

Underpinning of Object Oriented Software Agent in portalware

Anand Kumar Pandey
Research Scholar, Mewar University
Chittorgarh, Rajasthan, India
anandpandey.ind@gmail.com

Dr. A.S. Saxena
Guide, Dept. of C.S.E.
anand.saxena42@gmail.com

Abstract— The multi agent system (MAS) in Object - Oriented techniques has represented an new and exciting means of analyzing, developing, and building advanced and useful software systems. The MAS are applied with different features of software system. In this paper we represented the application of object oriented software agent in Portalware using MAS methodology and local behavior as a role of multi agents system which gives an idea about appropriate system-level study of object oriented software engineering design and technology. MAS developers however have relied mostly, on object -oriented design techniques and programming languages in portalware, such as Java. This paper discusses software engineering approaches for MASs, and presents a new method for integrating agents into object-oriented software engineering as a portalware from an early stage of design.

Keywords- Multi -agent systems, software agents, software engineering, object-oriented systems, aspect-oriented software development.

I. INTRODUCTION

A variety of software development paradigms like procedural, object oriented, agent oriented etc, have been developed and designed in software engineering. Agent technology has been revisited as a complementary approach to the object paradigm and as a means of designing and implementing complex distributed software. Agents and objects are abstractions recognizably different from a software engineering viewpoint. The importance of agent properties in today's software systems is reflected by the support provided for such properties in object-oriented middleware, programming languages, and implementation of frameworks. Portalware is a web-based environment that supports the development and management of e-commerce portals [1]. This environment introduces a systematic approach for producing and maintaining portals by means of a clear separation of user roles. Portalware also supports communication facilities for coordinating and organizing the activities of the portal developers. Software agents have been introduced to Portalware in order to assist its users with time-consuming activities and automate repetitive user tasks. We selected this system as a case study, because the agency concerns handled in this project are ones typical of many existing applications. This MAS encompasses several agency concerns, including agent types, roles, collaboration, interaction, adaptation, and autonomy [3]. Most existing object oriented software architectures typically incorporate agent models which focus on one type of agent, and do not provide direct support for handling and reusing properties and capabilities separately. When developing object-oriented software architecture types of agent systems, software agents are being specialized for special fields of application according to their mobility, co-ordination, co-operation, security and interoperability.

In this paper, we briefly discuss the current research in software engineering of agent systems, and present an innovative aspect -based approach for designing and implementing agent based object -oriented systems for portalware.

II. MULTI AGENT SYSTEM (MAS)

Multi agent systems are an approach to build complex distributed applications. A multi agent system consists of a population of autonomous entities (agents) situated in a shared structured entity (the environment). One classic definition of an autonomous agent is: an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [4]. We put forward agents and the environment as first-order abstractions in multi agent systems. This allows to clearly define the environment responsibilities that differ from the agent responsibilities. We also demonstrate our

multi-agent approach through the Portalware system, a web-based environment for the development of e-commerce portals. Our proposal explores the benefits of aspect-based design and implementation for mastering the increasing complexity of integrating software agents into the object model. A multi-agent application generally has several types of software agents [9], such as information agents, user agents, and interface agents.

The extent to which agents are able to access a particular resource or service may depend on several factors such as the nature of the resource or service, the capabilities of the agent, the interrelationships with other resources, services or agents, etc. In this paper, we represent a web-based environment for the construction and management of e-commerce portals object oriented design of portalware.

III. OBJECT ORIENTED MODELING APPROACH

Traditionally, existing object-oriented proposals often focus on the implementation phase, and do not provide direct support for handling and reusing properties and capabilities separately. But the proposed Object-oriented languages, such as C++ or JAVA, provide significant advances over standard procedural languages with respect to the reusability and modularity of code:

- Encapsulation: encourages the creation of library interfaces that minimize dependencies on underlying algorithms or data structures. Changes to programming internals can be made at a later date with requiring modifications to the code that uses the library.
- Inheritance: permits the extension and modification of a library of routines and data without requiring source code to the original library.
- Polymorphism: allows one body of code to work on an arbitrary number of data types.

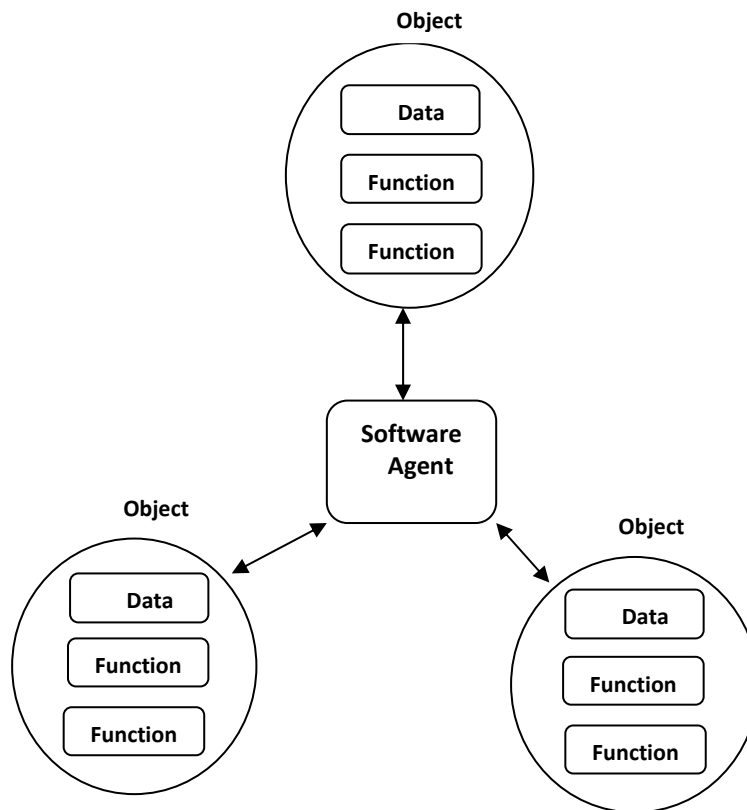


Figure 1: Object oriented Model for Software Agent

The object-oriented programming (OOP) languages can be used to build monolithic programs out of many object building blocks. Software agent systems are a very complex, highly-developed form of software architecture [2]. The pursued architecture-based approach is suitable to analyze, synthesize and to substantiate software agents and agent systems theoretically, to analyze systematically the properties, structure and classification of agents and agent systems. Object-oriented design and programming has been proposed as a technique for improving separation of concerns in software design and implementation. The central idea is that while hierarchical modularity mechanisms of object-oriented design and implementation languages are

extremely useful, they are inherently unable to modularize all concerns (properties) of interest in complex systems.

IV. OBJECT ORIENTED DESIGN OF PORTALWARE AGENT

Object-oriented software development has been proposed as a technique for improving separation of concerns in software construction and support improved reusability and maintainability. The object-oriented technique is not restricted to the object paradigm, but it is our focus in this experiment. The central idea is that while pure abstractions of the object paradigm are extremely useful, they inherently are unable to modularize all concerns of interest in complex systems [8].

Following figure 2 illustrates the software agents in Portalware of web based system using object oriented design. Portalware encompasses three agent types: (i) interface agents, (ii) information agents, and (iii) user agents. Each of them has different capabilities and properties, but everyone implements the fundamental aspects defined by agenthood. Figure 2 summarizes capabilities and agency properties for the Portalware agents. For purpose of brevity, we discuss in detail only the Portalware's information agents. These agents are responsible for mediating the conversation between all system agents, providing services such as naming service [6]. User agents represents portalware users and are implemented to reduce the need for cross talk between working users. Several roles are attributed to portalware users and their respective agents. Portalware users often need to search for information which is stored into two different databases.

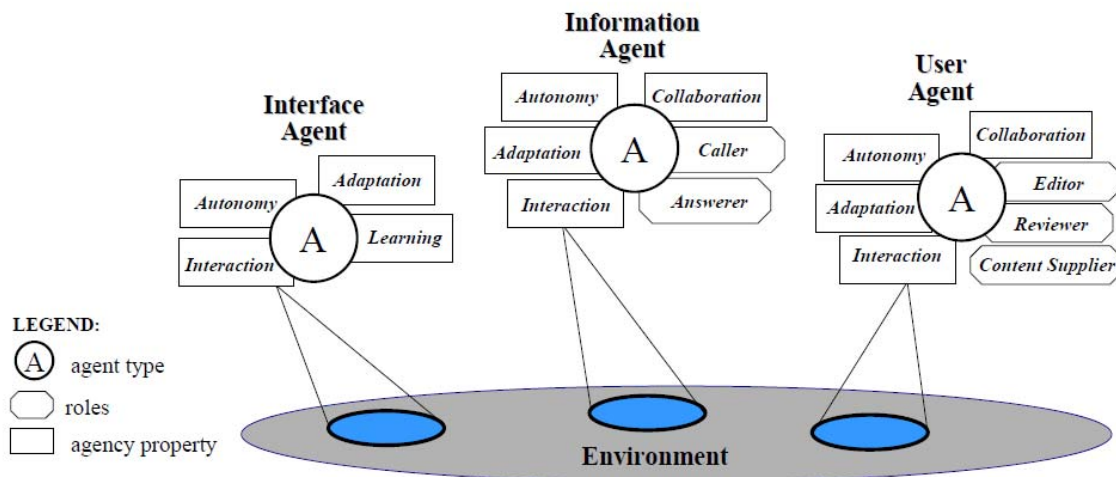


Figure 2: Portalware Agent

Each information agent is attached to a database, and contains plans for searching for information. The search plan determines the agent's searching capability. An alternative collaborative capability is used, when an information agent is not able to find the information in the attached database [7]. The agent uses its calling collaborative capabilities in order to call the other information agent and ask for this information. Similarly, the latter uses its answering capabilities so that it can receive the request and send the search result. Note that both of them need to include calling and answering capabilities.

V. REUSABILITY AND MAINTAINABILITY OF PORTAWARE AGENT

The study was divided into two major phases: the Construction phase, and the Reuse and Evolution phase with the concept of portalware. Reusability is the ability of software elements to serve for construction of other different elements in the same software system or across different ones. In this study, we are interested in evaluating the reusability of agency concerns in design and code. Maintenance is the activity of modifying a software system after initial delivery [5]. Software maintainability is the ease with which the software components can be modified. In this paper, we presents a framework for assessing reusability and maintainability of object – oriented software. The portalware agent model defines precisely how to measure reusability and maintainability based on a set of proposed metrics.

For each scenario, the difficulty of maintainability and reusability was defined in terms of structural changes to the artifacts in the agent oriented and object oriented systems. The total line of code, that were added, changed, or copied to perform the maintenance tasks as per scheduled design.

VI. CONCLUSION

However, the development of software agents is not a trivial task. This work discussed the problems in dealing with agency properties and capabilities as well as overviewed software engineering approaches to addressing these problems. So, we presented an object-based approach to make development of sophisticated agents simple enough to be practical. We developed a framework to evaluate the produced systems in terms of our defined goals, questions, and hypotheses. The set of object-oriented models, its design and implementation were based on the same requirement specifications and satisfying the same set of scenarios. We have presented a framework design, which is based on a suite of metrics and a quality model, to assist the assessment of object – oriented software in terms of reusability and maintainability.

REFERENCES

- [1] A. Graesser and S. Franklin, "Is it an agent, or just a Program?" A Taxonomy for Autonomous Agents". Third International conference On Agents Theories, Architecture, and Languages,1996.
- [2] Briand, L., Daly, J., Wüst, J. "A Unified Framework for Cohesion Measurement in Object Oriented Systems". Empirical Software Eng. Journal, 3 (1), 1998, pp 65 - 117.
- [3] D. Brugali and K. Sycara. "A Model for Reusable Agent Systems". In: Implementing Application Frameworks – Object-Oriented Frameworks at Work, M. Fayad et al. (editors), John Wiley & Sons, 1999.
- [4] Jeffrey M. Bradshaw, An Introduction to Software Agents, page 10.
- [5] Leonard Foner, "What's an Agent Anyway?", Agents Memo 93-01, Agents Group MIT Media Lab, 1993.
- [6] Michael Wooldridge and Nick Jennings, Dept. of Computer science, \Manchester Metropolitan University, IEEE Review, January 1996, pp 17-20.
- [7] Nwana, H.S. 1996. Software Agents: An Overview. In Knowledge Engineering Review, Vol.1 1, No.3, 205-244, Cambridge University Press.
- [8] Qusay H. Mahmoud, Software Agents: Characteristics and Classification, School of Computing Science, Simon Fraser University Burnaby, BC. Canada, In Proceeding of the fifth International conference 2004.
- [9] Syed Mahbubur Rahman, Robert J Bignall, Internet Commerce and Software Agents: Case Technologies and opportunities, Vol-I, 75-88, Idea Group Publication.