A Study of Tools Used in Implement Agent Oriented Software Engineering

Aanchal Bawa, Varinder Kaur Attri

Researcher, Dept. of Computer Science, RC GNDU, Jalandhar, India

Associate Professor, Dept. of Computer Science, RC GNDU, Jalandhar, India

ABSTRACT*:* Agent oriented software engineering is a field which deals with agents and multi agents. There exist various tools to virtually implement the agents such as notation tools, formal tools and case tools. As there exist various methodologies to their corresponding implementing tools. The paper represents the different methodologies and the tools used to implement them. Although methodologies have their tools but still there exists various shortcomings to show the exact behavior of the agent. So this paper review on a tool named Petri net which is used to represent formal behavior of agents in more concise way so that their timing, their communication and working can been seen more clearly.

KEYWORDS: Petri Net, Methodologies, AOSE

I. INTRODUCTION

Agents are powerful, robust elements designed to work in intelligent manner to accomplish tasks which are not even programmed. According to software engineering point of view while designing agents various phases are considered: analysis, design, implementation and testing.

Analysis: This phase defines the roles of the agents, agent concept used by specific languages and the case tools for supporting languages.

Design: This phase defines the designing of agents and agent development environment and thus a multi agent can developed from both.

Implementation: Different approaches to implement agent oriented technology are specified so that the best methodology and agents can be designed with specific languages, frameworks and the supporting tools.

Testing: This phase defines the various testing techniques to check agents and multi agent systems are checked to know that whether they are developed as per the initial requirements if the requirements are fulfilled then multi agent system is as per required otherwise it is required to be re designed.[1]

There exist various types of tools defining their corresponding functions.
I.   Notation Tools: These are the tools which are used to represent the outcomes of every development phase of software for example entity relationship diagrams, flow charts, equations etc.
II.   Formal Tools: These are the tools which are used to prove properties of software before developing it fully for example lambda and pi calculus, Petri Nets, Unity, Z etc.
III.   Computer aided software engineering (CASE) Tools: These are the tools which are used to facilitate activities and these are based on notation and models such as Simulators, rapid proto typing, code generators etc.[2]

II. METHODOLOGIES WITH CORROSPONDING TOOLS

Agent Oriented Software Engineering includes various methodologies so as to design best and reliable agents and multi agent systems. Each methodology has its own properties and corresponding tool to check its behavior and to implement it. Here given the different methodologies with their characteristics and the corresponding tools.

A.   Tropes

This methodology is used to develop complex, distributed system. It form agents by grouping smaller entities (Roles and Capabilities), thus it is a requirement driven approach in which goals and demands are being analyzed before designing the required software.[3] The classes of applications designed in Tropes collectively form a single application having distributed system of management system in it. The different corresponding tools designed for Tropes are

 i.   TAOM4e: Model driven approach is supported for agent oriented software development.
 ii.   T- Tool: Model Checking of Tropes is performed.
iii.   GR- Tool: Formal reasoning is supported on goal models.
iv.   JADE: It is the extension of TAOM4e, which is designed to generate automatic code after Tropes specification.[4]
 v.   eCAT: It supports a goal oriented approach which is used for automatic testing of MAS.[5]

B.   Multi agent Systems Engineering (MaSE)

It was proposed by Deloach.[6] 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. [7] The tool supported by MASE is agent Tool III. It is the JAVA based tool which provides environment in which user can analyze, design and implement multi agent systems. It produces predictive metrics for performance so as to design intelligent agents and generate code. agent Tool III is developed using JAVA 1.5 with Eclipse 3.4[8]

C.   PROMETHEUS

Prometheus is used to develop intelligent agent systems. [9] 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. [3]

D.   INGENEAS

This is quite similar to Prometheus methodology, the only difference exist is that it not only decomposes goals but lay emphasis on resources used, inputs and output and there exist a development kit supported by Ingeneas to develop multi agent through Ingeneas methodology.[3]

E.   Process for Agent Societies Specification and Implementation (PASSI)

After a long research and study, PASSI has been generated.[11] This methodology is designed to remove various flaws in the existing methodologies. The major aim to produce this methodology is to improve the formal representation of agents in terms of robotics. The different 5 models applied on this methodology having their own phases are given as under:

i)   System Requirements
ii)   Agent Society
iii)   Agent Implementation
iv)   Code Model
v)   Deployment Model[13]

The tool designed specifically for PASSI is Rational Rose so as to check design pattern, it is a UML based CASE tool. For the code generation phase, FIPA (Foundation For Intelligent Physical Agents)[10] architecture is used in which XML based tools are used to communicate between agents and content is directly fetched from design.[12]

III.   PROCEDURE TO SELECT METHODOLOGY FOR A TASK

While selecting a methodology, various selection methods are applied. At the starting stage they are being shortlisted (such as short listing of job applicants) by applying under given constraints:

A.   Documentation: The methodologies which are selected are explained in detail to know their every bit so as to choose the best among them.
B.   Maturity: The methodology selected must be verified by the people who have to work on it and the needed updates are being checked along with the time refinement.
C.   Tool Requirement: Some methodologies have their own corresponding tools and some have don't, an ongoing tool is required to accomplish the given task.

Thus, the methodologies having no tool are not efficient to design an accurate system. So as to have accurate formal analysis and behavior analysis a tool named Petri Net is used.

IV.   COLORED PETRI NETS

Petri Net is a tool which provides graphical and mathematical presentation of agents based on bipartite graph, having same properties with Task Graphs. It shows the communication of concurrent, asynchronous, synchronous, parallel, non parallel forms of behavior. [14] Colored Petri net is a extension to classical Petri nets. [15] The various reasons for choosing colored petri nets to implement various systems are given as under:
i)   Colored Petri net is used for formal representation of agents and various types of analysis can also be performed.
ii)   It can be executed properly and can be used for simulation, rapid prototyping, gaming etc.
iii)   Colored Petri net tools are supported by Colored Petri net language.[16]
Through CPN (Colored Petri Net), the agents represented in them can change state by applying them on different transitions and by applying different rules on them with different joins.[18] The various applications used to implemented through CPN includes various application domains such as communication protocols, data networks, distributed algorithms and embedded systems.[17]

V.   LITERATURE SURVEY

Piermarco Burrafato, Massimo Cossentino(2003) compared the PASSI methodology with GAIA and MaSE to check the difference of efficiency of methodologies with tools and without tools by implementing a bookstore system . PASSI methodology uses RATIONAL ROSE and FIPA for designing architecture and code generation respectively. PASSI implements the bookstore system more clearly than other because it have its own implementing organized tools.

Chia-En Lin, Krishna M. Kavi(2005)[6] 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.

M. Morandini, D. C. Nguyen (2010) [4] analyzed the different tools for Tropos methodology and implement a conference system with it. The same system is analyzed and implemented with different methodologies and the conclusion conculuded is that there exists various tools for each methodology and each implements the system with its own way but still there exists various short comings such as Tropos required to become Goal oriented testing methodology with more flexible integration technique so as to implement system more clearly.

Hoa Khanh Dam, Michael Winikoff(2012)[3] 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 comparative analysis of seven prominent AOSE methodologies, which identified strengths, weaknesses, commonalities and differences.

Vijay Gehlot, Carmen Nigro(2010)[18] The paper describes the Colored Petri net, it's origin with classical petri net and how classical petri net combines with high level petri net so as to have colored petri net to perform better simulation. It explains important features of Petri net including its state space techniques, time division methods and hierarchy formation. Practical use of Colored Petri net and it's combination with Colored Petri tools which generates diagrams. Thus Colored Petri Net tool is defined and introduction to CPN tools are defined.

Anthony Spiteri Staines(2013)[14] Task graphs are quite similar to Petri net tool in which hierarchy of events are organized .Logic gates and concepts of Or and Xor gates are used. In Task graph each node is assigned with task and in petri net these task graphs are simulated by putting transitions to them and the movement of task is shown with every step.

VI. FUTURE SCOPE

As per the previous research, it has been concluded that there exists various methodologies and some methodologies have some tools concerned with them which help to simulate the system and some have don't. The methodologies with tools are more efficient to represent system and check formal behavior of agents. But some methodologies have tools only to design the system not express the behavior. So to check the working and communication between agents a tool named Petri net is used. To express behavior of agents more clearly every methodology needs to design a appropriate or a tool supporting for all methodologies must be designed. Thus there is a great need of a powerful tool to represent complex system implemented from agent oriented software engineering.

VII. CONCLUSION

The paper represents the methodologies (the method and rules) required to design a system in agent oriented software engineering. There are some methodologies which have their own tools but some don't have tools concerned with them. So to represent formal behavior of agents a tool named petri net is used which have some good features including state space diagrams, the hierarchical representation and timely pattern of agents. Although Petri net is a good tool to represent behavior but it can enhanced more to have deeper view of behavior of agents in complex system.

ACKNOWLWDGEMENT

REFERENCES

[1]Jorge J. Gomez-Sanz, Marie-Pierre Gervais, Gerhard Weiß, "A Survey on Agent-Oriented Software Engineering Research"
[2]Franco Zambonelli, "Agent Oriented Software Engineering", April 2008
[3] Hoah Khanh Dam, Michael Winikoff, "Towards a next generation AOSE methodology", January 2012
[4] M. Morandini, D. C. Nguyen, A. Perini, A. Siena, and A. Susi, "Tool-supported Development with Tropos: The Conference Management System Case Study" (2010)
[5] http://sra.itc.it/people/kundo/ecat
[6] Chia-En Lin, Krishna M. Kavi, Frederick T. Sheldon and Thomas E. Potok, "A Methodology to Evaluate Agent Oriented Software Engineering Techniques", The International Journal of Multiagent and Grid Systems, Nov. 2005
[7] Deloach, Scott A., Wood, Mark F., Sparkman, Clint H, Multi agent Systems Engineering. (2001)
[8] http://agenttool.cis.ksu.edu/
[9] Bresciani, Paolo, Perini, Anna. "Tropos: An Agent-Oriented Software Development Methodology, (2004)
[10] Piermarco Burrafato, Massimo Cossentino," Designing a multi agent solution for bookstore with the PASSI methodology" , (2003)

[11] Chella, A., Cossentino, M., Infantino, I., and Pirrone, R. An agent based design process for cognitive architectures in robotics in proc. Of WOA'01 (Modena, Italy, Sept. 2001).

[12] FIPA Abstract Architecture Specification (Refinements). Foundation for Intelligent Physical Agents, Document FIPA PC00094 (2001). http://www.fipa.org/specs/ fipa00094/PC00094.html

[13] Massimo Cossentino, Colin Potts, "A CASE tool supported methodology for the design of multi-agent systems", (2003)

[14] Anthony Spiteri Staines, "From Task Graph to Petri Nets", International Journal of Emerging Technology and Advanced Engineering",ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 5, (May 2013)

[15] L.M. Kristensen, S. Christensen, and K. Jensen," The Practitioner's Guide to Coloured Petri Nets." , International Journal on Software Tools for Technology Transfer, (2004)

[16] M. Pesic and W.M.P. van der Aalst, "Modeling Work Distribution Mechanisms Using Colored Petri Nets", [2007]

[17] Jensen, K., and L.M. Kristensen, " Coloured Petri Nets. Modelling and Validation of Concurrent Systems. Springer." (2009)

[18] Vijay Gehlot, Carmen Nigro , "AN INTRODUCTION TO SYSTEMS MODELING AND SIMULATION WITH COLORED PETRI NETS", Proceedings of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.(2010)