Available online at www.sciencedirect.com
ScienceDirect
Computer Standards & Interfaces 29 (2007) 481 – 498
COMPUTER STANDARDS & INTERFACES
www.elsevier.com/locate/csi

Writing and executing ODP computational viewpoint specifications using Maude

José Raúl Romero, Antonio Vallecillo *, Francisco Durán

Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga, Spain

Received 2 February 2006; received in revised form 27 October 2006; accepted 11 November 2006
Available online 19 January 2007

Abstract

The Reference Model of Open Distributed Processing (RM-ODP) is a joint standardization effort by ITU-T and ISO/IEC for the specification of large open distributed systems. RM-ODP is becoming increasingly relevant now because the size and complexity of large distributed systems is challenging current software engineering methods and tools, and because international standards have become key to achieve the required interoperability between the different parties and organizations involved in the design and development of complex systems. RM-ODP defines five viewpoints for decomposing the design activity into separate areas of concern. One of the RM-ODP viewpoints, the computational viewpoint, focuses on the basic functionality of the system and its environment, independently of its distribution. Although several notations have been proposed to model the ODP computational viewpoint, either they are not expressive enough to faithfully represent all its concepts, or they tend to suffer from a lack of formal support. In this paper we introduce the use of Maude as a formal notation for writing and executing ODP computational viewpoint specifications. Maude is an executable rewriting logic language specially well suited for the specification of object-oriented open and distributed systems. We show how Maude offers a simple, natural, and accurate way of modeling the ODP computational viewpoint concepts, allows the execution of the specifications produced, and offers good tool support for reasoning about them.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Rewriting logic; Maude; RM-ODP; ODP computational viewpoint; Environment contracts

1. Introduction

Viewpoint modeling is becoming an effective approach for dealing with the inherent complexity of large distributed systems. Current software architectural practices, as described in IEEE Std. 1471 [18], divide the design activity into several areas of concerns, each one focusing on a specific aspect of the system. Examples include the "4+1" view model [22], the Zachman's framework [40], or the Reference Model of Open Distributed Processing (RM-ODP) [19].

In particular, the rapid growth of distributed processing has led to the adaptation of the RM-ODP framework, which is a joint standardization effort by ISO/IEC and ITU-T that creates an architecture within which support of distribution, interworking and portability can be integrated. Several years after its final adoption as ITU-T Recommendation and ISO/IEC International Standard, the Reference Model of Open Distributed Processing is increasingly relevant, mainly because the size and complexity of current IT systems is challenging most of the current software engineering methods and tools. These methods and tools were not conceived for use with large, open and distributed systems, which are precisely the systems that the RM-ODP addresses. In addition, the use of international standards has become the most effective way to achieve the required interoperability between the different parties and organizations involved in the design and development of complex systems. As a result, we are now witnessing many major companies and organizations investigating RM-ODP as a promising alternative for specifying their IT systems, and for structuring their large-scale distributed software designs.

The RM-ODP framework provides five generic and complementary viewpoints on the system and its environment: *enterprise*, *information*, *computational*, *engineering* and *technology*

* Corresponding author.
E-mail addresses: jrromero@lcc.uma.es (J.R. Romero), av@lcc.uma.es (A. Vallecillo), duran@lcc.uma.es (F. Durán).

0920-5489/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2006.11.004

viewpoints. They allow different stakeholders to observe a system from different perspectives [23].

The RM-ODP is a general framework, and therefore it was conceived as both notation and methodology independent. However, the fact that the Reference Model does not prescribe any specific notation for representing its concepts and viewpoint languages really hinders the development of commercial tools for writing and analyzing ODP system specifications. Several notations have been proposed for the different viewpoints by different authors, which seem to agree on the need to represent the semantics of the ODP viewpoints concepts in a precise manner [3,7,19,23,37]. For example, formal description techniques such as Z and Object-Z have been proposed for the information and enterprise viewpoints [39], and LOTOS, SDL or Z for the computational viewpoint [19,38].

This paper is the third of a trilogy that explores the use of rewriting logic and Maude [9] for specifying ODP viewpoints. Maude is an executable rewriting logic language specially well suited for the specification of object-oriented open and distributed systems. We have already used Maude for successfully modeling the ODP enterprise viewpoint [12] and the information viewpoint [11]. Here we shall show how rewriting logic, and in particular Maude, provide the expressiveness required for modeling ODP computational viewpoint specifications.

The use of Maude provides additional advantages. The fact that rewriting logic specifications are executable allows us to apply a flexible range of increasingly stronger formal analysis methods and tools, such as run-time verification [17], model checking [13], or theorem proving [10]. Maude offers a comprehensive toolkit for automating such kinds of formal analysis of specifications. In addition, Maude offers support for the specification of real-time properties of systems, which are required to capture some quality of service (QoS) and other temporal constraints, that form part of the environment contracts of a computational specification. These kinds of contracts are essential in many service-based architectures (such as SOA) for modeling service level agreements and other QoS characteristics of distributed systems.

The structure of this document is as follows. First, Sections 2–4 serve as a brief introduction to the RM-ODP, the ODP computational viewpoint and Maude, respectively. Then, Section 5 presents our proposal, describing how to write computational specifications in Maude. Section 6 is dedicated to a case study that illustrates our approach. Section 7 compares our work with other related proposals and, finally, Section 8 draws some conclusions and outlines some future research activities.

2. Introduction to the RM-ODP

Distributed systems can be very large and complex, and the many different considerations which influence their design can result in a substantial body of specification, which needs a structuring framework if it is to be managed successfully. The purpose of the RM-ODP is to define such a framework.

RM-ODP is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture. The framework for system specification provided by the RM-ODP has four fundamental elements:

(1) an object modelling approach to system specification;
(2) the specification of a system in terms of separate but interrelated viewpoint specifications;
(3) the definition of a system infrastructure providing distribution transparencies for system applications;
(4) a framework for assessing system conformance.

Most complex system specifications are so extensive that no single individual can fully comprehend all aspects of the specifications. Furthermore, we all have different interests in a given system and different reasons for examining the systems specifications. A business executive will ask different questions of a system make-up than would a system implementor. The concept of RM-ODP viewpoints framework, therefore, is to provide separate viewpoints into the specification of a given complex system.

The RM-ODP framework provides five generic and complementary viewpoints on the system and its environment:

- The enterprise viewpoint, which focuses on the purpose, scope and policies for the system and its environment. It describes the business requirements and how to meet them.
- The information viewpoint, which describes the information managed by the system and the structure and content type of the supporting data.
- The computational viewpoint, which describes the functionality provided by the system and its functional decomposition in terms of objects which interact at interfaces.
- The engineering viewpoint, which focuses on the mechanisms and functions required to support distributed interactions between objects in the system.
- The technology viewpoint, which focuses on the choice of technology of the system. It describes the technologies chosen to provide the processing, functionality and presentation of information.

Each of these viewpoints satisfies an audience with interest in a particular set of aspects of the system. Associated with each viewpoint is a viewpoint language that optimizes the vocabulary and presentation for the audience of that viewpoint.

Although separately specified, the viewpoints are not completely independent; key items in each are identified as related to items in the other viewpoints. However, the viewpoints are sufficiently independent to simplify reasoning about the complete specification. The mutual consistency among the viewpoints is ensured by the architecture defined by RM-ODP, and the use of a common object model provides the glue that binds them all together.

3. The computational viewpoint

The computational viewpoint describes the functionality of the ODP system and its environment, in terms of objects performing individual functions and interacting at well-defined