Chapter 37
A Semantic Approach for Multi-Agent System Design

Rosario Girardi
Federal University of Maranhão, Brazil

Adriana Leite
Federal University of Maranhão, Brazil

ABSTRACT

Automating software engineering tasks is crucial to achieve better productivity of software development and quality of software products. Knowledge engineering approaches this challenge by supporting the representation and reuse of knowledge of how and when to perform a development task. Therefore, knowledge tools for software engineering can turn more effective the software development process by automating and controlling consistency of modeling tasks and code generation. This chapter introduces the description of the domain and application design phases of MADAE-Pro, an ontology-driven process for agent-oriented development, along with how reuse is performed between these sub-processes. Two case studies have been conducted to evaluate MADAE-Pro from which some examples of the domain and application design phases have been extracted and presented in this chapter. The first case study assesses the Multi-Agent Domain Design sub-process of MADAE-Pro through the design of a multi-agent system family of recommender systems supporting alternative (collaborative, content-based, and hybrid) filtering techniques. The second one evaluates the Multi-Agent Application Design sub-process of MADAE-Pro through the design of InfoTrib, a Tax Law recommender system that provides recommendations based on new tax law information items using a content-based filtering technique.

1. INTRODUCTION

Knowledge systems have evolved from expert systems to agent-oriented or multi-agent systems. A software agent is an entity that perceives its environment through sensors and acts upon that environment through actuators (Russel, & Norvig, 2009). Having the properties of autonomy, sociability and learning ability, software agents are a very useful software abstraction to the understanding, engineering and use of both complex software problems and solutions like distributed and open systems and to support

the decision making process (Leite, Girardi, Cavalcante, 2008b) (Newton, & Girardi, 2007). Agent attributes allow approaching the complexity of software development through appropriate mechanisms for software decomposition, abstraction and flexible interactions between components (Wooldridge, Jennings, & Kinny, 2000).

On the other hand, automating software engineering tasks is crucial to achieve better productivity of software development and quality of software products. Knowledge engineering approaches this challenge by supporting the representation and reuse of knowledge of how and when to perform a development task. Therefore, knowledge tools for software engineering can turn more effective the software development process by automating and controlling consistency of modeling tasks and code generation.

In the last years, many efforts have been devoted to the research on agent-oriented software engineering. The proposals have evolved from simple techniques for modeling specific applications to methodologies and software processes for supporting reuse in agent-oriented development. Earlier phases of agent-oriented development have been extensively explored in the literature. There is a different situation on medium development phases like architectural and detailed multi-agent design where several research topics are still open, between them the lack of systematic approaches for modeling architectures of software agents and multi-agent systems. This article contributes with concrete solutions to this challenge by formalizing a knowledge-based approach to the architectural and detailed design of families of multi-agent systems. A family of software systems (Pohl, Bockle, Linden, 2005) is defined as a set of systems sharing some commonalities but also having particular features.

Knowledge representation formalisms, like ontologies (Gruber, 1995), are used by modern knowledge systems, to represent and share the knowledge of an application domain. Supporting semantic processing, they allow for more precise information interpretation. Thus, knowledge systems can provide greater usability and effectiveness than traditional information systems. This is particularly the case of knowledge systems for software engineering.

MADAE-Pro ("Multi-agent Domain and Application Engineering Process") (Girardi, & Leite, 2011) (Leite, Girardi, 2009) is a process for the development and reuse of families of multi-agent software systems, which consists of two complementary sub-processes: Multi-agent Domain Engineering and Multi-agent Application Engineering. Multi-agent Domain Engineering is a process for the development of a family of multi-agent software systems in a problem domain, by applying MADEM ("Multi-agent Domain Engineering Methodology") (Girardi, & Leite, 2008); and Multi-agent Application Engineering, the one for constructing a specific agent-oriented application by reusing one or more of those families (Drumond, & Girardi, 2008), using MAAEM ("Multi-agent Application Engineering Methodology") (Drumond, & Girardi, 2008) (Leite, Girardi, & Cavalcante, 2008b). The process consolidates a long term research effort on techniques, methodologies and tools for promoting reuse on agent-oriented software development (Girardi, & Leite, 2008) (Leite, Girardi, & Cavalcante, 2008a) (Leite, Girardi, Cavalcante, 2008b).

Besides providing support for reuse in multi-agent software development, through the integration of concepts of Domain Engineering and Application Engineering, MADAE-Pro is an ontology-driven process where models of requirements, agents and frameworks are represented as instances of the ONTORMAS ("ONTOlogy driven tool for the Reuse of Multi-Agent Systems") ontology (Leite, Girardi, Cavalcante, 2008a). Thus, concepts are semantically related allowing effective searches and inferences, facilitating the understanding and reuse of software models during the development of specific applications in a domain. Also, the models can be easily documented, adapted and integrated.

Related Content

High-Speed Logic Level Fault Simulation
Raimund Ubarand Sergei Devadze (2011). *Design and Test Technology for Dependable Systems-on-Chip (pp. 310-337).*
www.irma-international.org/chapter/high-speed-logic-level-fault/51407

People: Communicating in Teams
(2019). *Software Engineering for Enterprise System Agility: Emerging Research and Opportunities (pp. 169-179).*
www.irma-international.org/chapter/people/207087

Kansei's Physiological Measurement and Its Application (2): Estimation of Human States Using PCA and HMM
Santoso Handriand Shusaku Nomura (2011). *Kansei Engineering and Soft Computing: Theory and Practice (pp. 319-329).*
www.irma-international.org/chapter/kansei-physiological-measurement-its-application/46406

Characterizations of Fuzzy Sublattices Based on Fuzzy Point
Chiranjibe Janaand Faruk Karaaslan (2020). *Handbook of Research on Emerging Applications of Fuzzy Algebraic Structures (pp. 105-127).*
www.irma-international.org/chapter/characterizations-of-fuzzy-sublattices-based-on-fuzzy-point/247650

An Intimate Relation: Human Beings with Humanoids
Elisabeth Damour (2011). *Kansei Engineering and Soft Computing: Theory and Practice (pp. 169-179).*
www.irma-international.org/chapter/intimate-relation-human-beings-humanoids/46397