INTEGRATING AGENT-ORIENTED ENTERPRISE SOFTWARE ENGINEERING INTO SOFTWARE ENGINEERING CURRICULUM

Gilda Pour[1]

Abstract *– Agent-oriented enterprise software engineering (AOESE) has emerged as a promising approach to developing Web-based enterprise software systems. AOESE is based on developing and evolving enterprise software systems from selected pre-engineered and pre-tested software agents, and then assembling them within appropriate software architectures. We view agent-oriented software engineering as an extension of conventional component-based software engineering, and software agents as next-generation software components. Software agents offer greater flexibility and adaptability than traditional components. However, making transition to agent-oriented enterprise software engineering requires software engineers to learn a new set of technical skills. To provide such a learning opportunity, the author has created a new course sequence that integrates both agent-oriented enterprise software engineering and component-based enterprise software engineering into software engineering curriculum. A new course in the sequence is intended to provide students with the opportunity to acquire a good understanding of the key concepts and principles of agent-oriented enterprise software engineering, and the new opportunities and challenges involved in developing Web-based multi-agent systems. The course is designed to help build a solid foundation for integration of research into education in the area of agent-oriented enterprise software engineering. The course has special emphasis on developing architectures and frameworks for Web-based multi-agent enterprise systems rather than developing single agent. In this paper, the author shares her experience of developing the course, and presents the course organization, its components, and the future plans for the course.*

Index Terms - Agent-oriented enterprise software engineering, component-based enterprise software engineering, software agents, software engineering curriculum, workflows.

AGENT-ORIENTED ENTERPRISE SOFTWARE ENGINEERING

Rapidly rising demand for more flexible, adaptable, extensible, and robust Web-based enterprise application systems has motivated the search for new software engineering methodologies and development strategies.

The new strategies must be based on developing enterprise software systems by assembling flexible and adaptable reusable software components built at different times by various developers. Traditional software development strategies and engineering methodologies, which require development of software systems from scratch, fall short in this regard.

Agent-oriented enterprise software engineering (AOESE) has emerged as a promising approach to building Web-based enterprise application systems. Agent-oriented enterprise software engineering has the potential to

- reduce significantly the cost and time-to-market of enterprise application systems by allowing the systems to be built through assembling a set of software agents rather than from scratch;
- enhance the reliability of enterprise application systems by allowing the systems to be developed through assembling a set of reusable software agents;
- improve the maintainability and flexibility of enterprise application systems by allowing replacement of old agents with quality agents in the system; and
- enhance the quality of enterprise application systems by allowing application-domain experts to develop software agents and software engineers to build the systems by assembling the agents.

Agent-oriented enterprise software engineering is based on developing and evolving software systems from selected pre-engineered and pre-tested software agents, and then assembling them within appropriate software architectures.

Agent-oriented software engineering allows developers to use a set of high-level, flexible abstractions to represent enterprise software systems. Rapid integration of distributed agents provides opportunities to build Web-based multi-agent application system for a wide variety of application domains; for instance, information (e.g. health care, education, missing children, automobiles, travel, real estate, employment), banks (e.g. virtual banks), finance, E-business, media, government (e.g. space exploration), personal assistants (PAs), wearable computers, smart clothes, virtual aquarium, virtual pets, and manufacturing.

[1] Gilda Pour, Department of Computer Engineering, San Jose State University, San Jose, CA 95192-0180, U.S.A. gpour@sjsu.edu

SOFTWARE AGENTS

Agent-oriented enterprise software engineering has emerged as an extension of conventional component-based software engineering (CBSE), and software agents as next-generation software components, and [1]-[3].

A software agent is a specialized software component that interacts as a surrogate for its user with its environment and other agents, and reacts to changes in the environment.

There is a general agreement that an agent is a reusable component that exhibits a combination of several of the following characteristics:

- *Autonomous.* The agent proactively initiates activities according to its goals, acts on its user's behalf, and exercises control over its own actions.
- *Adaptable.* The agent changes its behavior after deployment as a result of its learning and acquired information, user customization, and/or downloading new capabilities.
- *Mobile.* The agent moves from one executing context to another, continues execution in a new context, and retains its state to continue its work.
- *Knowledgeable.* The agent reasons about its goals, acquired information, and knowledge about other agents and users.
- *Collaborative.* The agent communicates and cooperates with other agents to form dynamic or static societies of agents, and performs a task in collaboration with some other agents.
- *Persistence.* The infrastructure enables agents to retain knowledge and state over extended periods, including robustness at possible runtime failures [1].

Software agents offer greater flexibility and adaptability than traditional components. There are three categories of agents based on the major function of agents:

- *Personal agents* interact directly with a user, present some personality or character, monitor and adapt to the user's activities, learn the user's style and preferences, and automate or simplify certain tasks.
- *Mobile agents* visit remote sites to collect information before returning with the results, and aggregate and analyze data or perform local control.
- *Collaborative agents* communicate and interact with some other agents as they represent their users, organizations, and services. Multiple collaborative agents exchange messages to negotiate or share information [1].

Intelligent agents are autonomous agents that have their own goals and beliefs and can reason about their present and future behavior—offer many opportunities for rapid, incremental development of Web-based enterprise application systems. Developers can apply these systems to a variety of complex, dynamic domains, ranging from e-commerce to human planetary exploration [1][2][5].

MAKING TRANSITION TO AGENT-ORIENTED ENTERPRISE SOFTWARE ENGINEERING

While agent-based systems are becoming increasingly understood, multi-agent systems development is not [6]. This is due to several major differences between agent-oriented software development lifecycle and the traditional software development lifecycle. For instance,

- *Design phase* in agent-oriented software development lifecycle includes selection and customization of software architectures, as well as selection and customization of a set of software agents. It is critical to design and architect the extensibility and scalability into a multi-agent enterprise system and all its parts.
- *Implementation phase* in agent-oriented software development lifecycle is not about extensive coding to build agent-oriented enterprise systems from scratch. It rather involves developing enterprise software systems that support interaction and cooperation among a set of agents within appropriate software architectures.
- *Testing phase* in agent-oriented software development lifecycle involves testing an enterprise system that is an assembly of a set of software agents built by various developers. The system integration testing has to address a series of new issues raised by the lack of confidence in and understanding of agents built by others [3][5].

Due to several major differences between the agent-oriented enterprise software development lifecycle and the traditional software development lifecycle, software engineers need to acquire a new set of software engineering skills for developing, maintaining, and evolving multi-agent enterprise software systems.

The author has developed a new course that integrates agent-oriented enterprise software engineering into the software engineering curriculum. The course is designed to provide students with the opportunity to learn the key concepts and principles of agent-oriented enterprise software engineering, software agents, and workflows for Web-based enterprise application systems, agent technologies, and multi-agent systems.

The availability and growing popularity of Java, XML, XML schema, and XML-based technologies helps make transition from traditional software engineering to agent-oriented enterprise software engineering.

ENTERPRISE SOFTWARE ARCHITECTURE

Software architecture refers to systems and components that must be structured for the support of independent development, and integration and evolution of components and systems [4]. Agents are specialized components.

Well-designed software architecture with flexibility to adapt and extend is required for development of agent-oriented enterprise software systems. Multi-agent enterprise software systems must be architected, designed, packaged, and supported for effective adoption of AOESE.

Agents reside and execute in a conceptual and physical location called an agency. The agency provides facilities for collecting knowledge about other agents and for locating and messaging agents.

INTEGRATING AGENT-ORIENTED ENTERPRISE SOFTWARE ENGINEERING INTO THE SOFTWARE ENGINEERING CURRICULUM

To prepare software engineering students for a challenging software engineering career, the author has created a course sequence that integrates both agent-oriented enterprise software engineering and component-based and reuse-driven software engineering into the software engineering curriculum [7][8].

The course sequence includes a new major course that is focused on agent-oriented enterprise software engineering, and particularly, on the integration of research into education in this area.

The new course has special emphasis on developing architectures and frameworks for Web-based multi-agent enterprise systems rather than developing single agent.

Because agent-oriented enterprise software engineering deals with different types of agents for developing enterprise software systems, the course is not limited to intelligent agents as an extension of the work done in Artificial Intelligence (AI) research and development.

The author has developed a new laboratory at San Jose State University with the external funding. The students use the lab and a variety of software tools different computing environments available in the lab for their course projects.

Course Description

The course on agent-oriented enterprise software engineering is developed for students who are interested in pursuing software engineering profession. It is a graduate course. Senior level software engineering students who have met the prerequisites for the course are permitted to take the course as one of their technical electives.

The course is designed to provide students with the opportunity to

- learn key concepts and principles of agent-oriented enterprise software engineering, software agents and workflows for Web-based enterprise application systems, agent technologies, multi-agent systems; and
- enhance their professional software engineering skills, particularly, teamwork and effective communication skills (both written and verbal).

The course emphasizes both the theoretical and practical aspects of the topics covered in the course. It covers several related case studies and industrial examples. The course also includes a series of assignments and a team-based term project.

Course Prerequisites

Students are expected to have a good understanding of the key concepts and principles of object-oriented and component-based software engineering, and some experience working with Java and eXtensible Markup Language (XML) prior to taking the course. At San Jose State University, we offer courses on the subjects that constitute the prerequisites of this course.

Course Organization

The course is divided into four major parts as shown in Figure 1.

Part I	Introduction to Agent-Oriented Enterprise Software Engineering
Part II	Multi-Agent Systems
Part III	Software Agents and Workflows
Part IV	Application Areas

Figure 1. Course Organization

Part I – In the first part of the course, we introduce the key concepts and principles of agent-oriented enterprise software engineering in the first part of the course. We also provide a common terminology required for understanding of the

subject. We discuss the motivation for agent-oriented development of Web-based enterprise application systems, and new opportunities and challenges involved in making transition from traditional software engineering to agent-oriented software engineering. We also introduce agent-oriented software development lifecycle, and various types of software agents with a different mixes of characteristics such as intelligence, mobility, autonomy, collaboration, persistence, and adaptability.

Part II – In the second part of the course, we present the key concepts and principles of multi-agent systems. The following topics are covered in this part of the course: agent-based system architecture, agent interactions and cooperation, multi-agent organizations, agent actions and behaviors, agent communications, agent communication languages, Knowledge Query and Manipulation Language (KQML), the Foundation for Intelligent Physical Agents (FIPA), and Web-based multi-agent system modeling. We also discuss the use of Web-based technologies such as Java, XML, XML schema, XML-based technologies, and HTTP for development of multi-agent systems.

Part III – In the third part of the course, we discuss the combination of software agents and workflows as it provides major benefits beyond those associated with components and scripting.

Part IV – In the fourth part of the course, we discuss development of a wide variety of agent-based systems and their applications for

- network operations;
- e-business;
- information (e.g. health care, education, missing children, automobiles, travel, real estate, partners, employment);
- finance;
- banks (e.g. virtual banks);
- retail stores;
- media;
- government (e.g. space exploration);
- information technology;
- manufacturing;
- personal assistants (PAs);
- wearable computers;
- smart clothes;
- virtual aquarium; and
- virtual pets.

Agent technology will continue to be integrated into the Internet with increasing intelligence and complexity. We also discuss new opportunities (e.g. a combination of agents and avatars, virtual cooperation), new challenges (e.g. privacy and legal issues), and the role of standards. Future research directions include developing next-generation agents with new capabilities such as

- activating and inhabiting real world robotics and pursuing goals beyond the virtual world; particularly, nanotechnology-based mobile entities (miniature places and submarines, drugs, computers).
- self-replicating and developing other agents to meet specific needs. These manager agents will be independent and self-motivating, and in many respects demonstrate human capabilities.

Course Resources

The textbooks for the course are "Intelligent Software Agents" [9] and "Multi-Agent Systems" [10]. The books contain an extensive list of other useful resources on related topics. The students are also provided with additional notes that the author has provided for the course.

In addition, we provide our students with a list of related publications for our class discussion and further study of the subject.

Course Assignments

To help balance the understanding of the key concepts and principles of agent-oriented enterprise software engineering, the application of the agent-oriented enterprise software engineering theories and the best software engineering practices, the course includes a series of individual and team-based assignments.

Furthermore, the course includes projects that are designed to provide students with opportunity to develop a demonstration prototype of a Web-based multi-agent system. The class is divided into project teams of three or four students. Each team is assigned a different project topic. The examples of project topics include developing Web-based multi-agent systems for medical record management, air navigation, supply chain management, and collaborative review of software development.

Cooperative Learning Styles

We have adopted cooperative and hands-on, inquiry-based, and active learning styles in this new course on agent-oriented enterprise software engineering. We plan to

continue to enhance the course to take full advantage of the pedagogical possibilities.

Course Evaluation

We have evaluated the effectiveness of various aspects of the course. Students and industry representatives have participated in the course evaluation. They have ranked the course materials, course organization, student assignments and projects, and the overall effectiveness of the course *very good* to *excellent*.

CONCLUDING REMARKS AND

We have created and developed a new course that integrates the agent-oriented enterprise software engineering into the software engineering curriculum. The course is design to help build a solid foundation in agent-oriented enterprise software engineering for software engineering students. The emphasis is on developing enterprise-wide multi-agent systems rather than a single agent.

The course provides students with a great opportunity to acquire a good understanding of the key concepts and principles of agent-oriented enterprise software engineering, and the new challenges and opportunities involved in developing Web-based multi-agent systems.

The course is designed to emphasize both the theoretical and practical aspects of the course topics. To help students gain hands-on experience in this new software engineering area, the course includes a require term project that is focused on developing a demonstration prototype of a Web-based multi-agent system as term project.

The demand for this course has been high. There is also a great demand for an online version of the course. Those who will benefit most from an online version of the course are the working adults who are seeking the opportunities for learning new skills relevant to the software engineering profession for their professional growth and career advancement, but their schedules and commitments do not permit the conventional campus experience.

Developing an online version of this course to provide a learning opportunity for such working adults will be considered. We believe that development of an online version of the course has to ensure the human dynamics of the online course follow the practices and skills of the contemporary workplace. Students should work on individual as well as team-based projects; collaborate through a variety of electronic media, and complete projects. The online version of the course should emphasize the combination of software engineering, writing and interacting, as this will be a powerful combination for students who must not only master the technical skills, but also develop ability to communicate and collaborate professionally to use their technical skills in the actual workplace.

The online version of this course will not only offer life-long learners the access to this new valuable course, but it will also serve as a pedagogical template for other courses offering similar or related technical skills, using similar technologies, or reaching out to the same audience.

REFERENCES

[1] Griss, M. and Pour, G., "Accelerating Development with Agent Components," Cover Feature Article, *IEEE Computer*, Vol. 34, No. 5, May 2001, pp. 37-43.

[2] Pour, G., "Internet-Based Multi-Mobile-Agent Framework for Planetary Exploration," *Proc. International Conference on Intelligent Agents, Web Technology and Internet Commerce*, July 2001, pp. 153-163.

[3] Pour, G., "Web-Based Architecture for Component-Based Application Generators," *Proc. International Conference on Internet Computing*, June 2002.

[4] Shaw, M. and Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.

[5] Pour, G., "Jini-Based Mobile Agent Architecture for Human Planetary Exploration," *Proc. International Conference on Technology of Object-Oriented Systems and Languages (TOOLS USA)*, IEEE Computer Society Press, August 2001, pp. 337-346.

[6] Woodridge M. and Jennings, N., "Software Engineering with Agents: Pitfalls and Pratfalls," *IEEE Internet Computing*, Vol. 3, No. 3, May/June 1999, pp. 20-27.

[7] Pour, G., "Component Technologies: Expanding the Possibilities for Component-Based Development of Web-Based Enterprise Applications," *Handbook of Internet Computing*, CRC Press, 2000, pp. 133-156.

[8] Pour, G., "Integrating Component-Based and Reuse-Driven Software Engineering into the Software and Information Engineering Curriculum," *Proc. Frontiers in Education (FIE)*, IEEE Computer Society Press, 2000, Section TC2, pp. 18-23

[9] Murch, R. and Johnson, T., *Intelligent Software Agents*, Prentice Hall, 1999.

[10] Ferber, J., *Multi-Agent Systems*, Addison-Wesley, 1999.