Different FDT’s Confronted with Different ODP-Viewpoints of the Trader

Joachim Fischer, Andreas Prinz and Andreas Vogel *

Department of Informatics
Humboldt-University Berlin
Unter den Linden 6
P.O. Box 1297
Germany, Berlin D-O-1086
\{fischer, prinz, avogel\}@informatik.hu-berlin.de

Abstract. The Reference Model of Open Distributed Processing (ODP RM) is intended to create an international standard for the design and realization of open distributed systems by both ISO and CCITT.

The use of formal methods in the design process of ODP systems is explicitly required. In this article the use of the Formal Description Techniques (FDT's) $Z$, LOTOS and SDL'92 is investigated and evaluated. These three FDT’s are considered as representatives of the classes of formal techniques characterized by their structuring concepts and their underlying models, logic (set theory), process algebras and finite state machines, respectively.

The ODP trader which is intended to be a standard as well is selected as case study.

1 Introduction

1.1 Open Distributed Processing

The Reference Model of Open Distributed Processing (ODP RM) is intended to create an international standard for the design and realization of open distributed systems by both ISO and CCITT.

The ODP RM is organized in five parts, three defining and two giving explanations. The three defining parts are

- part 2 Descriptive model,
- part 3 Prescriptive model, and

Whereas the descriptive model defines concepts for modeling of distributed systems in general, the prescriptive model defines the constraints which an ODP-system has to fulfill. Specification techniques and styles are investigated with respect to their suitability in the ODP context in part 5.

Rather than to deal with a distributed system in its full complexity, five viewpoints, abstractions on different levels, are defined in the ODP RM. These are the enterprise, information, computation, engineering and technology viewpoint.

* supported by a postgraduate grant of the Deutscher Akademischer Austauschdienst
**Enterprise Viewpoint.** The enterprise viewpoint is concerned with the following objectives:

- the rules and activities that exist within the enterprise,
- the interaction between the system and its environment,
- organizational structure of the enterprise,
- artifacts used in the enterprise rules,
- types of processing done in the enterprise, and
- security and management policies used in the enterprise.

The system is modeled in terms of enterprise objects. These objects represent the user rules, the business and management policies, the ODP system itself and its environment. The behaviour is defined by sequencing enterprise activities.

**Information Viewpoint.** From the information viewpoint are visible:

- information elements, their structure, their quality attributes and the relationships between them,
- the information flow,
- how information and information processing is visible to the user, and
- logical partitions of the ODP system.

An ODP system from the information viewpoint has to be modeled in terms of (information) objects, templates and classes. An object is defined by its states.

**Computation Viewpoint.** From this viewpoint, processing functions (instantiation, assignment, invocation, synchronization, communication, etc.) and data types are visible. A specification from the computation viewpoint describes a system independently of computers and networks.

The system is modeled by computation objects. Computation objects offer computation operations through computation interfaces. These operations are defined by one initial and a finite set of termination events. The invocation of an operation is of type interrogative or announcement.

**Engineering Viewpoint.** Engineering viewpoint specifications deal with control and transparency mechanisms, processors, memory, and communication networks. The distribution of behaviour and data is specified.

The main modeling concept are engineering objects. Computation objects are templates for the engineering ones. Furthermore, transparency and nucleus objects are visible.

- nucleus objects: provide a basic set of processing, storage and communication functions.
- transparency objects: provide transparency functions; there are transparency functions with respect to access, location, migration, concurrency and livelyness.
- engineering objects: interact with each other via transparency and nucleus objects.