batch 2013- 2015. Her interest areas includes Software Engineering and her research areas includes Agent oriented software engineering and its methodologies.