

Chapter 26

Formal Approach to Ensuring Interoperability of Mobile Agents

Linas Laibinis

Åbo Akademi University, Finland

Elena Troubitsyna

Åbo Akademi University, Finland

Alexei Iliasov

Newcastle University, UK

Alexander Romanovsky

Newcastle University, UK

ABSTRACT

Mobile agent systems are complex distributed systems that are dynamically composed of autonomous agents. Since agents are often developed independently, they may lack interoperability, i.e., do not communicate in a correct way. In this chapter, the authors propose a formal approach to ensuring interoperability of agents implemented by independent developers. The essence of the approach is the decomposition of a specification of a multi-agent application into a set of specifications of agent roles. While decomposing the specification, they explicitly define the communication mechanisms between the agents. To ensure interoperability, the implementation of each agent should adhere to its formal specification. However, each agent can be implemented completely independently, i.e., without knowing the specifications of other agents. The authors use refinement in the Event-B framework to define formally the process of decomposing a multi-agent application into a set of interacting roles. The approach is illustrated by a case study involving the development of an electronic auction.

1 INTRODUCTION

Mobile agent systems have emerged as a result of the integration of computing and communication. Such systems are examples of the complex distributed systems that are dynamically composed of independent agents. The complexity associated with these systems makes designing mobile

agent software notoriously error prone. Since agents are usually developed independently of each other, one of the major problems is a lack of agent interoperability, resulting in a mismatch in agent communication. In this paper we propose a formal approach to designing mobile agent systems, which allows the designer to ensure agent interoperability.

DOI: 10.4018/978-1-61520-655-1.ch026

In our approach, agent development starts with a formal specification of a scope, which is the logical unit structuring the behaviour of a multi-agent application. Each agent plays a certain role in a scope. We formally specify the behaviour in the scope and decompose it into a set of role specifications by refinement. We use the Event-B framework (Abrial, 2000) as our formal basis. It is an extension of the B Method, supporting reasoning about reactive and distributed systems. The communication between agents is introduced in the process of decomposition. As a result of decomposition, all the information about the role functionality and the communication mechanisms is contained within a role specification. Hence, if each agent implementation adheres to their role specifications then the implementations of the agents are compatible with each other, i.e., interoperability is achieved.

Our approach enables decentralised development of agent systems. Developers of individual agents do not have to communicate with each other or disclose the agent source code in order to produce agents that are able to collaborate. We see this as the main prerequisite for building and deploying large-scale agent systems where new agents with different yet compatible functionalities can be created at any moment.

As a result of uncertainty about behaviour of other agents, developers often have to rely on the extensive use of checks on incoming messages. With our approach most of such checks and assertions can be eliminated. This results in simpler and more efficient agents, which are easier to implement and understand. Moreover, our approach supports reuse, since a specification of a role should be developed once, but can be reused many times. Furthermore, the formal specifications are language neutral and in many aspects implementation neutral. Therefore, a developed specification is usually well-suited to any combination of platforms, middlewares, and programming languages.

We proceed as follows. In Section 2 we introduce the Context-Aware Mobile Agents middleware (CAMA), for which our approach is adapted. In Section 3 we give an overview of Event B, the chosen formal development method. Section 4 presents our main contribution – the methodology for formal specification and decomposition of multi-agent applications. Section 5 demonstrates application of our approach to the decentralized development of agents from formal role specifications by presenting a case study involving an electronic auction. Finally, we discuss the proposed approach, and provide an overview of the related and future work in Section 6.

2 BACKGROUND: CAMA MIDDLEWARE

Context-Aware Mobile Agents (CAMA) systems (Iliasov & Romanovsky, 2005; Arief, Iliasov & Romanovsky, 2006; Iliasov, Romanovsky et al., 2007) are defined via a set of abstractions and operations modeling inter-agent communication and operability. The primary goal of defining a CAMA system is to offer programmers a formally-verified basis for rapid development of mobile agent software in a disciplined and structured way.

CAMA inter-agent communication is based on the LINDA paradigm (Gelernter, 1985). LINDA provides a set of coordination primitives that can be used for coordination of several independently running pieces of software. Since LINDA is language independent, it became quite popular and its coordination primitives have been implemented in many programming languages. Moreover, LINDA supports asynchronous and anonymous agent communication and hence is well-suited for mobile agent systems.

A CAMA system consists of a set of *locations*. The main role of a location is to provide the inter-agent communication service to its client agents. The communication service is based on a shared blackboard supporting LINDA operations.

19 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/formal-approach-ensuring-interopability-mobile/66482

Related Content

VORDi*: A Combined Approach for Requirements Engineering of Software as a Service

Farida Kherissi (2021). *International Journal of Information System Modeling and Design* (pp. 1-20).

www.irma-international.org/article/vordi/288553

Using Dynamic Time Warping to Detect Clones in Software Systems

Mostefai Abdelkader (2021). *International Journal of Software Innovation* (pp. 20-36).

www.irma-international.org/article/using-dynamic-time-warping-to-detect-clones-in-software-systems/266280

Monitoring Buffer Overflow Attacks: A Perennial Task

Hossain Shahriar and Mohammad Zulkernine (2012). *Security-Aware Systems Applications and Software Development Methods* (pp. 130-151).

www.irma-international.org/chapter/monitoring-buffer-overflow-attacks/65846

Sentilyser: Embedding Voice Markers in Homeopathy Treatments

Arunima Mookherjee, Preeti Mulay, Rahul Joshi, Pranav Sunil Prajapati, Sonali Johari and Swati Sunil Prajapati (2019). *Interdisciplinary Approaches to Information Systems and Software Engineering* (pp. 181-206).

www.irma-international.org/chapter/sentilyser/226400

Design Principles for Reference Modelling: Reusing Information Models by Means of Aggregation, Specialisation, Instantiation and Analogy

Jan vom Brocke (2009). *Innovations in Information Systems Modeling: Methods and Best Practices* (pp. 269-296).

www.irma-international.org/chapter/design-principles-reference-modelling/23794