# Testing an Agent Based E-Novel System – Role Based Approach

N. Sivakumar and K. Vivekanandan

**Abstract.** Agent Oriented Software Engineering(AOSE) methodologies are meant for providing guidelines, notations, terminologies and techniques for developing agent based systems. Several AOSE methodologies were proposed and almost no methodology deals with testing issues, stating that the testing can be carried out using the existing object-oriented testing techniques. Though objects and agents have some similarities, they both differ widely. Role is an important mental attribute/state of an agent. The main objective of the paper is to propose a role based testing technique that suits specifically for an agent based system. To demonstrate the proposed testing technique, an agent based E-novel system has been developed using Multi agent System Engineering (MaSE) methodology. The developed system is tested using the proposed role based approach and found that the results are encouraging.

**Keywords:** Agent-Oriented Software Engineering, Multi-Agent System, Role based testing.

## 1 Introduction

A software development methodology refers to the framework that is used to structure, plan, and control the process of developing a software system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. Now-a-days agent based systems are the solutions for complex application such as industrial, commercial, networking, medical and educational domain [1]. The key abstraction in these solutions is the agent. An “agent” is an autonomous, flexible and social system that interacts with its environment in order to satisfy its design agenda. In some cases, two or more agents should interact with each other in a Multi Agent System (MAS) to solve a

N. Sivakumar · K. Vivekanandan
Department of Computer Science and Engineering,
Pondicherry Engineering College, Puducherry, India
e-mail: {sivakumar11,k.vivekanandan}@pec.edu

A. Abraham and S.M. Thampi (Eds.): Intelligent Informatics, AISC 182, pp. 165–173.
springerlink.com
© Springer-Verlag Berlin Heidelberg 2013

problem that they cannot handle alone. The agent oriented methodologies provide us a platform for making system abstract, generalize, dynamic and autonomous. This important factor calls for an investigation of suitable AOSE frameworks and testing techniques, to provide high-quality software development process and products.

Roles provide a well-defined interface between agents and cooperative processes [2]. This allows an agent to read and follow, normative rules established by the cooperation process even if not previously known by the agent. Their major motivation to introduce such roles is to increase the agent system's adaptability to structural changes. Several AOSE methodologies were analysed and compared and found that the strong weakness observed from almost all the methodologies were, there is no proper testing mechanism for testing the agent-oriented software. Our survey states that the agent based software are currently been tested by using Object-Oriented (OO) testing techniques, upon mapping of Agent-Oriented (AO) abstractions into OO constructs [3]. However agent properties such as Autonomy, Proactivity, and Reactivity etc., cannot be mapped into OO constructs. There arises the need for specialized testing techniques for agent based software. The main objective of the paper is to propose a testing mechanism based on agent's important mental state, the role.

The Paper is organized as follows: Section 2 describes the literature study on the existing work on agent oriented methodologies and existing testing techniques. Section 3 explains the analysis, design and implementation process of an agent based E-Novel system using MaSE methodology. Section 4 explains the proposed role based testing mechanism and its effectiveness towards agent based system.

## 2 Background and Related Works

Agent-oriented software engineering is a new discipline that encompasses necessary methods, techniques and tools for developing agent-based systems. Several AOSE methodologies [4] were proposed for developing software, equipped with distinct concepts and modelling tools, in which the key abstraction used in its concepts is that of an agent. Some of the popular AOSE methodologies were MASCommonKADS, MaSE, GAIA, MESSAGE, TROPOS, PROMETHEUS, ADLEFE, INGENIAS, PASSI, AOR Modeling. Very few methodologies provide validation support but fail to contribute complete testing phase. The TROPOS methodology has an agent testing framework, called eCAT. eCAT is a tool that supports deriving test cases semi-automatically. Goal oriented testing [5] contributes TROPOS methodology by providing a testing process model, which complements and strengthens the mutual relationship between goals and test cases. PROMETHEUS methodology provides only debugging support. PASSI methodology contributes only unit testing framework. INGENIAS provides basic interaction debugging support through INGENIAS Agent Framework (IAF). Table 1 clearly indicates that the existing AOSE methodologies does not support testing phase, stating that testing an agent system has been accommodated using existing traditional and object-oriented testing techniques.

**Table 1** Lifecycle coverage of several AOSE methodologies [4]

| Life Cycle Coverage | MAS Common KADS(1996-98) | MaSE(1999) | GAIA(2000) | MESSAGE(2001) | TROPOS(2002) | PROMETHEUS (2002) | ADLEFE(2002) | INGENIAS(2002) | PASSI(2002) | AOR Modelling (2003) |
|---|---|---|---|---|---|---|---|---|---|---|
| Analysis | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Design | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Coding | No | No | No | No | Yes | Yes | Yes | Yes | Yes | No |
| Testing | No | No | No | No | No | No | No | No | No | No |

Role is an important mental attribute of an agent and often agent changes its roles to achieve its designated goal. Roles are intuitively used to analyze agent systems, model social activities and construct coherent and robust teams of agents. Roles are a useful concept in assisting designers and developers with the need for interaction. Generic Architecture for Information Access (GAIA) methodology [6] and Multiagent Systems Engineering (MaSE) methodology [7] were role-based methodologies for development of multi-agent systems.

## 3 Proposed Work

The main objective of this paper is to propose a role based testing technique that suits specifically for an agent based system. Role based testing is applied at different abstraction level such as unit, integration, system and acceptance. To illustrate the role based testing approach, an agent based E-novel system was developed using MaSE methodology. E-novel system is deployed on internet community to assist in interaction between novelists and readers. In the e-novel community, system accepts and contains a number of novels authored by various novelists. Readers simply browse to find and read novels. However, readers typically spend lot of time to browse and review a list of novels through categories and ranking. An e-novel system which is a subsystem designed from the notion of this study for this community.

E-Novel system has been developed using MASE methodology which is an iterative process. It deals the capturing the goals and refining the roles of an agent. It appears to have significant tool support. *agentTool* is a graphically based, fully interactive software engineering tool, which fully supports each step of MaSE analysis and design. Fig.1 represents goal hierarchy diagram of e-novel system designed using *agentTool*. The analysis phase involves capturing goal, Applying use cases and Refining roles (Fig 2) whereas the design phase involves Creating agent classes, Constructing conversations, Assembling agent classes (Fig 3) and System design.

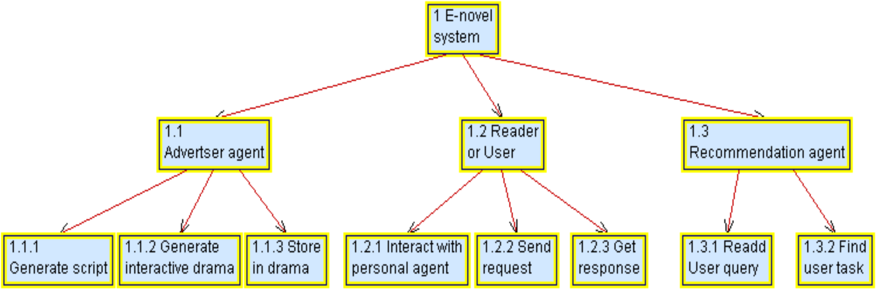


**Fig. 1** Goal Hierarchy Diagram for E-Novel System

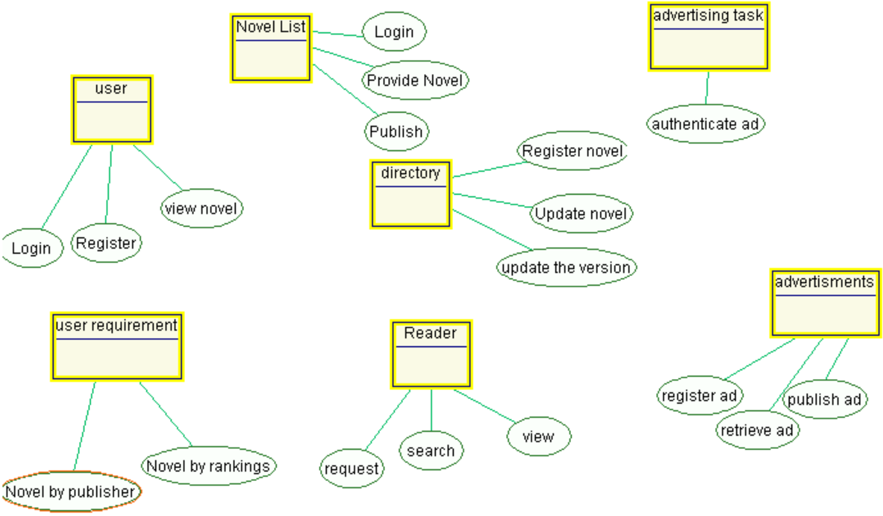


**Fig. 2** Role Diagram for E-Novel System

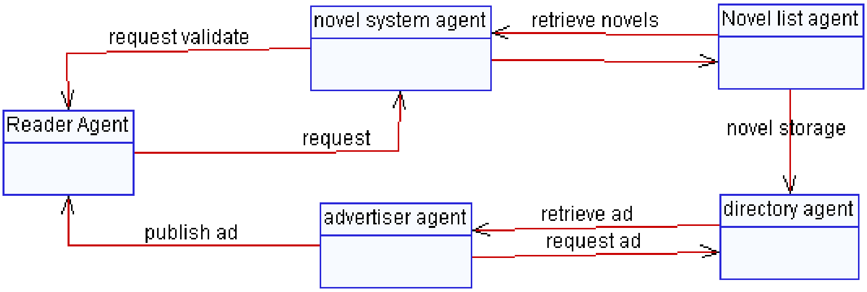


**Fig. 3** Agent Template Diagram for E-Novel System

After analysis and design, the Agent-based E-Novel system is implemented using JADE (Java Agent Development Framework), a software platform that provides basic middleware layer functionalities which are independent of the specific application and which simplify the realization of distributed applications that exploit the software agent abstraction [10].

# 4 Testing

Testing is inseparable in software development process. Though testing is mandatory, there is a hindrance for its uptake due to the fact that the existing AOSE methodologies failed to prioritise agent-oriented testing, stating that agent systems can be tested using the existing conventional and object-oriented software testing technique. Currently testing is accomplished by mapping Agent-Oriented (AO) abstractions into OO constructs. However agent properties such as Autonomy, Proactivity, and Reactivity etc., cannot be mapped into OO constructs. This leads to the need for specialized agent-oriented software testing technique for agent-oriented software systems. In this paper, role based testing technique is proposed for effectively testing an agent based system.

## *4.1 Role Based Testing*

Roles have been used both as an intuitive concept in order to analyse MAS and model inter-agent social activity and as a formal structure in order to implement coherent and robust agent-based software. Every individual agent has its own goal to be achieved and plans to do to fulfill the goal. In addition to goal and plan, role is one important mental state of the agent, which is defined as a set of capabilities and expected behavior. A role [8][9] can be represented as

<Goal, Responsibilities, Protocol, Permissions>

- Goal, for which the agent playing this role is responsible
- Responsibilities, Which indicates the functionalities of agents playing such roles
- Protocol, which indicates how an agent playing such role can interact with agents playing other role
- Permissions, which are a set of rights associated with the role.

Role based testing provides the full range of assurance and correctness for agents to manage the complexity of highly dynamic and unpredictable environments with a high degree of interaction and distributivity. Every Agent involved in the E-Novel system has their own roles for their accomplishment. Some roles involves only one agent and other involves more than one agent thereby interaction among agent is facilitated. Fig 5 represents the role diagram which shows the Goal-Role-Responsibility relationship.

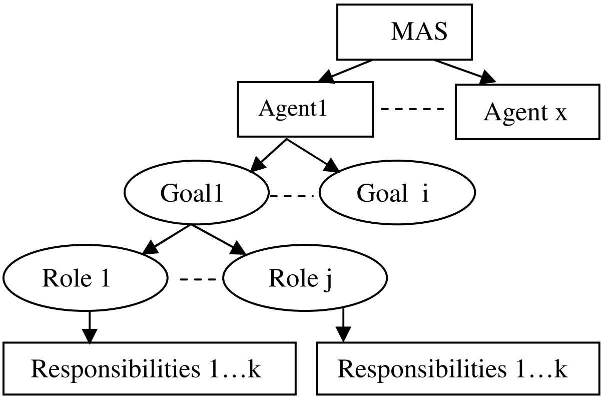


**Fig. 5** Role Model

Let ‘x’ be the number of agents involved in Agent-based E-novel system,

$$\text{Agents} = \{A_1, A_2, A_3, \ldots ,A_x\}$$

Let ‘i’ be the number of goals that every agent in the agent based e-novel system has to achieve.

$$\text{Goals} = \{G_1, G_2, G_3, \ldots. G_i\}$$

Let ‘j’ be the number of roles carried out by individual agent to accomplish every goal.

$$\text{Roles} = \{R_1, R_2, R_3, \ldots., R_j\}$$

Let ‘k’ be the number of functionalities to be accomplished for every role

$$\text{Responsibilities} = \{Re_1, Re_2, Re_3, \ldots.Re_k\}$$

After identifying the roles and their corresponding responsibilities of every agents involved in the MAS, test cases has to be derived to test whether the roles for the agents accomplish their task for the given set of inputs. Analyzing the Goal-Role relationship, it is found that, as long as the agent performs its role properly, the goal of the system is been achieved by default. Thus testing whether the agent performs its role properly is a challenging task. This paved way for a role-oriented testing mechanism by which the role functionalities were tested by deriving appropriate test cases. Random based test case generation technique is been used for generating test cases that suits for role based approach.

## *4.2 Role Schema*

Roles schema provide a well-defined interface between agents and cooperative processes. This allows an agent to read and follow, normative rules established by the cooperation process even if not previously known by the agent. Their major motivation to introduce such roles is to increase the agent system's adaptability to structural changes. They formally define a role as an entity consisting of a set of required permissions, a set of granted permissions, a directed graph of service invocations, and a state visible to the runtime environment but not to other agents. Sample role schema for an agent based e-novel system is represented in Table 2.

**Table 2** Sample Role Schema

| |
|---|
| **Role Name :** Prioritize Novel |
| **Agent involved :** Recommendation Agent |
| **Goal:** To present the best novel to the reader |
| **Description:** This role helps to prioritize the novels based on the author popularity and reader's interest. |
| **Protocol and Activities:** Analyzing No. of novels written by the author, Popularity among the readers, Novel writing skill and presentation skill, |
| **Permissions:** Read Request query, Result, Security policy**,** Change Result format // encrypt, Request format // decrypt |
| **Responsibilities:** *Activeness:* (count no. of novels + popularity of the novelist + writing skill + presentation skill + no. of readers read that novel<br>*Completeness:* Suggesting best novel to the reader |

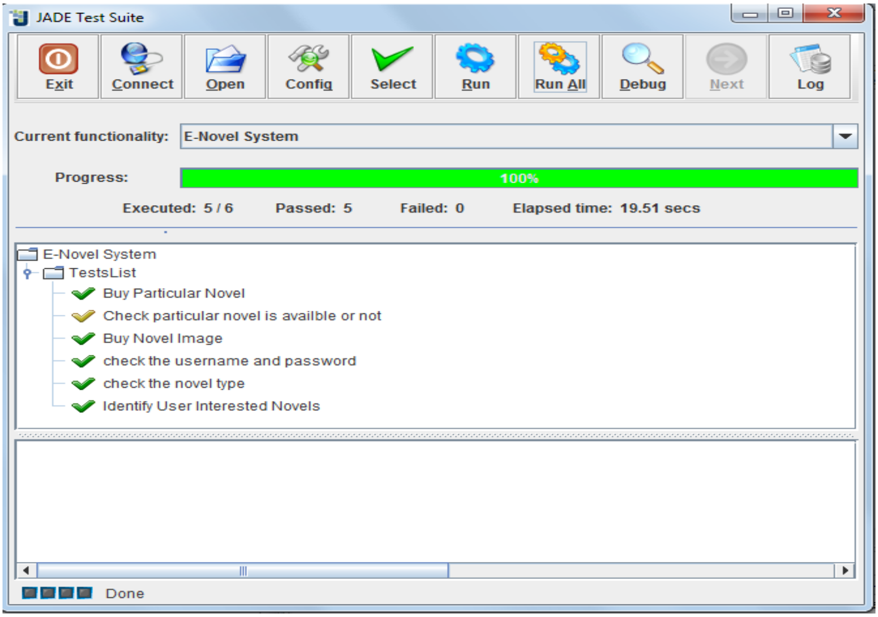
## *4.3 Random Based Test Case Derivation*

According to our approach, the role of an agent comprises the logic of the test. As every role of an agent has number of responsibilities to get satisfied, the derivation of test case focuses on the responsibilities and thereby validates whether the role hold by the agent serves the purpose. Random based test cases generation technique is applied for generating test suites. This technique generates test cases selecting the communication protocol and randomly generated messages. A sample test case is represented in Table. 3.

**Table 3.** Sample Test case for E-Novel system based on role

| T.ID | TESTED AGENT | GOAL | ROLE | SITUATION | INPUT | EXPECTED RESULT | OBSERVED RESULT | RESULT |
|---|---|---|---|---|---|---|---|---|
| T1 | AD Agent | To Publish Ad | Ad preference | Need for Novel | Novel name | Search for novel and publish relevant Advertisement | Novel list with appropriate Advertisement | Passed |

Once every roles were identified and test cases were generated, we tested our system in JADE Test Suite and we found that correctness of the system is validated fully and thereby role based testing can be applied for testing agent-oriented system



Current functionality: E-Novel System

Progress: 100%

Executed: 5 / 6 Passed: 5 Failed: 0 Elapsed time: 19.51 secs

- E-Novel System
  - TestsList
    - ✓ Buy Particular Novel
    - ✓ Check particular novel is availble or not
    - ✓ Buy Novel Image
    - ✓ check the username and password
    - ✓ check the novel type
    - ✓ Identify User Interested Novels

Done

**Fig. 6** Snapshot of JADE Test Suite

## 5 Conclusion

Testing being very important activity in Software Development Life Cycle (SDLC), there is no well defined testing technique for agent based system. None of the existing AOSE methodologies deals with testing phase stating testing can be carried out by using existing object-oriented technique. Although there is a well defined OO testing technique to test the agent based systems AO constructs cannot be mapped completely into OO constructs. Thus there arises vacancy for testing phase in the SDLC that should be filled-up. This paper deals with a new agent based testing technique i.e role based testing designed specifically for agent-oriented software so as to fit in the existing AOSE methodologies. To demonstrate our proposed testing technique, we developed an agent based E-novel system using MASE methodology and tested the system using role based approach and found that our results are encouraging. Thus the proposed testing technique performs adequately and accurately for testing the completeness of agent based system.

## References

1. Shoham, Y.: Agent oriented programming (Technical Report STAN-CS-90-1335) Stanford University: Computer Science Department (1994)
2. Cabri, G., Leonardi, L., Ferrari, L., Zambonelli, F.: Role-based Software agent interaction models: a survey. The Knowledge Engineering Review 25(4), 397–419 (2010)
3. Srivastava, P.R., Anand, K.V., Rastogi, M., Yadav, V., Raghurama, G.: Extension of Object-oriented Software testing techniques to Agent Oriented software testing. Journal of Object Technology 7(8), 155–163 (2008)
4. Henderson-Sellers, B., Giorgini, P.: Agent-Oriented methodologies. Idea Group Inc. (2005)
5. Nguyen, D.C., Perini, A., Tonella, P.: A Goal-Oriented Software Testing Methodology. In: Luck, M., Padgham, L. (eds.) AOSE VIII. LNCS, vol. 4951, pp. 58–72. Springer, Heidelberg (2008)
6. Henderson, B., Giorgini, P.: The Gaia Methodology for Agent-Oriented Analysis and Design Autonomous Agent and Multi-Agent Systems, vol. 3, pp. 285–312. Kluwer Academic Publishers (2000)
7. Wood, M.F.: Multiagent system engineering: A methodology for analysis and design of muti-agent systems. Master thesis, School of Engineering, Air Force Institute of Technology, USA (2000)
8. Xu, H., Zhang, X., Patel, R.J.: Developing Role-Based Open Multi-Agent Software Systems. International Journal of Computational Theory and Practice 2(1) (June 2007)
9. Kumar, M.: Roles and Ontology for Agent Systems. Global Journal of Computer Science and Technology 11(23), Version 1.0 (December 2011)
10. Padhagam, L., WinikOff, M.: Developing Intelligent Agent Systems A practical guide. RMIT University, Melbourne