

Guest-editorial

Shaping the emerging paradigm of agent-orientation: Special issue on agent-oriented software development methodologies

Hong Zhu

Department of Computing, Oxford Brookes University, Oxford OX33 1HX, UK

E-mail: hzhu@brookes.ac.uk

The Oxford English Dictionary defines the word *methodology* as ‘the branch of knowledge that deals with method generally or with the methods of a particular discipline or field of study’. Subsequently, the word *methodolgy* also refers to ‘the study of the suitability of the techniques employed in it’. For software development methodology, the discipline or the field of study is, of course, software development. Thus, it is a branch of knowledge that deals with a method or methods of software development and the study of the suitability of the techniques employed in it. There is no doubt that methodology plays a significant role in scientific study in almost all disciplines. However, the quest for science has often led to a largely futile battle of methodologies in answer to the question: How should we do it? Unfortunately, the study of software development is no exception. The recent rapid development in web technology as well as other computer hardware and communication network technologies provides a computation planform that has never been so powerful before. How to develop software systems that effectively and efficiently utilise such a powerful platform has imposed a challenge to software development methods. The past few years has seen a rapid growth in the research on new software development methods. Agent-orientation is one of the most active and promis-

ing among them. A great amount of work has been reported in the literature on software engineering of agent-based systems. A number of methodologies of software development have been proposed and investigated based on the concept of agent, such as Gaia [1], Tropos [2], and CAMLE [3], just to mention a few. Numerous workshops on this topic have been held at various conferences; see, for example [4,5]. A new international journal on Agent-Oriented Software Engineering has also been launched [6]. However, many fundamental questions concerning agent-orientation remain open and few general consensuses on the basic issues have achieved among researchers. The current state of the research on agent-oriented software engineering is very much a battle field.

This special issue is a direct result of research efforts on agent-oriented software development. In July 2005, the First International Workshop on Agent-Oriented Software Development Methodologies (AOSDM’2005) was successfully held at the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE’05) in Taipei. Five papers were presented at the workshop, which attracted more than 30 attendants with lively discussions after each presentation. There are also more than 20 other papers on related topics presented at the SEKE’05 and

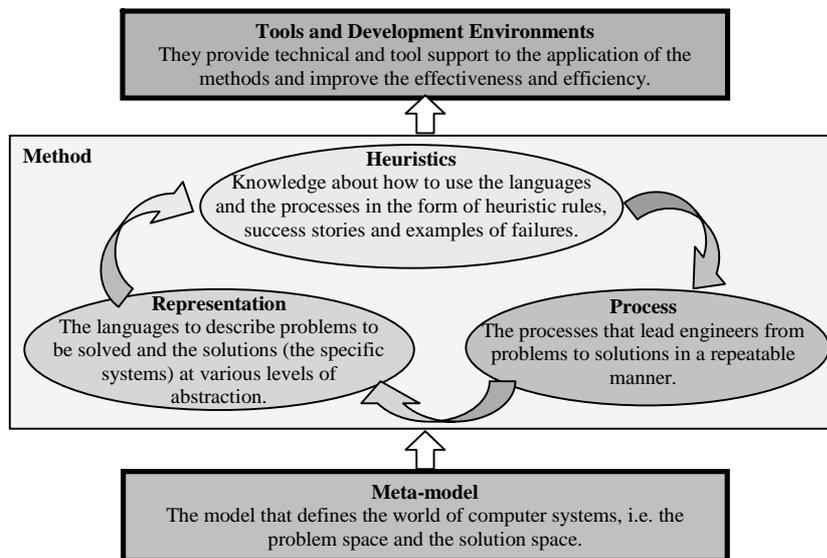


Fig. 1. Elements of software development methodologies.

published in the proceedings. The special issue contains the extended and revised versions of papers selected from them together with papers by invited authors who have been working in this area. Each of these papers has been rigorously reviewed by AOSDM'06 workshop PC members. The main purpose of the workshop and this special issue is to provide a forum for the researchers in the area of agent-oriented software development methodology to exchange ideas and report their most recent results.

As a branch of study on the methods of software development, a methodology may contain knowledge at three levels, as shown in Fig. 1. In the middle is the method of development, which in turn could consist of at least three main parts: (a) a *development process* that defines how development activities should be carried out so that computation problems can be solved; (b) the *representation or languages* that enable the problems as well as solutions, including intermediate solutions, to be represented and expressed; and (c) the *heuristic rules or empirical knowledge* for appropriate uses of the representation and process to solve problems effectively and efficiently. These aspects of agent-oriented methods were addressed in this special issue, where most of the papers involve more than one aspect.

Addressing the process aspect, in the paper titled *Agent-Oriented Conceptual Models to UML Sequence Diagrams via Effect Annotations*, Aneesh Krishna, Aditya K. Ghose, and Aleksandar Vranesovic suggest an approach for supporting the process of obtaining UML sequence diagrams from i^* models via effect an-

notations in order to benefit from the complementary representational capabilities of the two frameworks. In the paper titled *The Agent-Rule-Class Framework for Multi-Agent Systems*, Liang Xiao and Des Greer propose an Agent-Rule-Class framework that uses both object-oriented and agent-based representations and consists of a new process model to support the collaborations between developers and domain experts. They aim at building agents upon traditional OO system components and making use of business rules to dictate agent behaviours aided by the OO components.

The representation and language issues are the main topic of the paper titled *Formal Description and Verification of MAS Interaction Protocols* by Hongbing Chen, Qun Yang, Qianmu Li and Manwu Xu. They propose a calculus for the formal description of interaction protocols of Multi-Agent Systems based on process algebra in order to verify the properties of interaction protocols of heterogeneous agent systems, such as termination and deadlock-free properties.

Heuristic knowledge about various development activities and methods for developing various types of agent-based systems and in various application domains are also addressed in the special issue. The paper entitled *Implementing Flexible and Robust Agent Interactions using Distributed Commitment* by Michael Winikoff focuses on the question of designing flexible and robust interactions between entities such as agents or web services. It presents an alternative formalisation of Yolum and Singh's Commitment Machines, explores their properties and demonstrates how they

can be exploited to implement CM-based interaction in distributed systems. Hong Mei, Xi Sun, Xuanzhe Liu, Wenpin Jiao and Gang Huang, in the paper titled *An Agent-based Approach to Composing Web Services to Support Adaptable Business Processes*, report their method on the composition of business processes to build web services to achieve adaptability. They employ agents to control the executions of business processes via agents' behavior rules, which can be generated automatically and modified dynamically. This paper also describes a reflective middleware that provides the run-time support of lightweight agents. In the paper *A Formal Model-Based Approach for Developing an Interoperable Mobile Agent System*, Junhua Ding, Peter J. Clarke, Dianxiang Xu, Xudong He and Yi Deng investigate the development of mobile agents that provide an effective and flexible approach to developing complex distributed systems by formalizing the software architecture of mobile agent systems using two-layer predicate transition (PrT) nets. They aim at a natural integration of formal methods and practical approaches in each phase of the development life-cycle of agent-based systems. In the paper *An Architecture for an Internet Marketing Multi-Agent System using AUML*, Kai-Yi Chin, Jim-Min Lin, Zeng-Wei Hong, Chih-Wei Lin and Arthur J. Lin report their work on an Internet Marketing Multi-Agent System (IMMAS), which is proposed to enhance traditional Internet marketing mechanism through employing mobile agent techniques. They introduce two software agents, namely a Sales agent and a Personal agent, for performing the delegated jobs of the designated marketing activities and for helping customers to assist administration tasks, respectively. In the paper *Multi-Agent System Methodology for Wireless Sensor Networks*, Richard Tynan, Antonio Ruzzelli and Gregory O'Hare take a Multi-Agent Systems approach to the development and deployment of software to wireless sensor networks in order to address the difficulties due to the limitation of hardware resources on each node of sensor network such as the minimal availability of memory space, energy and display facilities. Their methodology consists of a process model aiming at enabling thorough testing and smooth deployment of software to devices and is supported by a number of tools. The paper also reports a case study with the methodology to illustrate its application.

The *meta-model* level underlying a software development method defines the spaces of the problems and the solutions. It represents the philosophical view of software systems in terms of what constitute software systems and how their constituent elements work to-

gether. For example, object-orientation views the world as consists of objects that are classified by classes. The objects of a system work together through sending messages to call methods. Meanwhile, structure methodology views the world as consists of data and their processing entities that work together through data flows and control flows. For agent-oriented systems, there is no doubt that agents must be constituent elements. However, questions remain with regards to what agents are and how they work. In the paper titled *Approach for Multi-agent Meta-modelling*, Selma Azaiez, Marc-Philippe Huget, and Flavio Oquendo address two aspects of multi-agent meta-modelling. They propose a generic meta-model describing the main concepts of Multi-Agent Systems and a transformation approach that allows designers to manipulate the meta-model in order to adapt to their needs. These transformations are specified within patterns and then applied step by step according to a refinement approach. In the paper titled *DIVAs: Illustrating an Abstract Architecture for Agent-Environment Simulation Systems*, R.Z. Mili, R. Steiner, and E. Oladimeji propose a meta-model in the form of an abstract architecture in which the environments of Multi-Agent Systems are treated as first class entities. They illustrate the concepts through a geographically based social simulation tool DIVAs and demonstrate the flexibility and adaptability of the architecture.

On top of method level, development tools and environments can be constructed to provide the technical supports for the method to be applied to solve real problems effectively and efficiently. There are two papers in this special issue that are devoted to this aspect of methodology. Ontology has been widely recognised as an important vehicle for agent representation and exchange of knowledge, which has also find their application in web-based applications, such as Web Services. Hai Wang, Jin Song Dong, Jing Sun and Jun Sun in the paper *Reasoning Support for Semantic Web Ontology Family Languages Using Alloy* investigate the uses of existing formal modelling tools, in particular Alloy, to provide an automatic reasoning service for the Semantic Web ontology family languages (OWL/SWRL/SWRL-FOL). In the paper titled *Agent-Oriented System Design and Comprehension*, D.N. Lam, J. Ahn and K.S. Barber present the Designer's Agent Creation and Analysis Toolkit (DACAT) for the architectural design phase of the software engineering lifecycle, and the Tracer tool to automate interpretations and explanations of the implementation, thus helping users comprehend implemented agent behaviours.

As Zambonelli and Omicini pointed out [7], agent-orientation is still far away from a mature paradigm of software development. I hope the papers presented in this special issue stimulate the research on agent-oriented software development methodologies and help the shaping up of this emerging paradigm.

References

- [1] F. Zambonelli, N.R. Jennings and M. Wooldridge, Developing multiagent systems: the Gaia Methodology, *ACM TOSEM* **12**(3) (2003), 317–370.
- [2] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos and A. Perini, TROPOS: An Agent-Oriented Software Development Methodology, *Journal of Autonomous Agents and Multi-Agent Systems* **8**(3) (May 2004), 203–236, Kluwer Academic Publishers.
- [3] L. Shan and H. Zhu, CAMLE: A Caste-Centric Agent-Oriented Modelling Language and Environment, in: *Software Engineering for Multi-Agent Systems III*, R. Choren, A. Garcia, C. Lucena and A. Romanovsky, eds, Springer, 2005, pp. 144–161.
- [4] J. Odell, P. Giorgini and J.P. Muller, Agent-Oriented Software Engineering V: 5th International Workshop, AOSE 2004, New York, NY, USA, July 2004, Revised Selected Papers (Lecture Notes in Computer Science, 3382), Springer; January 24, 2005.
- [5] R. Choren, A. Garcia, C. Lucena and A. Romanovsky, eds, Software Engineering for Multi-Agent Systems III · Research Issues and Practical Applications, *Lecture Notes in Computer Science* **3390** (2005), Springer.
- [6] International Journal of Agent-Oriented Software Engineering (IJAOSE), ISSN (Online): 1746-1383 – ISSN (Print): 1746–1375, <http://www.inderscience.com/browse/index.php?journalID=174>.
- [7] F. Zambonelli and A. Omicini, Challenges and Research Directions in Agent-Oriented Software Engineering, *Autonomous Agents and Multi-Agent Systems* **9** (2004), 253–283.