Chapter 2.8
From Requirements to Code with the PASSI Methodology

Massimo Cossentino
ICAR-CNR, Italy

ABSTRACT

A Process for Agent Societies Specification and Implementation (PASSI) is a step-by-step requirement-to-code methodology for designing and developing multi-agent societies, integrating design models and concepts from both object-oriented (OO) software engineering and artificial intelligence approaches using the UML notation. The models and phases of PASSI encompass representation of system requirements, social viewpoint, solution architecture, code production and reuse, and deployment configuration supporting mobility of agents. The methodology is illustrated by the well-known Bookstore case study.

INTRODUCTION

At present, several methods and representations for agent-based systems have been proposed (Aridor & Lange, 1998; Bernon, Camps, Gleizes, & Picard, 2004; Bresciani, Giorgini, Giunchi-glia, Mylopoulos, & Perini, 2004; DeLoach & Wood, 2001; Jennings, 2000; Kendall, Krishna, Pathak, & Suresh, 1998; Zambonelli, Jennings, & Wooldridge, 2001, 2003). In order to explore them, we shall consider a relevant aspect in modelling software, that is, fidelity. Robbins, Medvidovic, Redmiles, and Rosenblum (1998) have defined fidelity as the distance between a model and its implementation. This means that low fidelity models are problem-oriented, while high fidelity models are more solution-oriented.

Since agents are still a forefront issue, some researchers have proposed methods involving abstractions of social phenomena and knowledge (Bernon et al., 2004; Bresciani et al., 2004; Jennings, 2000; Zambonelli, Jennings, & Wooldridge, 2001, 2003) (low-fidelity models); others have proposed representations involving implementation matters (Aridor & Lange, 1998; DeLoach & Wood, & Sparkman, 2001; Kendall et al., 1998) (higher fidelity models).

There exists one response to these proposals, which is to treat agent-based systems the same

as non-agent based ones. However, we reject this idea because we think it is more natural to describe agents using a psychological and social language. Therefore, we believe that there is a need for specific methods or representations tailored for agent-based software. This belief originates from the related literature. To give an example, Yu and Liu (2000) say that "an agent is an actor with concrete, physical manifestations, such as a human individual. An agent has dependencies that apply regardless of what role he/she/it happens to be playing." On the other hand, Jennings (2000) defines an agent as "an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives." Also, Wooldridge and Ciancarini (2001) see the agent as a system that enjoys autonomy, reactivity, pro-activeness, and social ability.

Therefore, multi-agent systems (MAS) differ from non-agent based ones because agents are meant to be autonomous elements of intelligent functionality. Consequently, this requires that agent-based software engineering methods encompass standard design activities and representations as well as models of the agent society.

Two more responses exist. They both argue that agents differ from other software but disagree about the differences. The first, proposed by supporters of low-fidelity representations, is that agents are distinguished by their social and epistemological properties, only these need different abstractions. The second, proposed by supporters of high-fidelity representations, is that the difference is in the deployment and interaction mechanisms. With regard to the agent notion, DeLoach, Wood, and Sparkman (2001) argue that "an agent class is a template for a type of agent in the system and is analogous to an object class in object-orientation. An agent is an actual instance of an agent class," and "… agent classes are defined in terms of the roles they will play and the conversations in which they must participate." This definition in some

way conjugates the social- (conversational) and deployment- (implementation) oriented theories and positions DeLoach, Wood, and Sparkman in the middle.

We also reject these two views in their extreme forms. A designer may want to work at different levels of detail when modeling a system. This requires appropriate representations at all levels of detail or fidelity and, crucially, systematic mappings between them. Because such issues are, at present, not addressed by any of the existing MAS analysis and design methodologies, we have decided to create a brand new one.

The methodology we are going to illustrate is named a Process for Agent Societies Specification and Implementation (PASSI) or "steps" in the Italian language. It is our attempt at solving the scientific problem arising from the above considerations. In fact, it is a step-by-step requirement-to-code methodology for designing and developing multi-agent societies integrating design models and concepts from both object-oriented (OO) software engineering and MAS, using the Unified Modeling Language (UML) notation. It is closer to the argument made above for high-fidelity representations, but addresses the systematic mapping between levels of detail and fidelity. The target environment we have chosen is the standard, widely implemented Foundation for Intelligent Physical Agents (FIPA) architecture (O'Brien & Nicol, 1998; Poslad, Buckle, & Hadingham, 2000). PASSI is the result of a long period of theoretical studies and experiments in the development of embedded robotics applications (Chella, Cossentino, & LoFaso, 2000; Cossentino, Sabatucci, & Chella, 2003).

The remainder of this chapter is structured as follows. The next section gives a quick presentation of the methodology's models and provides a justification for PASSI. The third section presents the application of PASSI to the "Juul Møller Bokhandel A/S" case study (Andersen, 1997), giving a detailed description of the steps and the use of UML notations within each of them.

20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/requirements-code-passi-methodology/24298

Related Content

Uncovering Insights by Comparing Machine Learning Algorithms for Customer Churn Analysis
Prashantkumar Mishra, Talha Ali Khan, Arslan A. Rizvi, Rand Kouatly, Iftikhar Ahmedand Thomas Heinrich Musiolik (2024). *Enhancing and Predicting Digital Consumer Behavior with AI (pp. 343-374).*
www.irma-international.org/chapter/uncovering-insights-by-comparing-machine-learning-algorithms-for-customer-churn-analysis/347209

A Brief Review on Deep Learning and Types of Implementation for Deep Learning
Uthra Kunathur Thikshajaand Anand Paul (2018). *Deep Learning Innovations and Their Convergence With Big Data (pp. 20-32).*
www.irma-international.org/chapter/a-brief-review-on-deep-learning-and-types-of-implementation-for-deep-learning/186468

Some Information Measures for Fuzzy Rough Sets
Omdutt Sharma, Pratiksha Tiwariand Priti Gupta (2021). *International Journal of Fuzzy System Applications (pp. 82-102).*
www.irma-international.org/article/some-information-measures-for-fuzzy-rough-sets/277112

Emotion Identification From TQWT-Based EEG Rhythms
Aditya Nalwaya, Kritiprasanna Dasand Ram Bilas Pachori (2022). *AI-Enabled Smart Healthcare Using Biomedical Signals (pp. 195-216).*
www.irma-international.org/chapter/emotion-identification-from-tqwt-based-eeg-rhythms/306956

Robust Target Tracking Algorithm Based on Superpixel Visual Attention Mechanism: Robust Target Tracking Algorithm
Jia Hu, Xiao Ping Fan, Shengzong Liuand Lirong Huang (2019). *International Journal of Ambient Computing and Intelligence (pp. 1-17).*
www.irma-international.org/article/robust-target-tracking-algorithm-based-on-superpixel-visual-attention-mechanism/225767