Chapter VIII

Extension of the Unified Modeling Language for Mobile Agents

Cornel Klein
Siemens ICN, Munich, Germany

Andreas Rausch, Marc Sihling and Zhaojun Wen
Institut für Informatik, Technische Universität München, Germany

Mobile agents gained immense attraction as a new programming concept for implementing distributed applications. However, up to now mobile agent programming has been mainly technology driven, with a focus on the implementation of mobile agent platforms and only small programming applications. In this chapter, we present an extension of the standard UML that provides language concepts for modeling mobility both in analysis and design phases. This extended version of UML is applied to the modeling of an advanced telecommunication system.

INTRODUCTION

Mobile agents are software entities that can migrate autonomously on a network from host to host. As a result of this intrinsic characteristic, they raise considerable interest as a new concept for networked computing. The often-cited advantages attributed to mobile agents are: reduction of network traffic, load balancing, fault tolerance, asynchronous interaction, data access locality and flexible distribution of intelligence in a network.

In the past, development of mobile agent concepts was mostly technology-driven. In particular since the invention of Java, which allows to easily move an agent’s code even in heterogeneous networks, dozens of mobile agent platforms have been developed both within the research community as well as within industrial projects (Bäumer, Breugst, Choy, & Magedanz, 1999; Chess, 1995; Chang, & Lange, 1996; Jul, Levy, Hutchinson, & Black, 1988; Krause, 1997). On the more theoretical side, authors like Luca Cardelli invented formalisms for modeling and reasoning about mobile agent systems. Cardelli and others argue that mobility is not only important as a programming concept, but that it is
inherently introduced by the advent of mobile devices such as laptops, mobile phones, PDAs, etc. and hence has to be appropriately represented in corresponding models.

However, in spite of these activities there is yet little evidence of an engineering approach to the development of mobile agent-based applications. At present, the majority of existing agent-based applications are created in an ad-hoc fashion, following little or no rigorous design methodology and producing only limited specifications with respect to requirements or design aspects of mobile agents. That might be because of the vast lack of appropriate modeling concepts in standard languages like the Unified Modeling Language (UML). This pertains even to most basic characteristics, for instance, concepts for mobility or cloning.

In this chapter, we present an extension of the UML with a focus on mobile agents. In the next section we start with a discussion on basic concepts and notions of mobile applications. Then, we introduce a small application example that is used to motivate and illustrate the UML extensions. From those we choose a set of extensions and present them in greater detail. We end up with a table showing all additional language elements for mobile applications for each UML diagram. A discussion about the future trends and a short conclusion ends the chapter.

BACKGROUND

The term “agent” is associated with various expectations and used in many different contexts. Therefore, there is no common understanding of the involved concepts. A good starting point to describe the idea of agents is to find what makes them different from “ordinary” components (e.g. mobility, cooperativity, autonomy, etc.). However, we do not want to discuss the characteristics of agents in detail but believe that “mobility” is the most prominent, promising and advanced feature. Mobility provides clear advantages as a programming concept and thus should be seen as an important concept within the early phases of the development process by modeling languages such as the UML. For this reason, we focus on “mobile agents” in the following.

Mobile agents promise advantages over other existing computing paradigms such as client/server programming or distributed systems. In particular, the ability to move allows mobile agents to bring the computation to the data instead of the data to the computation. In many situations, this reduces network traffic extremely. Hence, some kind of applications may be more efficient utilizing mobile agents. They might also profit from the agents' high degree of autonomy: mobile agents can operate asynchronously and independently from the user or the requesting program. This allows, for instance, a mobile device to dispatch an autonomous diagnostic and search agent into some network and then to disconnect. Some time later, it might reconnect to the network to collect the results of the issued query. A deeper discussion of the benefits of mobile agents can be found in Chess, (1995); Hurst, Cunningham, & Somers, (1997); Krause, (1997); and Chang, & Lange, (1996).

Note that any task that can be performed with mobile agents can also be realized using existing technologies and concepts but the traditional solution might be less flexible, less efficient or much more difficult to deploy.

Some advantages of the mobile agent paradigm have been discussed. They stem from the capability of the involved concepts to reduce network usage, increase asynchrony between clients and servers, to add client-specific functionality to servers and to introduce concurrency. Several application types may exploit these advantages, for instance, workflow applications, groupware systems, electronic commerce applications, personal
Related Content

Enhancing RUP Business Model with Client-Oriented Requirements Models
[www.irma-international.org/chapter/enhancing-rup-business-model-client/30539/](http://www.irma-international.org/chapter/enhancing-rup-business-model-client/30539/)

Version Control of Software Models
[www.irma-international.org/chapter/version-control-software-models/4930/](http://www.irma-international.org/chapter/version-control-software-models/4930/)

The Debugging of Rule Bases
[www.irma-international.org/chapter/debugging-rule-bases/35864/](http://www.irma-international.org/chapter/debugging-rule-bases/35864/)

Security Issues in Outsourced XML Databases
[www.irma-international.org/chapter/security-issues-outsourced-xml-databases/27784/](http://www.irma-international.org/chapter/security-issues-outsourced-xml-databases/27784/)

The CORAS Methodology: Model-based Risk Assessment Using UML and UP
[www.irma-international.org/chapter/coras-methodology-model-based-risk/30550/](http://www.irma-international.org/chapter/coras-methodology-model-based-risk/30550/)