

Open Distributed Processing in SC7

Peter Linington and Antonio Vallecillo

JTC1/SC7/WG19

“Techniques for the specification of IT Systems”

Peter.Linington@kent.ac.uk av@lcc.uma.es

Agenda

- Intro to WG19
- ODP* system specifications
- Use of UML for ODP system specifications
- Revision of the RM-ODP

**ODP ≡ Open Distributed Processing*

WG19 Mission

- The development of standards to enable the integration of business and IT system specifications, and to facilitate the provision of software and system engineering tools and techniques to implement information systems.
- Fundamental to this objective is the recognition that information systems must be realized in an environment where data and processing are distributed across heterogeneous IT resources and multiple organizational domains.

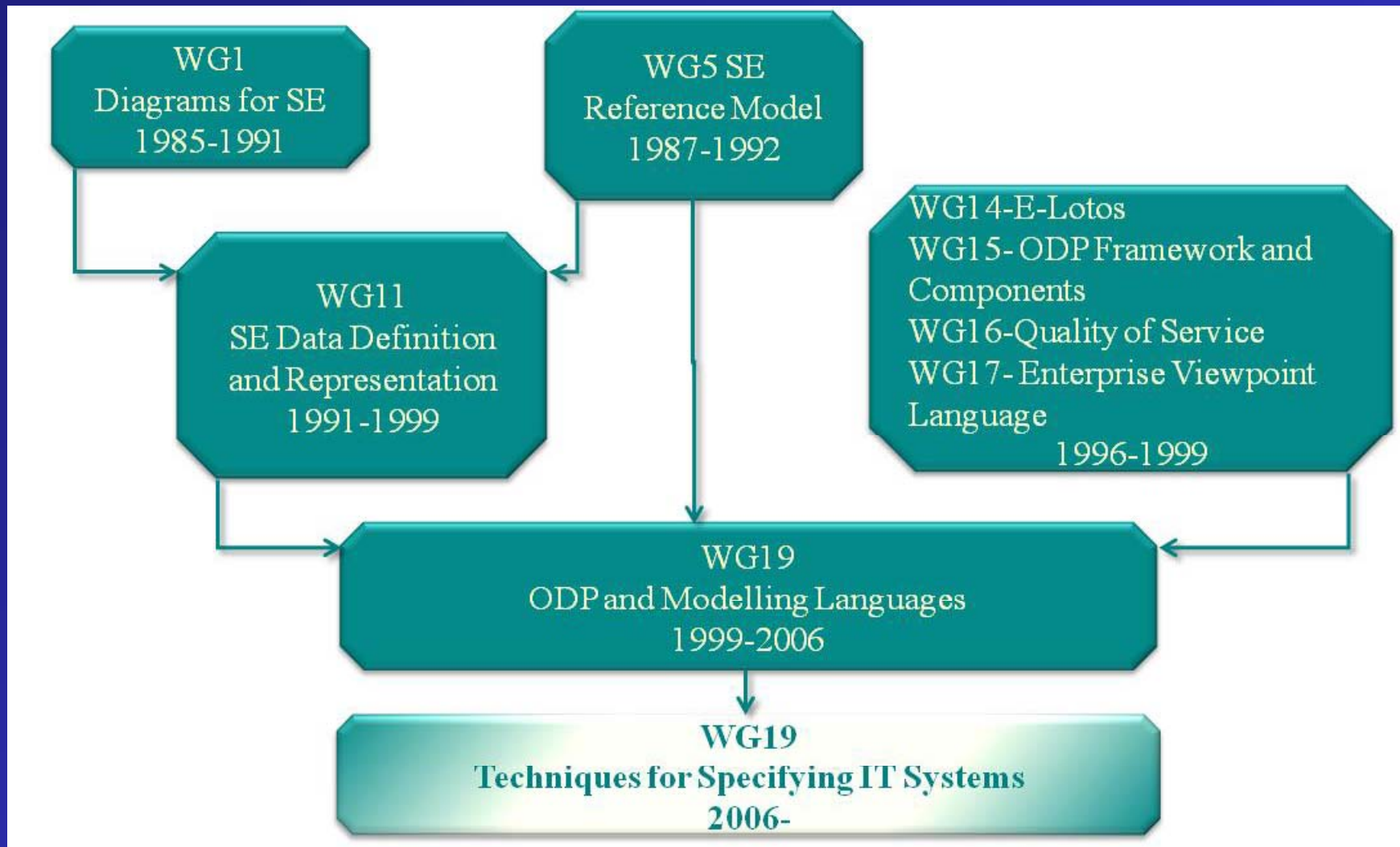
WG19 Terms of Reference

- WG19 is responsible for standards for rigorously specifying information systems covering:
 - **architectural frameworks** for distributed processing systems, defining the structure of system specifications for both complete systems and for specific areas such as naming, security and conformance assessment;
 - **metadata and representations of data** for communication by both humans and machines, and the definition of the corresponding interfaces, such as **data interchange formats**;
 - **modeling languages** for expressing specifications of systems and their integration and distribution, and rules for relating different specifications;
 - **functions** to support distributed system operation;

WG19 Terms of Reference (2)

- WG19 ensures the evolution of the standards portfolio by:
 - developing standards and technical reports;
 - facilitating the processing of PAS and Fast-Track documents in its areas of work;
 - providing the focal point for collaborative work with OMG and ITU-T on its areas of work, and with other organizations if required (e.g. IEEE).

WG19 DNA – The Family Tree



WG19 project list (active)

Product	Title	Category	Status	Editor
ISO/IEC FCD 15909-2	Software and system engineering -- High-level Petri nets -- Part 2: Transfer Format	Metadata Interchange	FDIS Ballot	Ekkart Kindler, Germany
ISO/IEC 15909-1:2004/FPDAmd 1	Software and system engineering -- High-level Petri nets -- Part 1: Concepts, definitions and graphical notation	Modelling and Specification	FPDAM Ballot Editing Meeting	Jonathan Billington, Australia
ISO/IEC 24744:2007/CD Amd 1	Information Technology -- Software Engineering -- Metamodel for Development Methodologies Amd 1 Graphical Notation	SE Method Metamodel	FPDAM Ballot	Cesar Gonzales-Perez, Spain
ISO/IEC FPDAM 10746-2	Information technology -- Open Distributed Processing -- Reference Model: Foundations - Amendment 1	IT/SE Frameworks	FPDAM Ballot	Antonio Vallecillo, Spain
ISO/IEC FPDAM 10746-3	Information technology -- Open Distributed Processing -- Reference Model: Architecture - Amendment 1	IT/SE Frameworks	FPDAM Ballot	Bryan Wood, UK
ISO/IEC DCOR 19793	Defect Report for Enterprise Language	Modelling and Specification	DCOR Ballot	Jean Bérubé, Canada
ISO/IEC 19793	Study Period on the expression of policies and obligations	Modelling and Specification	Study period -> NWi	Peter Linington, UK
ISO/IEC 15909-3	Study Period on High-level Petri Nets - Part 3: Extensions	Modelling and Specification	Study Period -> NWI	Laure Petrucci, France
ISO/IEC 15909-1	Study Period on the revision of High-level Petri Nets	Modelling and Specification	Study Period -> NWI	Jonathan Billington, Australia

WG19 project list (active PAS)

Product	Title	Category	Status	Editor
ISO/IEC DIS 19500-3	Information technology -- Architecture-Driven Modernization -- Knowledge Discovery Meta-model (KDM), v1.1	SE Data Metamodel	DIS Ballot	OMG
ISO/IEC DIS 19505-1	OMG Unified Modeling Language (UML2.1.2) - Part 1 Infrastructure	Modelling and Specification	DIS Ballot	OMG
ISO/IEC DIS 19505-2	OMG Unified Modeling Language (UML2.1.2) - Part 2 Superstructure	Modelling and Specification	DIS Ballot	OMG
ISO/IEC DIS 19500-1	Information technology -- Open Distributed Processing -- Common Object Request Broker Architecture (CORBA) specification -- Part 1: CORBA interfaces	ODP Services	DIS Ballot	OMG
ISO/IEC DIS 19500-2	Information technology -- Open Distributed Processing -- Common Object Request Broker Architecture (CORBA) specification -- Part 2: Interoperability	ODP Services	DIS Ballot	OMG
ISO/IEC DIS 19500-3	Information technology -- Open Distributed Processing -- Common Object Request Broker Architecture (CORBA) specification -- Part 3: Components	ODP Services	DIS Ballot	OMG

WG19 products (completed 04-09)

Product	Title	Category	Pub.Y
ISO/IEC 19793:2008	Information technology -- Open Distributed Processing -- Use of UML for ODP system specifications	Modelling and Specification	2008
ISO/IEC 24744:2007	Information Technology -- Software Engineering -- Metamodel for Development Methodologies	SE Method Metamodel	2007
ISO/IEC 15476-6:2006	Information technology -- CDIF semantic metamodel -- Part 6: State/event models	SE Data Metamodel	2006
ISO/IEC 15476-3:2006	Information technology -- CDIF semantic metamodel -- Part 3: Data definitions	SE Data Metamodel	2006
ISO/IEC 13235-3:1998/Cor 1:2006	Information technology -- Open Distributed Processing -- Trading Function -- Part 3: Provision of Trading Function using OSI Directory service	ODP Services	2006
ISO/IEC 15414:2006	Information technology -- Open distributed processing -- Reference model -- Enterprise language	Modelling and Specification	2006
ISO/IEC 15476-4:2005	Information technology -- CDIF semantic metamodel -- Part 4: Data models	SE Data Metamodel	2005
ISO/IEC DTR 15476-5	Information technology -- CDIF semantic metamodels -- Part 5: Data flow models	SE Data Metamodel	Not Pub
ISO/IEC 19501:2005	Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2	Modelling and Specification	2005
ISO/IEC 15909-1:2004	Software and system engineering -- High-level Petri nets -- Part 1: Concepts, definitions and graphical notation	Modelling and Specification	2004

WG19 products (completed 00-03)

Product	Title	Category	Pub.Y
ISO/IEC 19500-2:2003	Information technology -- Open Distributed Processing -- Part 2: General Inter-ORB Protocol (GIOP)/Internet Inter-ORB Protocol (IIOP)	ODP Services	2003
ISO/IEC 15474-1:2002	Information technology -- CDIF framework -- Part 1: Overview	IT/SE Frameworks	2002
ISO/IEC 15474-2:2002	Information technology -- CDIF framework -- Part 2: Modelling and extensibility	Modelling and Specification	2002
ISO/IEC 15475-1:2002	Information technology -- CDIF transfer format -- Part 1: General rules for syntaxes and encodings	Metadata Interchange	2002
ISO/IEC 15475-2:2002	Information technology -- CDIF transfer format -- Part 2: Syntax SYNTAX.1	Metadata Interchange	2002
ISO/IEC 15475-3:2002	Information technology -- CDIF transfer format -- Part 3: Encoding ENCODING.1	Metadata Interchange	2002
ISO/IEC 15476-1:2002	Information technology -- CDIF semantic metamodel -- Part 1: Foundation	SE Data Metamodel	2002
ISO/IEC 15476-2:2002	Information technology -- CDIF semantic metamodel -- Part 2: Common	SE Data Metamodel	2002
ISO/IEC 14769:2001	Information technology -- Open Distributed Processing -- Type Repository Function	ODP Services	2001
ISO/IEC 15437:2001	Information technology -- Enhancements to LOTOS (E-LOTOS)	Modelling and Specification	2001
O/IEC 10746-4:1998/Amd 1:2001	Computational formalization	Modelling and Specification	2001
ISO/IEC 14752:2000	Information technology -- Open Distributed Processing -- Protocol support for computational interactions	ODP Services	2000

WG19 products (completed 95-00)

Product	Title	Category	Pub. Y
<u>ISO/IEC 14750:1999</u>	Information technology -- Open Distributed Processing -- Interface references and binding	ODP Services	1999
<u>ISO/IEC 14753:1999</u>	Information technology -- Open Distributed Processing -- Naming framework	ODP Services	1999
<u>ISO/IEC 14771:1999</u>	Information technology -- Open Distributed Processing -- Trading Function -- Part 3: Provision of Trading Function using OSI Directory service	ODP Services	1998
<u>ISO/IEC 10746-1:1998</u>	Information technology -- Open Distributed Processing -- Reference model: Overview	IT/SE Frameworks	1998
<u>ISO/IEC 10746-4:1998</u>	Information technology -- Open Distributed Processing -- Reference Model: Architectural semantics	IT/SE Frameworks	1998
<u>ISO/IEC 13235-1:1998</u>	Information technology -- Open Distributed Processing -- Trading function: Specification	ODP Services	1998
<u>ISO/IEC 13235-3:1998</u>	Information technology -- Open Distributed Processing -- Trading Function -- Part 3: Provision of Trading Function using OSI Directory service	ODP Services	1998
<u>ISO/IEC 14568:1997</u>	Information technology -- DXL: Diagram eXchange Language for tree-structured charts	Metadata Interchange	1997
<u>ISO/IEC 10746-2:1996</u>	Information technology -- Open Distributed Processing -- Reference Model: Foundations	IT/SE Frameworks	1996
<u>ISO/IEC 14750:2000</u>	Information technology -- Open Distributed Processing -- Reference Model: Architecture	IT/SE Frameworks	1996
<u>ISO/IEC 14753:2000</u>	Information technology -- Representation for human communication of state transition of software	Modelling & Specification	1995

WG19 products (completed -95)

Product	Title	Category	Pub. Y
<u>ISO/IEC 8631:1989</u>	Information technology -- Program constructs and conventions for their representation	Modelling and Specification	1989
<u>ISO 8807:1989</u>	Information processing systems -- Open Systems Interconnection -- LOTOS -- A formal description technique based on the temporal ordering of observational behavior	Modelling and Specification	1989
<u>ISO 8790:1987</u>	Information processing systems -- Computer system configuration diagram symbols and conventions	Modelling and Specification	1987
<u>ISO 5807:1985</u>	Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts	Modelling and Specification	1985
<u>ISO 5806:1984</u>	Information processing -- Specification of single-hit decision tables	Modelling and Specification	1984
<u>ISO 3535:1977</u>	Forms design sheet and layout chart	Modelling and Specification	1977

The Reference Model of Open Distributed Processing (RM-ODP)

ISO/IEC 10746, ITU-T X.901-4

www.rm-odp.net

ODP system specifications

- The Reference Model of ODP (ITU-T Rec X.901-904 | ISO/IEC 10746) defines a framework for system specification
- It covers all aspects of a distributed system - “enterprise”, information, functionality, infrastructure, technology

The RM-ODP

- Family of ISO/IEC Standards & ITU-T Recommendations
- Developed initially as reference standards for developing standards for open distributed systems
- Better to consider now as vendor neutral distributed system description framework
- Object Oriented
- Distinguishing feature: Five Standard Viewpoints defined for any system description

RM-ODP provides

- a structure for system specifications in terms of **viewpoints** on a system
- a **language** (concepts and rules) for expressing each viewpoint specification
- a set of object-oriented **foundation modelling concepts** common to all viewpoint languages
- A set of **correspondences** between the viewpoints
- Sets of **common functions, transparencies and conformance points**
- A **framework** for ODP standards

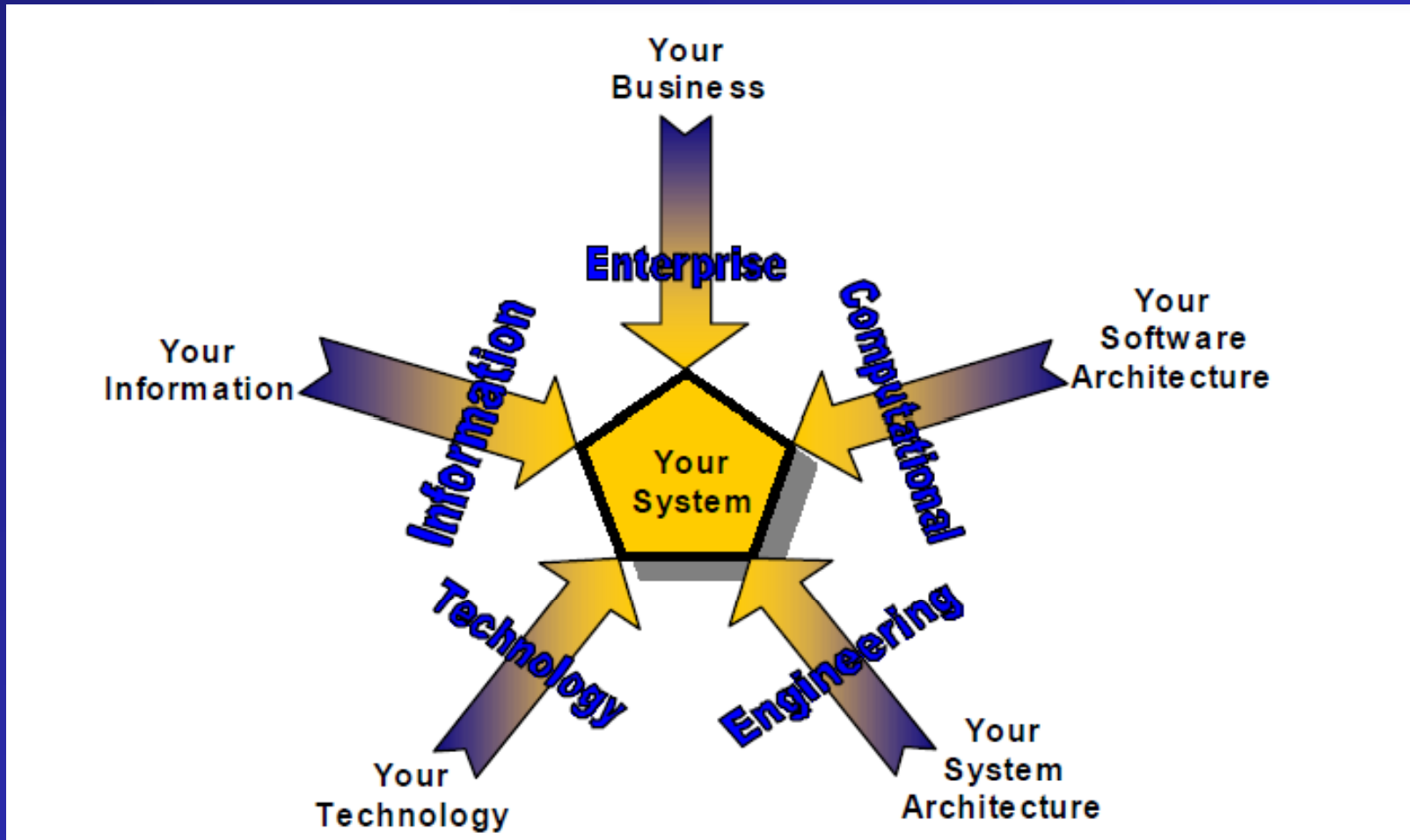
ODP viewpoints

- 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

ODP viewpoint specification

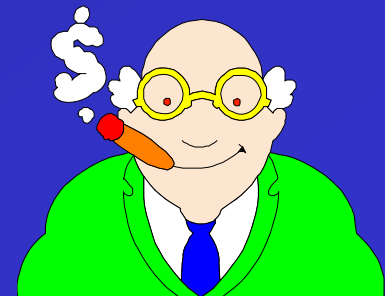
- Specification of a system from a specific viewpoint
- Expressed in terms of the viewpoint concepts and rules (the viewpoint language)
- Includes defined correspondences with other ODP viewpoint specifications

ODP Viewpoints



The Enterprise specification

- Specifies the roles played by a system in its organisational environment
- An object model of, for example, part of some social/commercial organisation in terms of:
 - *enterprise objects*
 - *communities (of enterprise objects)*
 - *objectives*
 - *behaviour*
 - *roles (of enterprise objects in a community)*
 - *processes*
 - *policies*
 - ...



Example: Bank information system

- A bank is composed of branches, spread all over the country
- The bank's central office manages and coordinates the branches' activities
- Each branch has a manager and is responsible to provide banking services to its customers
- Branches may interact with each other and with the bank central office
- Each branch will have an ATM and a main server, and each branch's employee will have a computer and a printer
- The Bank information system (BIS) will manage all IS-related issues

BIS – Enterprise Specification (1)

- Each branch, and will be specified by a community
 - Its **goal** is to “provide banking services to its customers”
 - Its **objects** model the branch entities: people (“Joe Smith”, “Lucy Brown”), computers (PC #123-45, printer #xyz), bank accounts, etc.
 - Its **roles** are: branch manager, controller, customer (active),..., or bank account, money, etc. (passive)
 - **Assignment policies** (e.g., the requirements of a person to become a customer)
 - **Policies:**
 - **Permissions:** what can be done, e.g. money can be deposited into an open account
 - **Prohibition:** what must not be done, e.g. customers must not withdraw more than 600 Euros per day
 - **Obligations:** what must be done, e.g. the bank manager must advise customers when the interest rate changes, customers must present some ID for withdrawing money.
 - **Authorizations:** accounts of some VIP customers are allowed to have overdrawn.

BIS – Enterprise Specification (2)

- **Environment contracts**: e.g., transactions performed using other banks' ATMs should have effect within at most 24 hours; information about a branch's customers cannot be disclosed to other branches
- **Accountability**: e.g., the branch manager is responsible for authorizing an overdrawn, but can delegate to the branch's controller officer
- The bank's central office will be specified by another community
 - It's **goal** is to “manage and coordinate the branches' activities”
 - It's **objects** are...
 - It's **roles** are ...
 - It's **assignment policies** are...
 - It's **policies** are...
 - **Environment contracts**...
 - **Accountability**....
- Branches interact with each other and with the central office...

The Information specification

- Specifies system behaviour to fulfil its enterprise roles, 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*



BIS – Information specification

- Describes a model with the information types, their relationships, and constraints on these types and relationships
 - e.g., a bank account consists a balance and the “amount-withdrawn-today”.
- A **Static schema** captures the state and structure of a object at some particular instance
 - e.g., at midnight, the amount-withdrawn-today is 0.
- An **invariant schema** restricts the state and structure of an object at all times
 - e.g., the amount-withdrawn-today is less than or equal to 600.
- A **dynamic schema** defines a permitted change in the state and structure of an object
 - e.g. a withdrawal of \$X from an account decreases the balance by \$X and increases the amount-withdrawn-today by \$X.
- Static and dynamic schema are always constrained by invariant schemata
 - \$400 could be withdrawn in the morning but additional \$200 cannot be withdrawn in the afternoon as the amount-withdrawn-today cannot exceed \$500
- Schemas can also be used to describe relationships or associations between objects
 - e.g., static schema ownsAccount could associate each account with a customer.

The computational specification

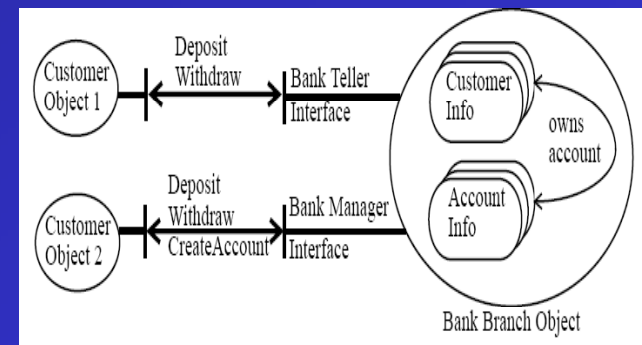
- Specifies computational structure of the system in terms of units of distribution and portability and the interactions between them
- An object model of the *system* describing the structure of processing in terms of:
 - *computational objects*
 - *Interfaces (of computational objects):* identifying functions supported
 - *Invocations (by computational objects):* identifying functions invoked
 - *activities:* sequences of invocations
 - *computational bindings:* *QoS* constraints on *invocations*



BIS – Computational spec (1)

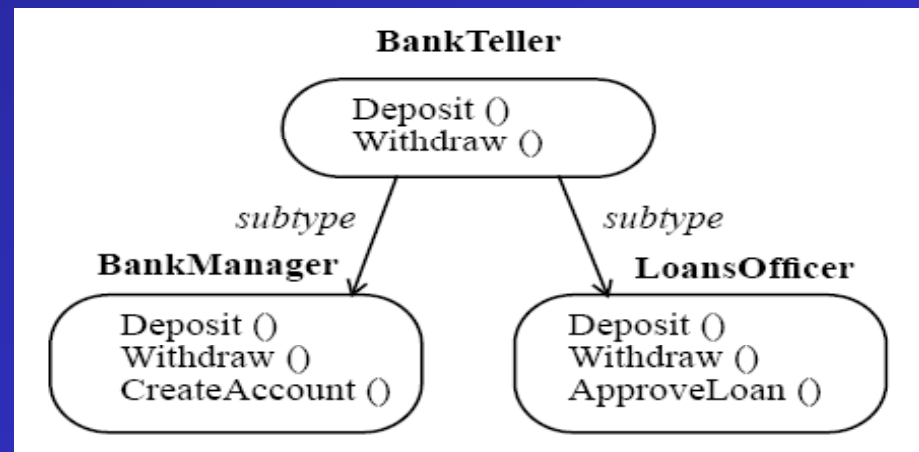
- Objects in a computational specification can be application objects (e.g. a bank branch) or ODP infrastructure objects (e.g. a type repository or a trader)
- Objects interact at well defined interfaces, using signals, operations or flows.

```
BankTeller = Interface Type {  
  operation Deposit (c: Customer, a: Account, d: Dollars)  
    returns OK (new_balance: Dollars)  
    returns Error (reason: Text);  
  operation Withdraw (c: Customer, a: Account, d: Dollars)  
    returns OK (new_balance: Dollars)  
    returns NotToday (today: Dollars, daily_limit: Dollars)  
    returns Error (reason: Text);  
}
```



BIS – Computational spec (2)

- Interfaces allow subtyping
- Environment contracts capture non functional requirements
 - Security,
 - performance,
 - availability,
 - etc.

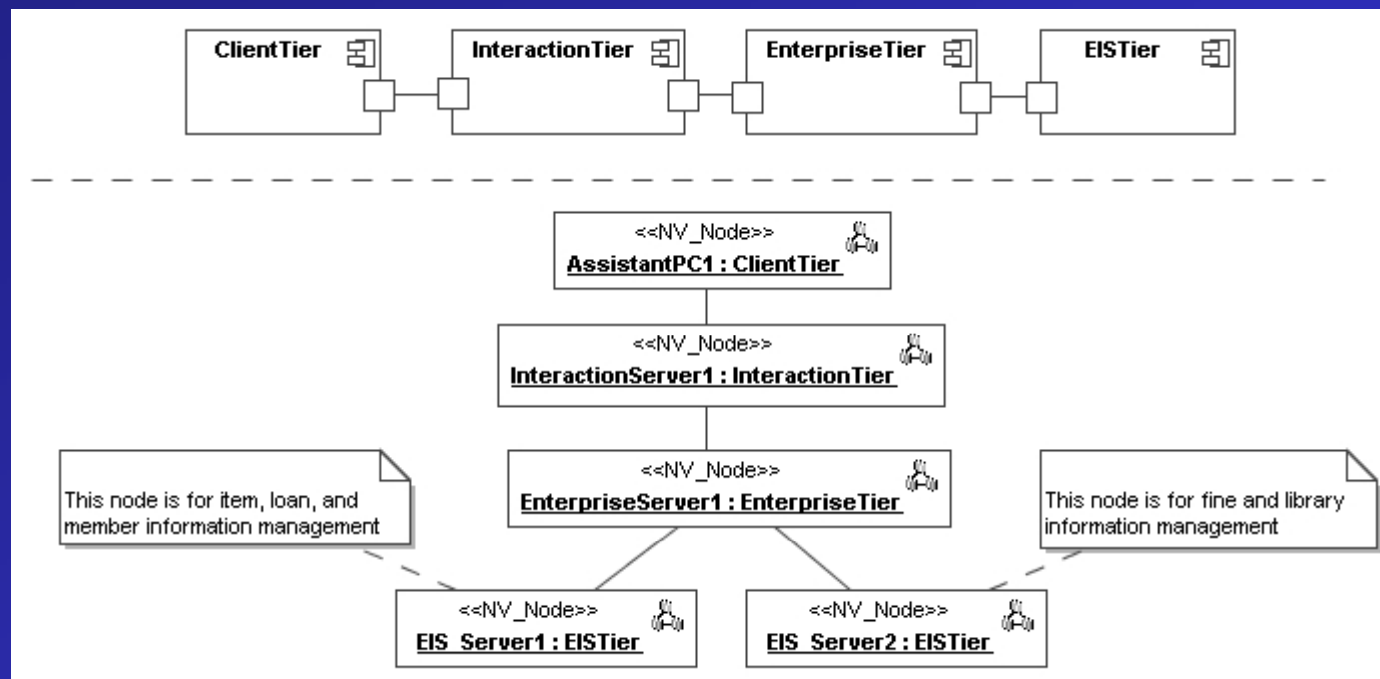


The engineering specification

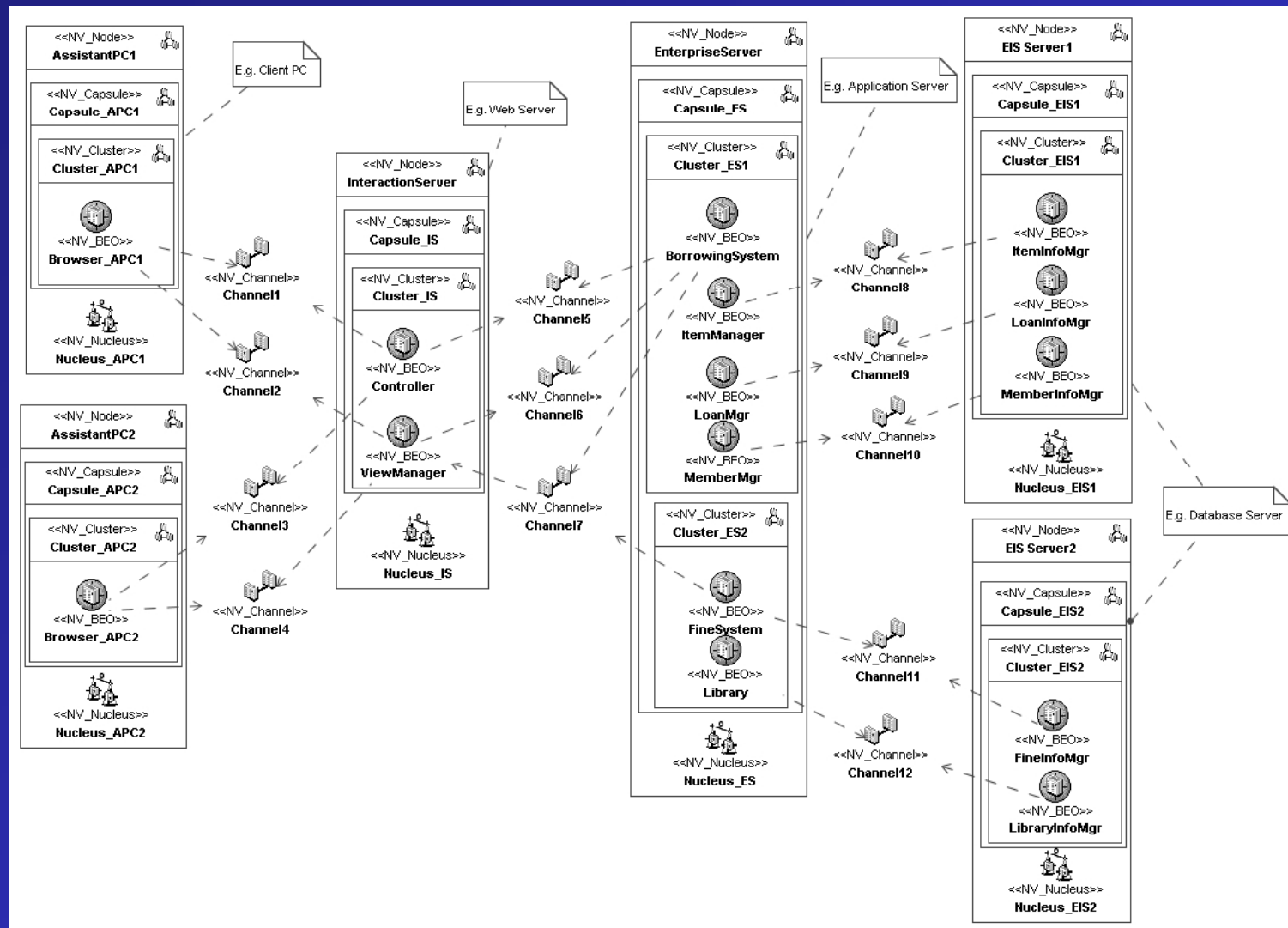
- Specifies the mechanisms and services that provide the distribution transparencies and QoS constraints required by the system independent of platform
- An object model of the system describing the infrastructure supporting the computational structure
 - *basic engineering objects*
 - *(infrastructure) engineering objects*
 - *clusters, capsules, nodes*
 - *channels*
 - *functions*



BIS – Engineering specification (1)



BIS – Engineering specification (2)



The technology specification

- Specifies the H/W and S/W 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*



BIS – Technology specification

- Technology object types
 - Types of PCs, servers, ATMs, printers
 - Types of Operating Systems and Applications (text editors, etc)
 - Types of connections (LANs, WANs, Intranets, etc.)
- Technology selection process
 - Providers' selection and contracts
 - Conformance points
 - Compliance tests
- Implementation, deployment, maintenance, evolution
 - Deployment plans
 - Configuration guides
 - Evolution plans

Correspondences, Common Functions and Transparencies

- **Correspondences**
 - An ODP specification of a system is composed of five views and a set of correspondences between them
 - Correspondences do not belong to any view
 - ODP distinguishes two kinds of correspondences
 - Required correspondences
 - Correspondence statements
- **Common functions**
 - An ODP specification can make use of some of the common functions defined by the RM-ODP. They are “standard”
- **Transparencies**
 - An ODP specification can implement some of the transparencies defined by the RM-ODP
 - The specification should state which ones are used, and how they are implemented

(Some) Sources

- COMBINE and Synapses EU-funded projects
- Reference Architecture for Space Data Systems (RASDS), NASA/JPL, US
- Interoperability Technology Association for Information Processing (INTAP), Japan
- Japanese Association of Healthcare Information System Industry (JAHSI) - Hospital Information Reference Enterprise Model project , Japan
- NEHTA Interoperability Framework (National E-Health Transition Authority), Australia
- DASIBAO reference architecture, EDF, France

ODP standards from SC7

Notation and Architectural Frameworks:

- ISO/IEC 14750 ODP – Interface Definition Language
- ISO/IEC 14771 ODP – Naming framework
- ISO/IEC 14753 ODP – Interface references and binding
- ISO/IEC 14752 ODP – Protocol support for comp. interactions
- ISO/IEC 15414 ODP – Enterprise Language
- ISO/IEC 19793 ODP – Use of UML for ODP system specs

Components:

- ISO/IEC 13235 ODP – Trading Function
- ISO/IEC 14769 ODP – Type repository

ODP standards from SC7

Notation and Architectural Frameworks:

- ISO/IEC 14750 ODP – Interface Definition Language
- ISO/IEC 14771 ODP – Naming framework
- ISO/IEC 14753 ODP – Interface references and binding
- ISO/IEC 14752 ODP – Protocol support for comp. interactions
- ISO/IEC 15414 ODP – Enterprise Language
- **ISO/IEC 19793 ODP – Use of UML for ODP system specs**

Components:

- ISO/IEC 13235 ODP – Trading Function
- ISO/IEC 14769 ODP – Type repository

OMG PAS submissions

- ISO/IEC 19500-2 ODP – Open Distributed Processing - General Inter-ORB Protocol (GIOP)/ Internet Inter-ORB Protocol (IIOP)
 - providing basic ODP protocol support for computational interactions
- ISO/IEC 19501 Information technology – Unified Modeling Language (UML)
 - providing notation for ODP specifications
- CORBA (Common Object Request Broker Architecture) Services
 - providing basic ODP functions
- ADM – Knowledge Discovery Metamodel
- ...

What is defined (and not defined) in the RM-ODP

- **Defined**
 - Vocabulary to define viewpoint specifications
 - Structuring rules
 - Set of viewpoints, correspondences, transparencies
- **NOT defined**
 - Notation (i.e., concrete syntax) for viewpoint languages: Could be text or any language or technique like FDT, UML, etc.
 - Process or methodology for developing the specifications.

Use of UML for ODP system specifications **(UML4ODP)**

ISO/IEC 19793, ITU-T X.906

www.rm-odp.net

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

- A standard defining:
 - a set of UML profiles for expressing a system specification in terms of ODP 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

**currently Version 2.1.1*

UML4ODP

- Why?
 - RM-ODP is notation- and methodology-independent
 - This is an advantage (a-priori) ...
 - ...but in fact it hampers the widespread adoption and use of ODP
 - No notation...
 - No tool support...
 - No ODP-based methodologies...

Target audiences

- UML Modelers
 - who need to structure (somehow) their LARGE system specifications
- ODP Modelers
 - who need some (graphical) notation for expressing their ODP specifications and tool support
- Modeling tool suppliers
 - who wish to develop UML-based tools capable of expressing RM-ODP viewpoint specifications.

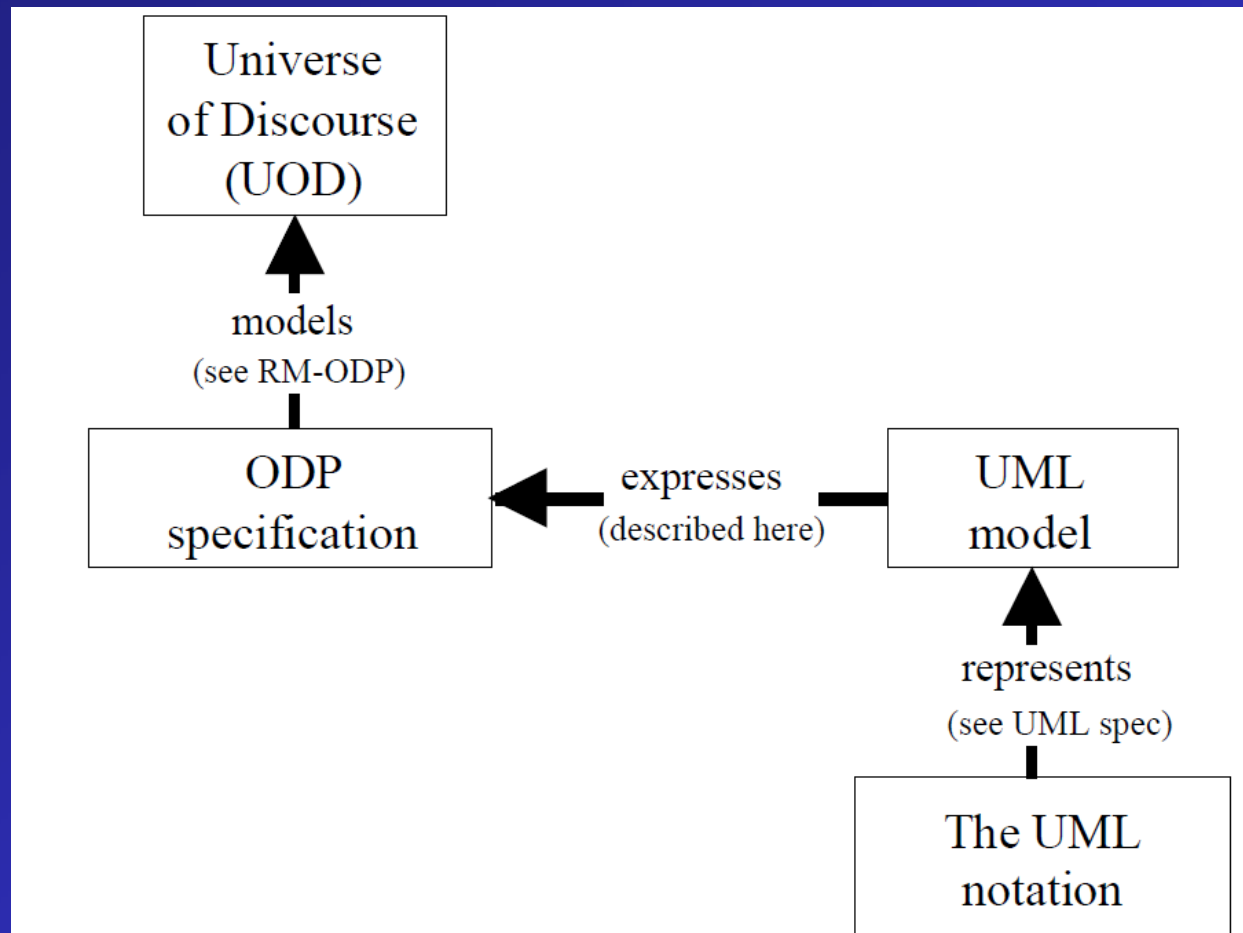
UML4ODP defines

- a UML based notation for the expression of ODP specifications
- an approach for structuring of them using the notation, thus providing the basis for model development methods

UML4ODP provides

- The expression of a system specification in terms of RM-ODP viewpoint specifications, using defined UML concepts and extensions
 - A set of **UML 2 profiles** (one for each viewpoint)
 - A way of using these profiles (**structuring rules**)
- Relationships between the resultant RM-ODP viewpoint specifications
 - A way of modelling ODP correspondences
 - A **UML profile for correspondences**
- A way for modelling conformance of implementations to specifications
 - A **profile for conformance** (reference points, conformance statements,...)

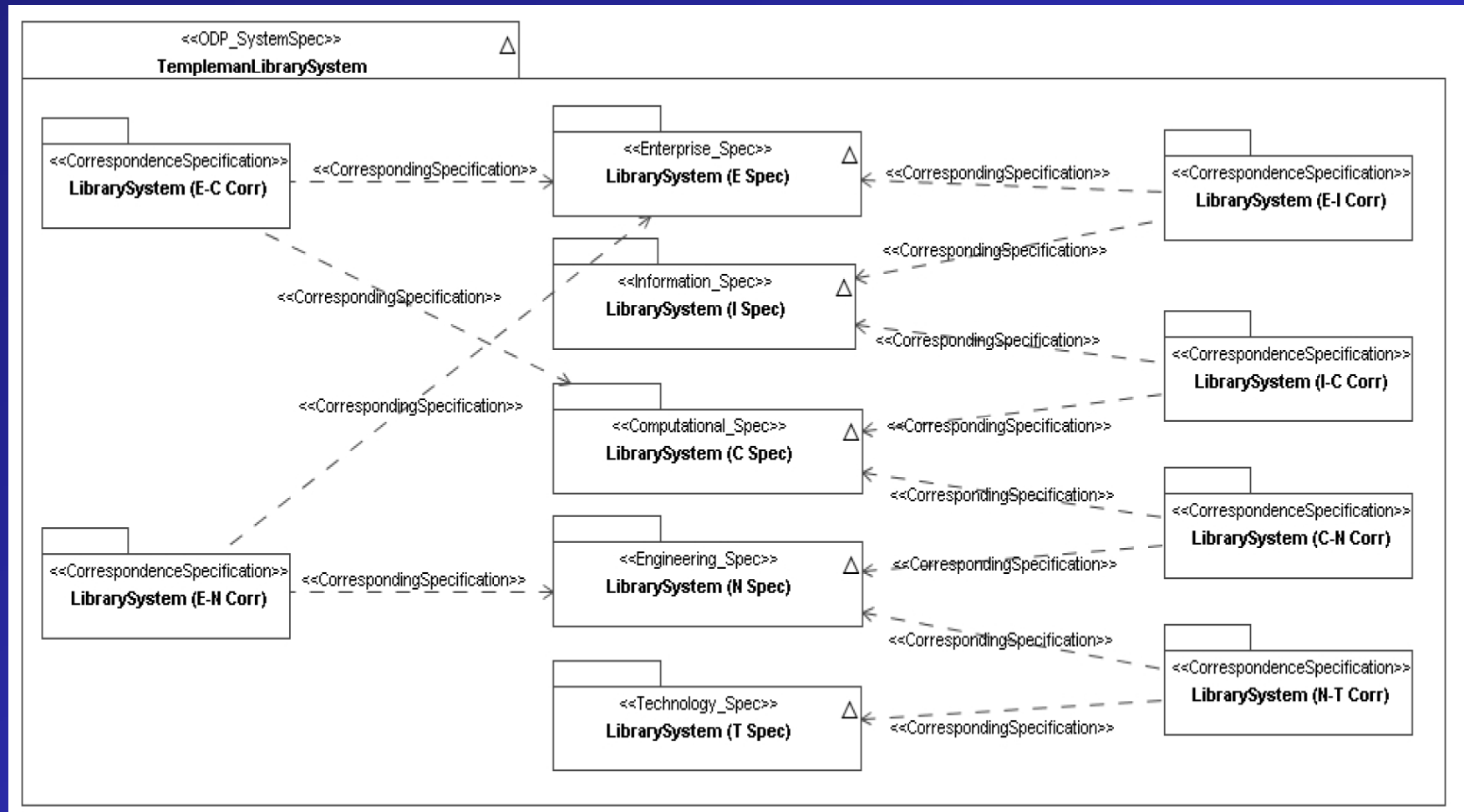
UML4ODP notation scope



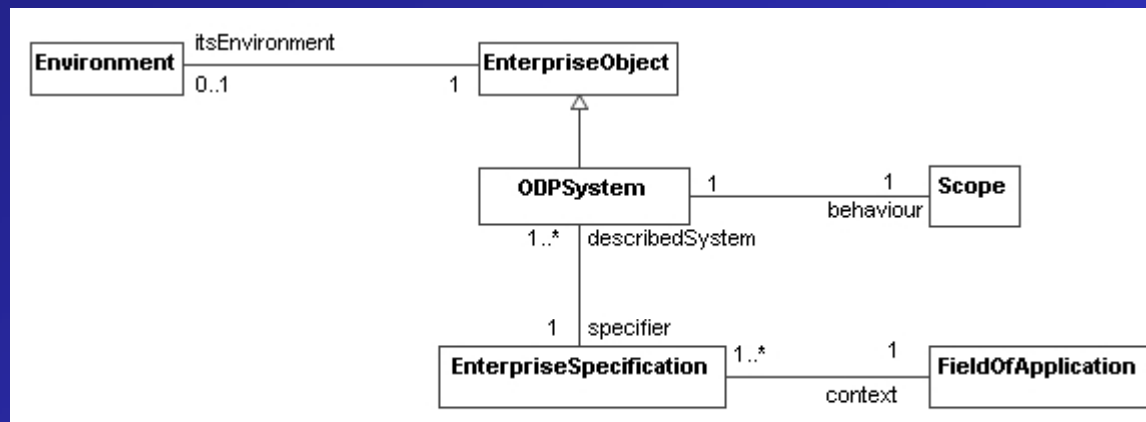
Viewpoint languages in UML

- The DSLs used to represent the viewpoint languages are defined using the UML **lightweight** extension mechanism (**UML Profiles**)
- Stereotypes are used to represent domain specific specializations of UML metaclasses in order to express the semantics of the RM-ODP viewpoint language concerned
- Each viewpoint specification uses the appropriate UML profile for that language, as described in Clauses 7 to 11

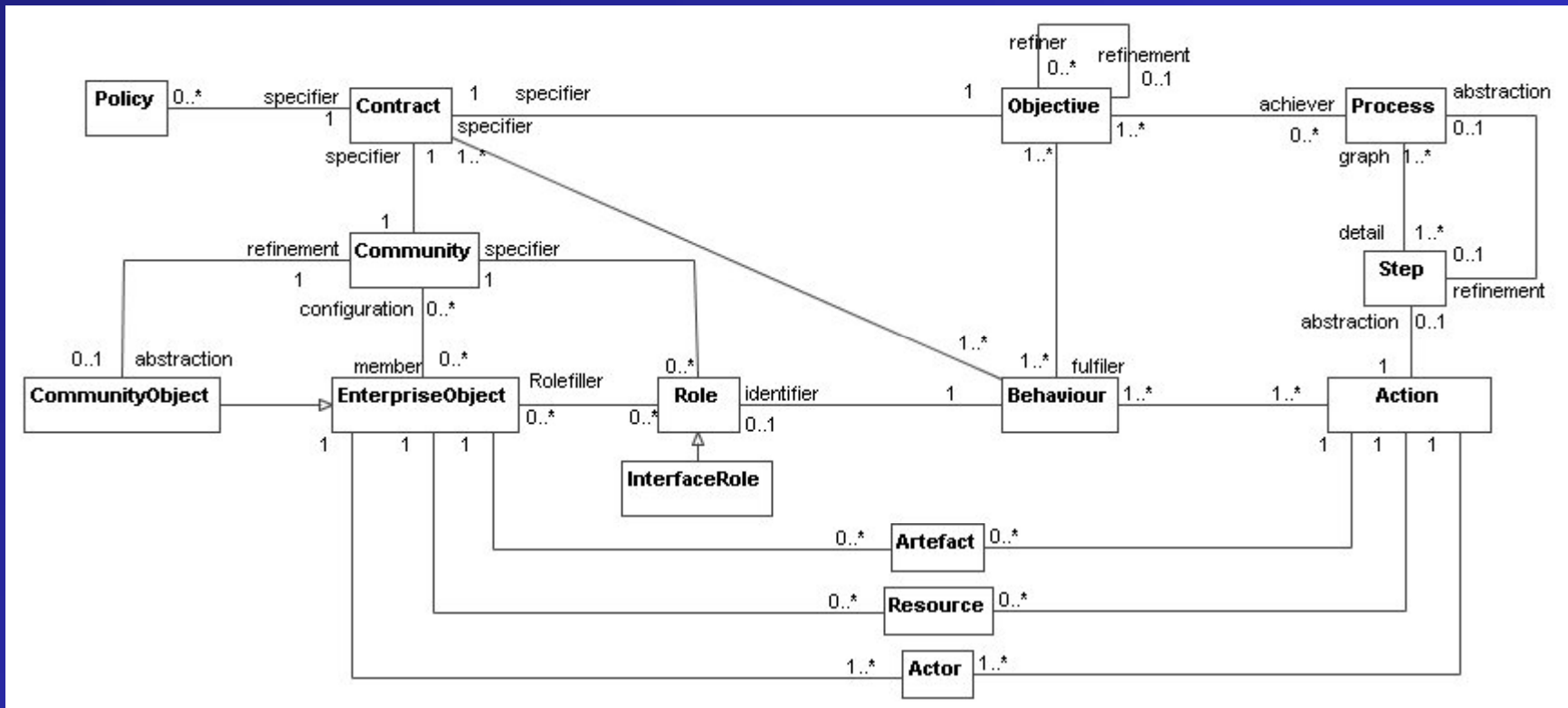
UML4ODP specification structure



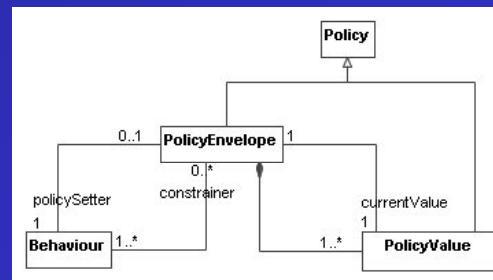
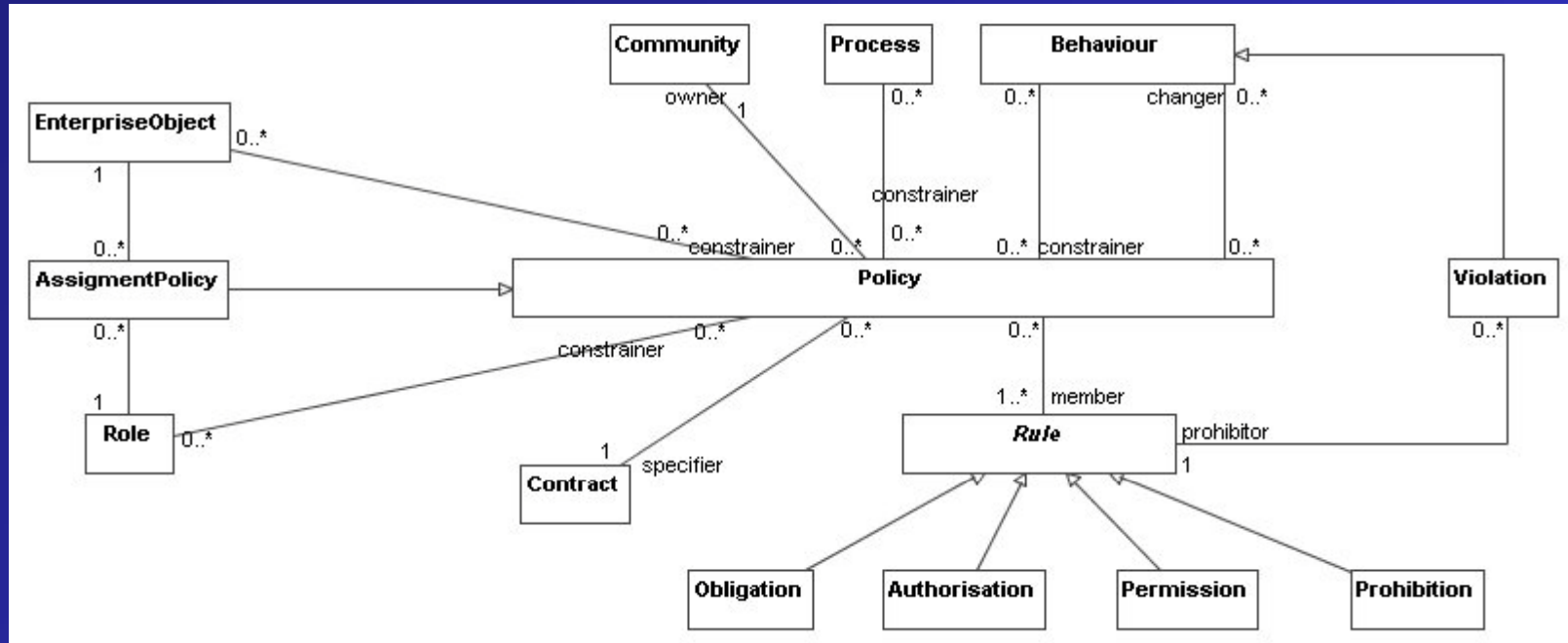
Enterprise concepts (1)



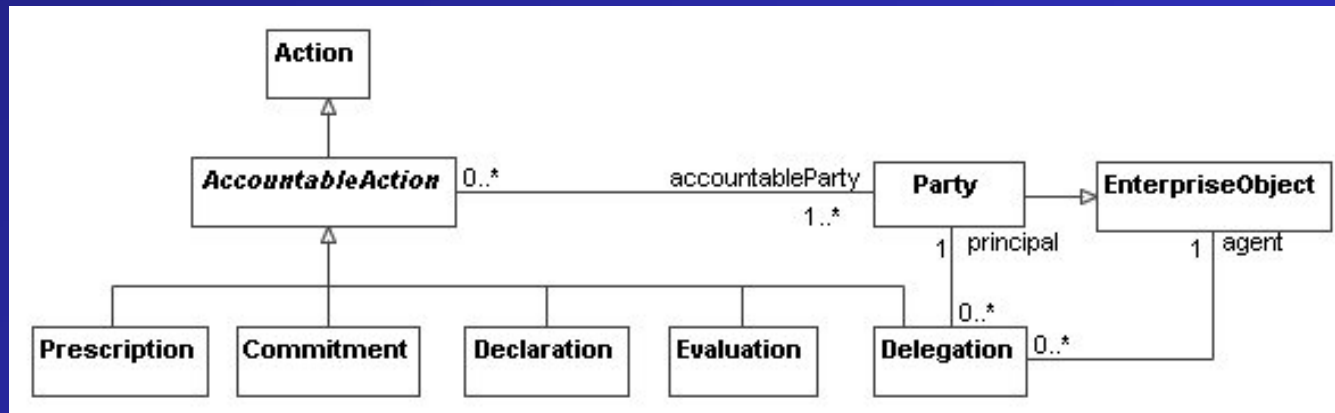
Enterprise concepts (2)



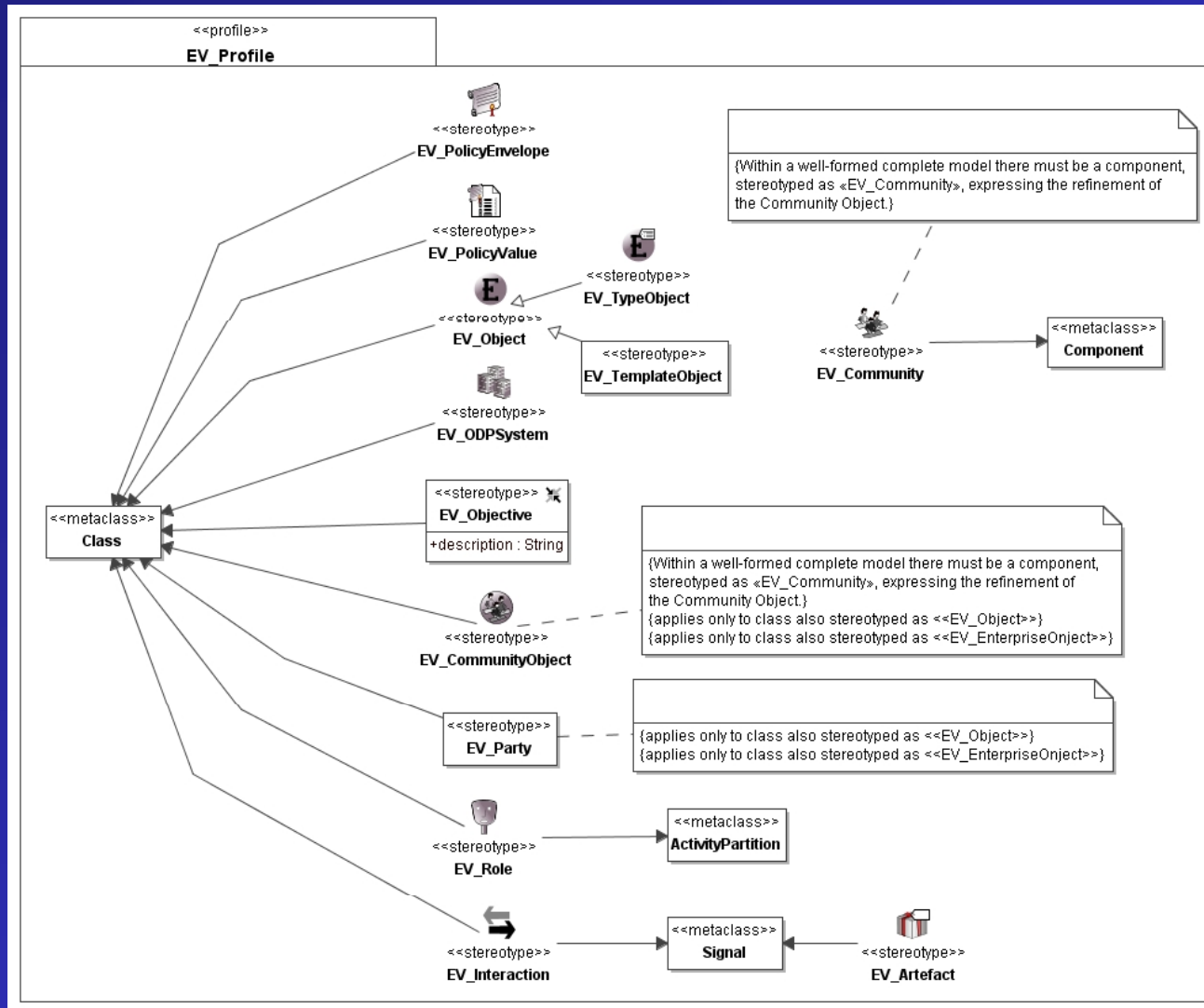
Enterprise concepts (3)



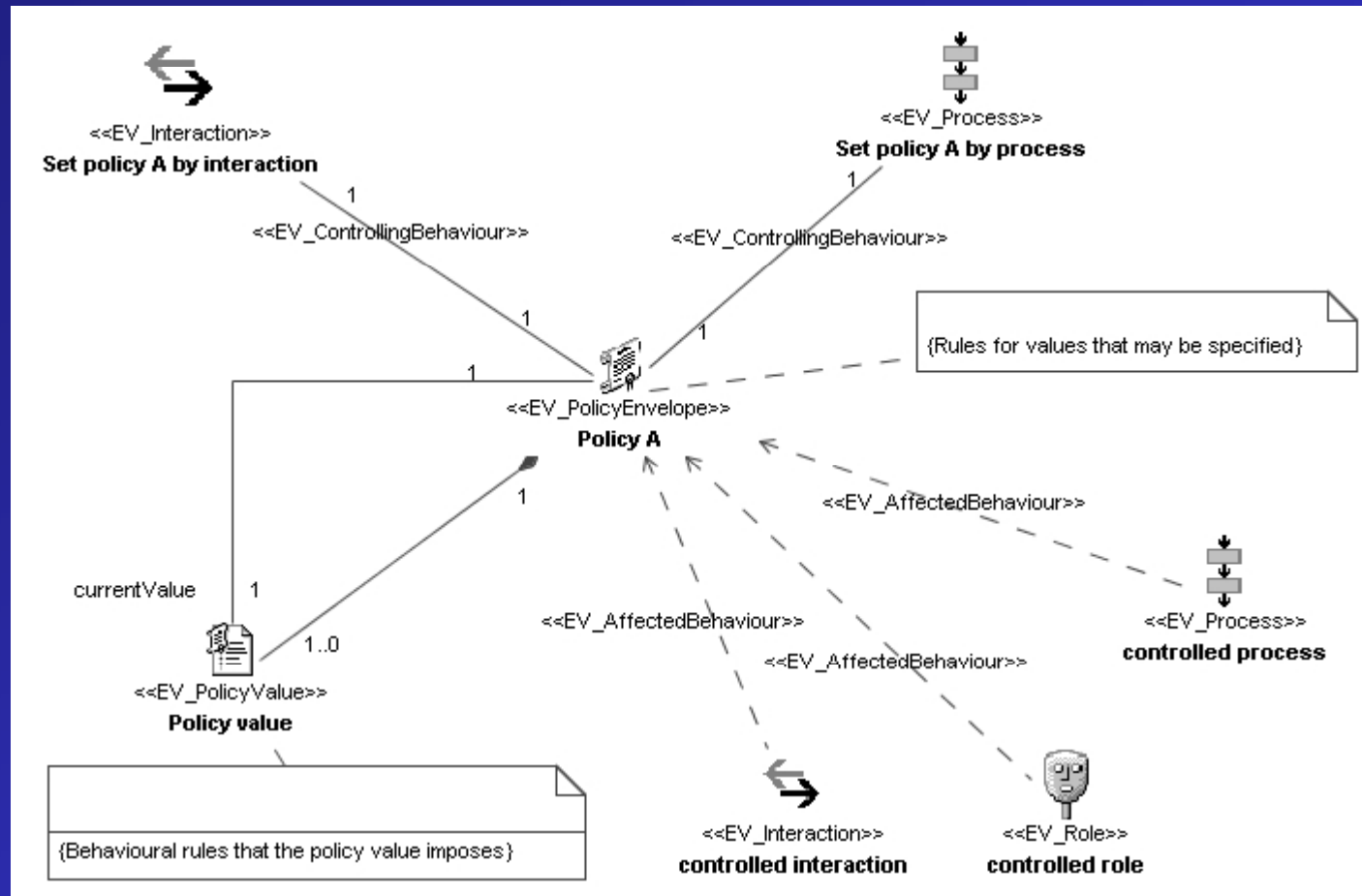
Enterprise concepts (4)



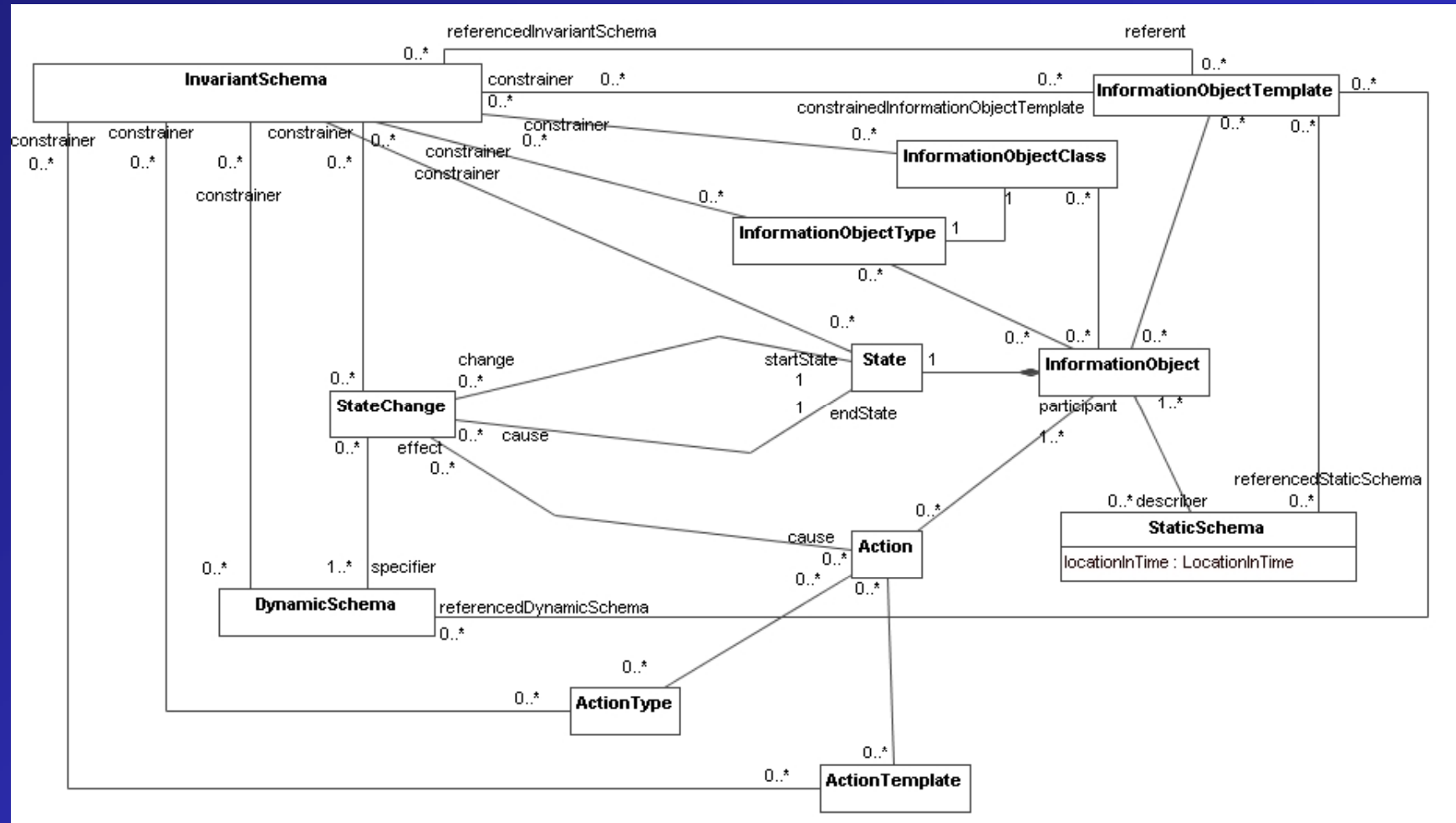
Enterprise Profile (excerpt)



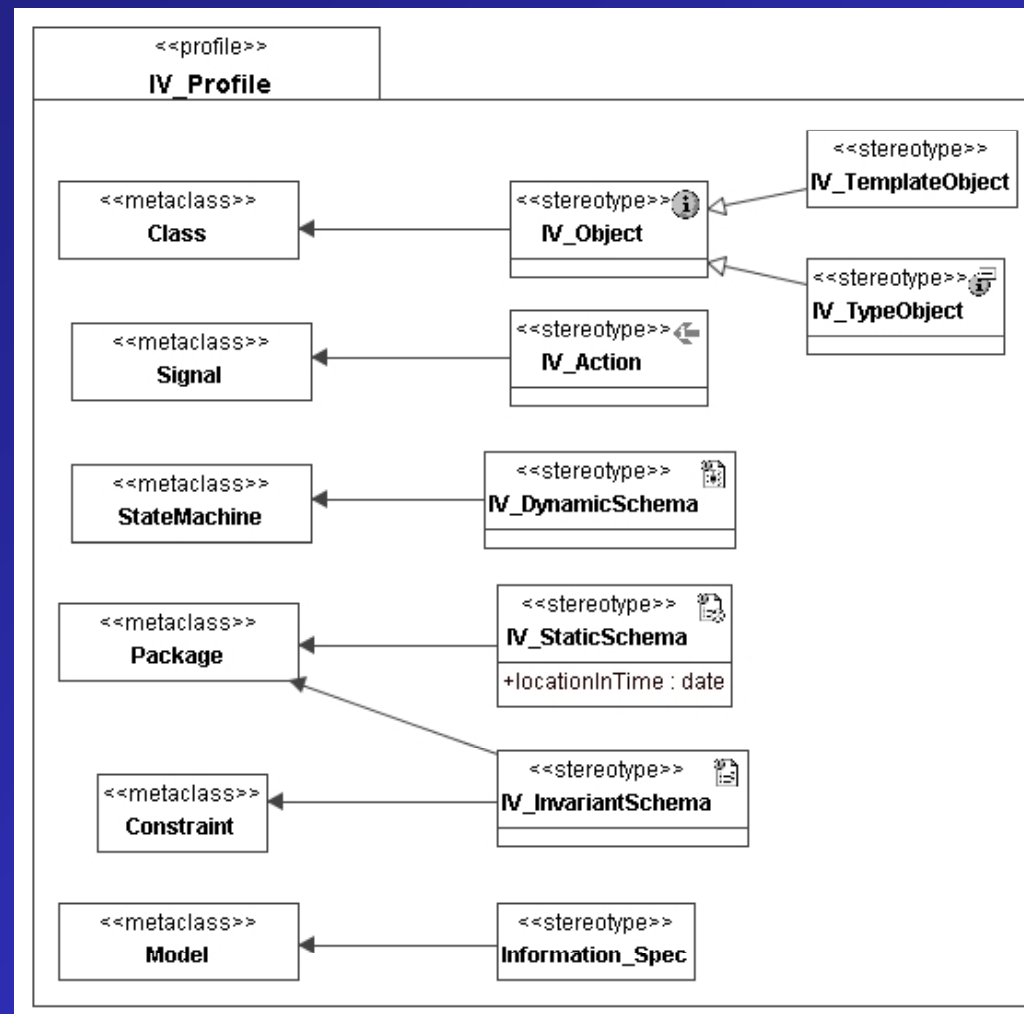
Pattern for policy concepts



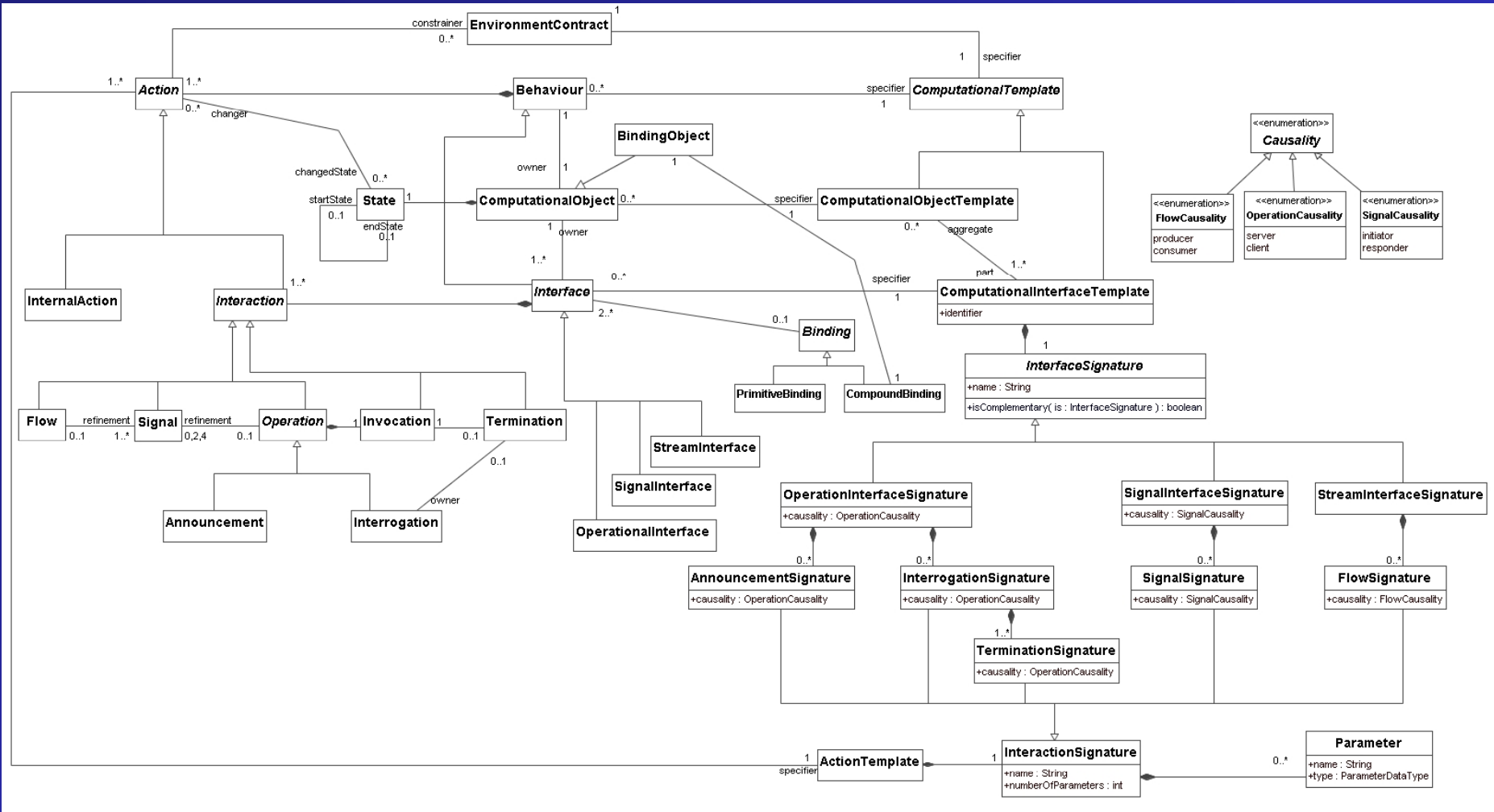
Information concepts



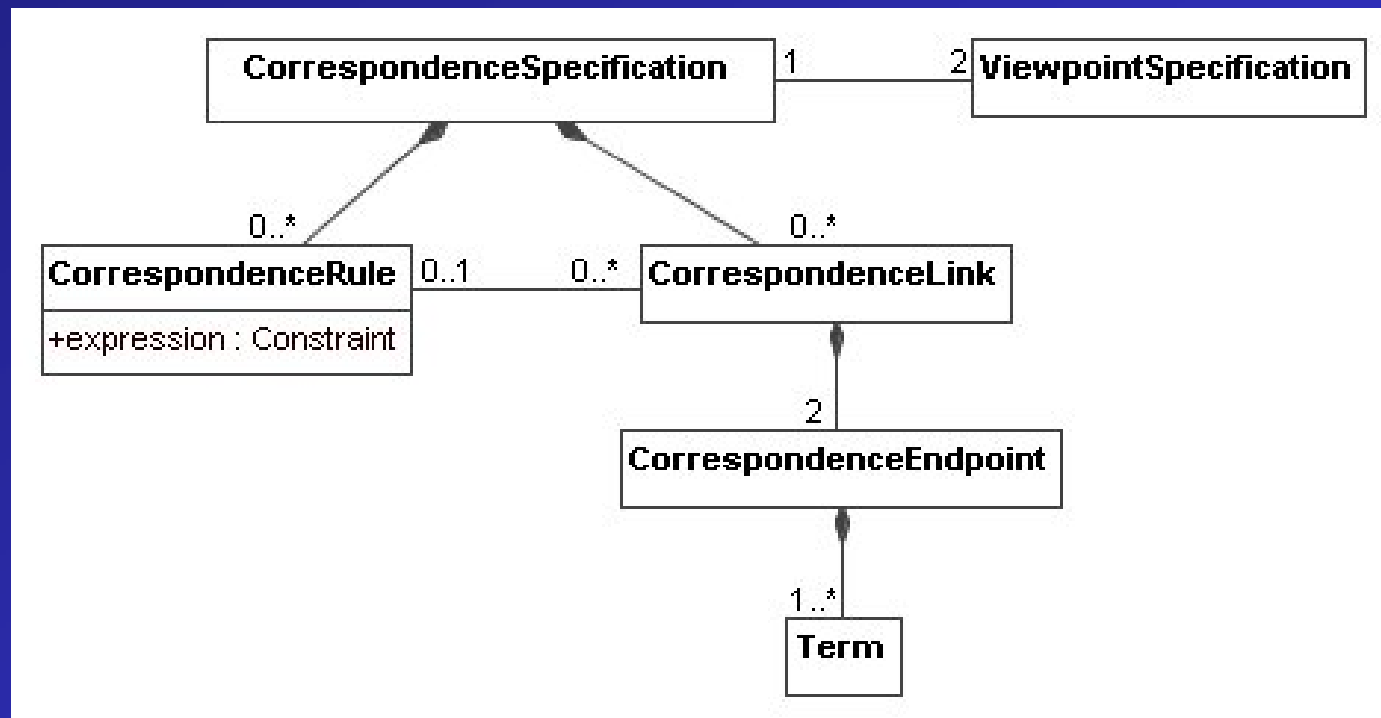
Information profile



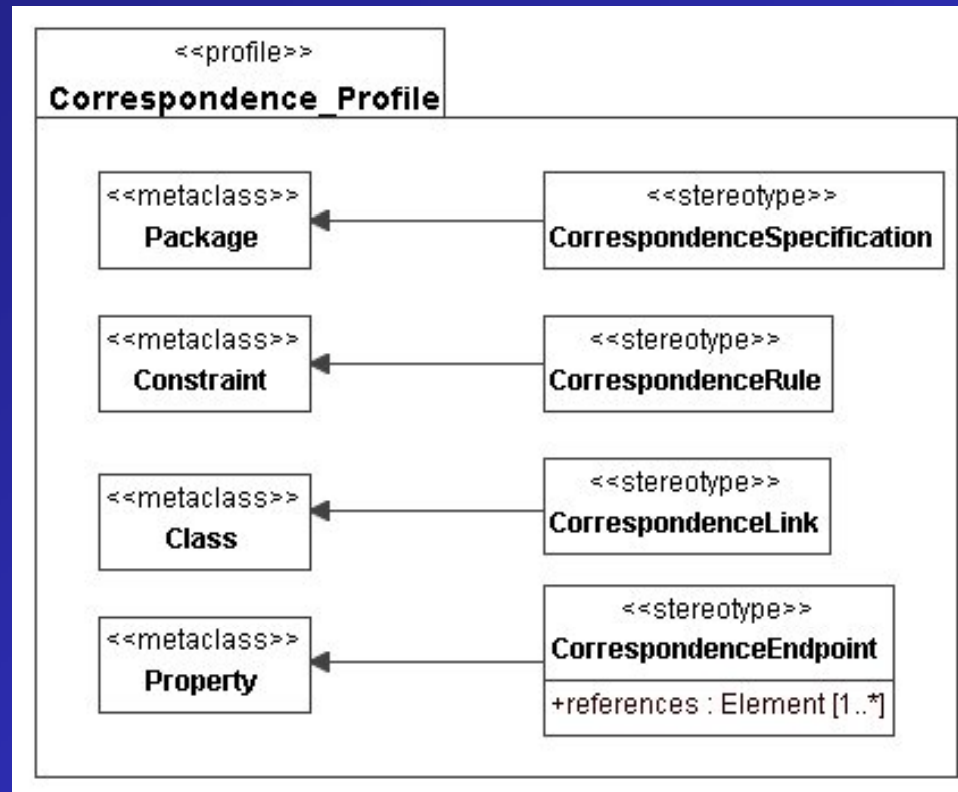
Computational concepts



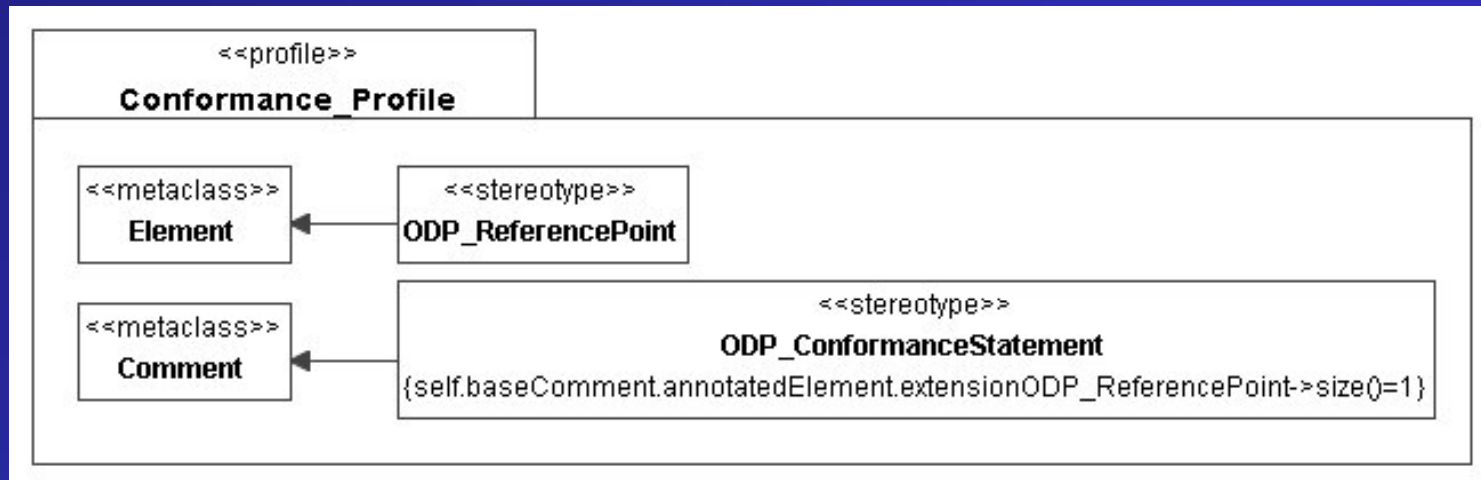
Correspondence metamodel



Correspondence profile



Conformance profile



UML4ODP document structure

- 1 Scope
- 2 Normative references
- 3 Definitions
- 4 Abbreviations
- 5 Conventions

- 6 Overview of modelling and system specification approach

- 7 Enterprise Specification
- 8 Information Specification
- 9 Computational Specification
- 10 Engineering Specification
- 11 Technology Specification

- 12 Correspondence Specification
- 13 Modelling conformance in ODP system specification

- 14 Conformance and compliance to this document

- Annex A An example of ODP specifications using UML

Document structure (clauses 7-11)

X <Viewpoint> Specification

X.1 Modelling concepts

- A brief description of the <viewpoint> language
- Summary of the <viewpoint> MOF-metamodel

X.2 UML Profile

- Description on how the language concepts are mapped to UML, by extending the appropriate metaclasses
- UML specification of the profile

X.3 <Viewpoint> specification structure (in UML terms)

- UML packages and grouping rules

X.4 Viewpoint correspondences for the <Viewpoint> language

- Description of the correspondences to other viewpoints

More on UML4ODP

- IS already published
- Tool support currently available
 - Profiles for several UML modeling tools
 - A plugin for MagicDraw
 - Writing ODP specifications using UML4ODP
 - Validation and conformance capabilities
- Visit www.rm-odp.net

Revision of the RM-ODP

Amendments to

ISO/IEC 10476-2 | ITU-T X.902 (Foundations)

ISO/IEC 10476-3 | ITU-T X.903 (Architecture)

www.rm-odp.net

Revision of RM-ODP

- Improvement/Explicit definition of some existing concepts
 - Role
 - Specification, Notation, Model, ...
 - Viewpoint correspondence
 - Policy concepts (rule, policy, policy envelope, ...)
- Introduction of new concepts
 - Service concepts (service, interoperability)
 - Component concepts (component, event, factory, ...)
 - Relationship/Relation
 - Patterns
- Improvement of alignments between Parts, and with other ODP standards

Policy concepts

11.2.7 Rule: A constraint on a system specification. Where appropriate, a rule can be expressed as an obligation, a permission or a prohibition.

NOTE – Rules may apply to the structure, behaviour or other properties of the system, including for example Quality of Service

11.2.8 Policy: A constraint on a system specification foreseen at design time, but whose detail is determined subsequent to the original design, and capable of being modified from time to time in order to manage the system in changing circumstances.

NOTES

1 Policies can be applied in any viewpoint; examples are an enterprise delegation policy, a computational persistence policy or an engineering scheduling or quality support policy.

2 The expectation of change is fundamental to the concept of policy, and a rule that does not envisage change is not a policy.

3 Policies may be expressed in terms of obligations, permissions or prohibitions, but this is not necessary for simple policies.

11.2.9 Policy Declaration: An element in a specification defined in order to allow incorporation of future constraints, together with rules determining the allowed form of acceptable constraints and the circumstances in which such constraints can be applied.

11.2.10 Policy Value: The specific constraints associated with a policy in some particular epoch.

11.2.11 Policy Envelope: The set of acceptable policy values that could be applied at a particular policy declaration. Restricting policy values to be within the policy envelope allows future flexibility but guarantees that the required properties of the system design will be preserved by all valid policies.

11.2.12 Policy Setting Behaviour: the behaviour defined in a specification via which a policy may be changed. A policy setting behaviour can be both an establishing behaviour (13.2.1) and a terminating behaviour (13.2.5).

Service (Part 2-13.3.1)

Service: a behaviour, triggered by an interaction, that adds value for the service users by creating, modifying, or consuming information; the changes become visible in the service provider's environment.

NOTES

1. Services are associated with interfaces and defined by the structural, behavioural and semantic rules of the interaction types involved.
2. A service can be characterized by a service type. A service is identifiable. A service may be composed of other services.
3. A service is in general invoked from within a liaison. Rules can be associated with the liaison, which refine the service for the duration of the liaison.
4. The service may be a complex behaviour, including both interactions and internal actions.
5. The provision of a service involves a collaboration between its provider and user. This collaboration may involve a complex series of interactions..

19793/19763 – Issues to consider

(to be completed)

- Overlaps of concepts (e.g., “service”)
 - In ODP, *basic concepts* span across viewpoints
 - Object, service, ...
 - They are interpreted in each viewpoint
 - Some ODP concepts are particular to individual viewpoints
 - Binding object, Cluster, IXIT (Implementation eXtra Information for Testing), ...
- Metamodels merging
 - Does it make sense?
- ...

Thanks!

Supporting Material

(Slides that could be of interest during the discussions)

- Maybe we can pass some of the previous slides to this part, in order to reduce the number of slides of the main presentation.
- The slides here can be shown if they are needed during the discussions
- E.g.,
 - Some of the viewpoint metamodels and profiles (NV, TV)?
 - The bank information system example?
 - The lists of projects and products on SC7