The use of ODP in MDA system specifications

Bryan Wood
Convenor, JTC1/SC7/WG19, Modelling Languages and Open Distributed Processing
Bryan.Wood@Open-IT.co.uk

with acknowledgements to
Antonio Vallecillo, Universidad de Málaga

Agenda

- ODP* system specifications
- Use of UML for ODP system specifications
  - What it is
  - Example
- ODP in MDA system specifications
- Sources and progress so far

*ODP = Open Distributed Processing
The Reference Model of ODP (ITU-T Rec X.901-904 | ISO/IEC 10746) defines a framework for system specification, covering all aspects of open distributed systems:

- “enterprise” context, functionality, distribution, infrastructure, technology
- It comprises
  - a structure for system specifications in terms of viewpoints
  - a language (concepts and rules) for expressing each viewpoint specification
  - a set of object-oriented foundation modeling concepts common to all viewpoint languages

Different abstractions of the same system

- each abstraction focuses on different concerns
- each abstraction achieved using a set of viewpoint concepts and rules

A mechanism for dealing with the complexity of distributed systems
An ODP system specification comprises a set of viewpoint specifications (or views).

A viewpoint specification:
- is a specification of a system from a specific viewpoint;
- uses language constructs for the viewpoint to express the concerns and decisions covered by the viewpoint specification;
- is related to, and consistent with, other viewpoint specifications.

ODP viewpoint specifications:
- Enterprise
- Information
- System
- Computation
- Engineering
- Technology
The enterprise specification

- Specifies the roles played by an IT system in its organisational environment
- An object model of a social/commercial organisation in terms of:
  - enterprise objects
  - communities (of enterprise objects)
    - objectives
    - behaviour
      - roles (fulfilled by enterprise objects in a community)
      - processes (meeting objectives)
    - Policy
    - ...

The information specification

- Specifies system behaviour to meet its objectives abstracted from implementation
- An object model of the system describing the semantics of information and of information processing in the system in terms of:
  - information objects
  - invariant schema - predicates on information objects that must always be true
  - static schema - state of information objects at some location in time
  - dynamic schema - allowable state changes of information objects
The computational specification

- Specifies computational structure in terms of units of functionality and distribution and their interactions
- An object model of the system describing the structure of processing in terms of:
  - computational objects
  - interfaces: operations supported
  - invocations: operations invoked
  - computational bindings
  - environmental contracts: QoS constraints
  - ...

The engineering specification

- Specifies the mechanisms and services to provide the distribution transparencies and meet QoS constraints required by the system
- An object model of the system describing the infrastructure supporting the computational structure
  - basic engineering objects
  - (infrastructure) engineering objects
  - clusters, capsules, nodes
  - channels
  - functions
The technology specification

- Specifies the hardware and software pieces from which the system is built.
- An object model of the system
  - defining the configuration of technology objects that comprise the ODP system, and the interfaces between them
  - identifying conformance points

An ODP system specification

- business context
- business processes
- information
- changes to information
- constraints
- object configuration
- interactions between objects at interfaces
- mechanisms and services to provide the required distribution transparencies and QoS constraints.
- hardware and software components implementing the system
Correspondences between ODP views

Enterprise

Information

Computational

Engineering

Technology

UML for ODP system specifications

- A standard defining:
  - a set of UML Profiles for expressing a system specification in terms of viewpoint specifications
  - possible relationships between the resultant ODP viewpoint specifications, and how they are represented
  - the structure of a system specification expressed as a set of UML models using ODP viewpoint profiles
- A standard that enables the use of MDA tools in developing and maintaining ODP system specifications

ITU-T Rec. X. 906 | ISO/IEC 19793 Use of UML for ODP system specifications
UML for ODP system specifications

- Why?
  - RM-ODP is notation- and methodology-independent
  - Which is an advantage (a-priori) ...
  - ...but hampers its widespread adoption and use

- Target audiences for ISO/IEC 19793
  - UML Modellers
    - who need to structure (somehow) LARGE system specifications
  - ODP Modellers
    - who need some (graphical) notation for expressing their ODP specifications and tool support
  - Tool vendors

UML Profiles for ODP Viewpoint Languages

Universe of Discourse (UOD)

is a model of (see RM-ODP)

ODP specification

maps to (defined in this document)

UML model

is used to express (see UML spec)

UML notation
Enterprise metamodel (excerpt)

UML Profile – Enterprise lang. (excerpt)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Stereotype</th>
<th>UML Metaclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP System</td>
<td>«EV_ODPSystem»</td>
<td>Class</td>
</tr>
<tr>
<td>Field of Application</td>
<td>«EV_FieldOfApplication»</td>
<td>Comment attached to Package</td>
</tr>
<tr>
<td>Community</td>
<td>«EV_Community»</td>
<td>Subsystem</td>
</tr>
<tr>
<td>Enterprise Object</td>
<td>«EV_EnterpriseObject»</td>
<td>Class</td>
</tr>
<tr>
<td>Enterprise Object fulfilling Role</td>
<td>«EV_FulfilsRole»</td>
<td>Association</td>
</tr>
<tr>
<td>Community Object</td>
<td>«EV_CommunityObject»</td>
<td>Class</td>
</tr>
<tr>
<td>Objective</td>
<td>«EV_Objectives»</td>
<td>Class</td>
</tr>
<tr>
<td>Role</td>
<td>«EV_Roles»</td>
<td>Class, StateMachine (in role model), Partition (in process model)</td>
</tr>
<tr>
<td>Action</td>
<td>«EV_Action»</td>
<td>State</td>
</tr>
<tr>
<td>Interaction</td>
<td>«EV_Interaction»</td>
<td>Class, StateMachine, ActivityGraph</td>
</tr>
<tr>
<td>Process</td>
<td>«EV_Process»</td>
<td>Class, ActivityGraph</td>
</tr>
<tr>
<td>Step</td>
<td>«EV_Steps»</td>
<td>ActionState</td>
</tr>
<tr>
<td>Artefact</td>
<td>«EV_Artefact»</td>
<td>ObjectFlowState (in process model), Signal (in role model)</td>
</tr>
</tbody>
</table>
UML Profile – Enterprise lang. (icons)

- «EV_EnterpriseObject»
- «EV_ODPSystem»
- «EV_Role»
- «EV_Interaction»
- «EV_Artefact»
- «EV_CommunityObject»
- «EV_Community»
- «EV_Objective»
- «EV_Process»
- «EV_Step»

Information Language metamodel
UML Profile – Information Language

- Profile
  - IV_Profile
    - Class
      - IV_Object
    - Constraint
      - IV_InvariantSchema
    - Package
      - IV_StaticSchema
      - locationInTime: Date
    - StateMachine
      - IV_DynamicSchema
    - Signal
      - IV_ActionType
    - Object
      - IV_ObjectType
      - IV_ObjectTemplate

UML Profile – Information lang. (icons)

- Information Spec
- IV_InformationObject
- IV_InformationObjectType
- IV_ActionType
- IV_InvariantSchema
- IV_StaticSchema
- IV_DynamicSchema
Proof of concept: the Templeman Library

- The standard will include, as an example, a partial specification for a computerized system that supports the operations of a University Library, in particular those related to the borrowing process of the Library items.
- The system should keep track of the items of the University Library, its borrowers, and their outstanding loans.
- The library system will be used by the library staff (librarian and assistants) to help them record loans, returns, etc.
UML specification of the ODP system

Enterprise Specification
Enterprise spec – Global

Enterprise spec – Library Community

[Diagram of Enterprise spec – Library Community]

[Diagram of Enterprise spec – Global]

[Comment: Templeton Library System (E-Spec)]

This specification manages an Academic environment, such as a University, in which a Library maintains a collection of books and periodicals that are to be borrowed by the different University members, and that uses a computerized system to keep track of the Library items, loans, and returning items.
Enterprise spec – enterprise objects

Enterprise spec – Role fulfilment
Enterprise spec – Processes

Borrowing processes:
- Borrow Item
- Return Item

Fine processes:
- Fine
- Pay Fines

Suspense processes:
- Suspend borrower
- Re-instate borrower

Administrative processes:
- Add borrower
- Release Item
- Add Item
- Remove borrower

Enterprise spec – Activity graph - Borrow Item

[Diagram of activity graph for Borrow Item process]
Enterprise spec – Interactions

Enterprise spec – Request Item interactions
Enterprise spec – state diagram for Borrower role

- Loan requirement
  - Has loan
  - Does not have loan

Enterprise spec – Artefact roles of Loan

- Loan request
- Loan refusal
- Item loaned
Undergraduate students are not allowed to borrow more than a set number of books. This maximum number of books may vary from time to time and is determined by the Librarian.

Information Specification
Information spec – Action types

context Library inv undergradLimits:
  (undergradMaxLoans = 8) and
  (undergradBookLoanPeriod = 28) and
  (undergradPeriodicalLoanPeriod = 0)

context Library inv uniqueIdentifiers:
  self.items->forAll (itm1,itm2 | itm1.id <> itm2.id) and
  self.users->forAll (usr1,usr2 | usr1.id <> usr2.id)

context Library inv oneLibrarianAndOneClockWhileOpen:
  self.isOpen implies
    (self.Librarian->size() = 1) and (self.Calendar->size() = 1)

context Library inv consistentNumberOfLoans:
  self.users.borrowedItems->sum() = self.loans->size()

context Loan inv wellFormedLoans:  issueDate <= dueDate
ODP in MDA system specifications

What is MDA…

- An approach to system development using models as a basis for understanding, design, construction, deployment, operation, maintenance and modification
- Three essential elements:
  - specifying a system independently of the platform that supports it,
  - specifying platforms,
  - transforming the system specification into one for a particular choice of platform.
- Three primary goals: portability, interoperability and reusability
- Prescribes the kinds of model to be used in specifying a system, how those models are prepared and the relationships between them
What MDA does

- Identifies different viewpoints on a system
  - different abstractions - reflecting different concerns
  - providing a way of dealing with system complexity
- Specifies three kinds of viewpoint model for a system:
  - a computation independent model (CIM): a view of a system that specifies its function without specifying details of its structure
  - a platform independent model (PIM): a view of a system that specifies its computational structure independent of any specific platform - usable with different platforms of similar type.
  - a platform specific model (PSM): a view of a system that combines the specifications in the PIM with a specification of the use of a particular type of platform.
- Specifies transformations between models

What MDA does not do

MDA does not offer:
- a definition of the concerns and design decisions to be covered by each MDA model
- language constructs to express the concerns and decisions covered by each MDA model

... but ODP can offer:
- a definition of the concerns and design decisions to be covered by each MDA model
- language constructs to express the concerns and decisions covered by each MDA model
ODP system specifications and the MDA

ODP and MDA together offer

An IT based approach to system development that provides a framework for:

- separating and integrating different system concerns
- combining skills and experience
- assigning responsibilities
- automating development
(Some) Sources

- Japanese Association of Healthcare Information System Industry (JAHIS) - Hospital Information Reference Enterprise Model project
- European research projects:
  - e.g. COMBINE - investigating the organisation and process for component-based system development
- Industrial Practice
- OMG
  - UML profile for Enterprise Distributed Object Computing (EDOC)
- A worked example for the standard

X.906 | ISO/IEC 19793 Progress and Targets

- Start of Project                May 2003
- SC7 WD                        May 2004     SC7 meeting
- 1st CD                        Dec 2004
- FCD                           May 2005     SC7 meeting
- FDIS                          Dec 2005
- IS                            May 2006     SC7 meeting

(1st CD is available as ISO-stds/04-06-01)
Questions

?