Open Distributed Processing Reference Model

Lea Kutvonen
Department of Computer Scienc
University of Helsinki

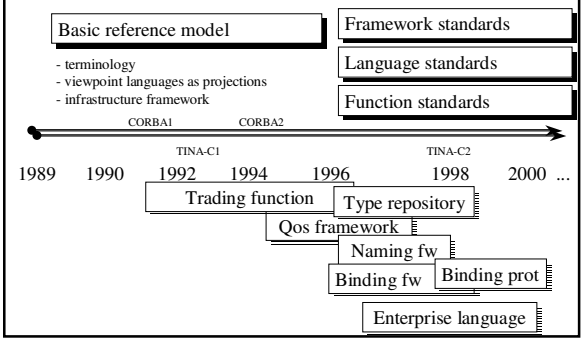
Outline

- Introduction
- Viewpoints
- System model: objects and infrastructure
- Services and functions, transparencies
- Usage
- Where to read more

Introduction -- RM-ODP Goals

- ODP mission in terms of research and industry needs
  - World-wide information services
  - Exploitation of new technology
  - Interoperation and portability
- ODP as a standardization effort
  - Additional concepts, consistent terminology
  - Meta-architecture, framework
  - Function standards

Contributions and Schedules



Introduction -- Audience and status

- ODP is aimed for architecture designers and tool developers
- basic reference model is stable (intentions and words), components under development (tools), marketing missing
- object model differs from OMA, TINA, and any object design methodology or object language; closer to component models

Overview on the viewpoints

- Enterprise viewpoint: WHY?
- Information viewpoint: WHAT INFO?
- Computational viewpoint: WHAT LOGIC?
- Engineering viewpoint: HOW?
- Technology viewpoint: STANDARDS?

NO LAYERING, NO PROCESS PRESCRIBED
NO LANGUAGE BINDINGS PRESCRIBED
EACH VP SPEC. MAY HAVE DIFFERENT OBJECTS

System model: Fundamental concepts

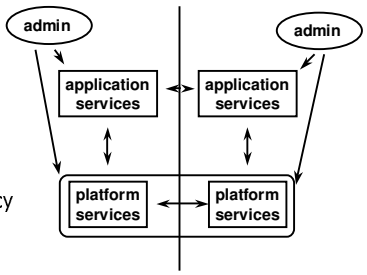
- Objects and interfaces
- Dynamic binding of interfaces
  - Late binding
  - Distribution transparency
  - QoS management
- Organisational concepts
  - domain, federation
- Meta-information management functions

Objects and interfaces

- specification vs. implementation objects
- introduction and instantiation
- separate hierarchies for types and templates
- separation of views to the shared interface: causality, heterogeneity
- multiple interfaces
- operation and stream interfaces

Organizational concepts

- object
- system
- X-community
- Y-domain
- X-federation
- Y-transparency

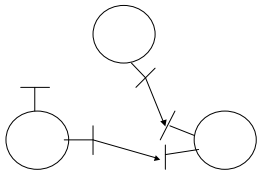


Another overview of viewpoints

- Enterprise – purpose and policy
  - A strategic view
- Information – shared information
  - A unifying view of information resources
- Computational – functional application design
  - An application designer view
- Engineering – structure of middleware
  - A view of communication and resource control
- Technology – known standards

Computational model

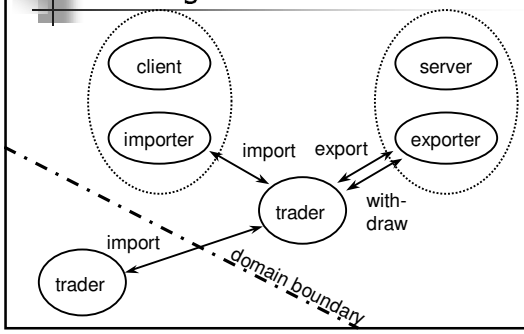
- interactions
  - interrogations
  - announcements
  - streams
- interface binding
  - interface type
  - client/server view
  - interface reference
  - properties of the binding



Infrastructure functionality

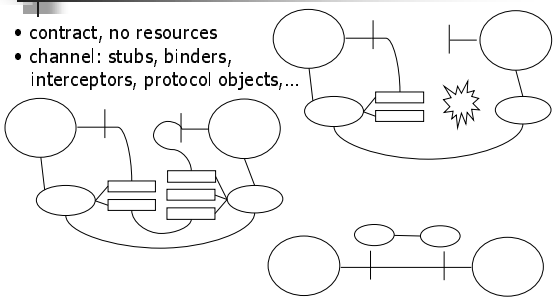
- Trading
  - Mediates information about services
  - Attribute-based search of objects/interfaces
- Type repository function
  - Defines known types and related attributes
- Naming framework with federation
- Binding framework
  - transparency support and QoS management

Trading service



Engineering model of communication

- contract, no resources
- channel: stubs, binders, interceptors, protocol objects,...



Distribution transparency

ODP

- federation transparency
- selective transparencies
  - access
  - location
  - relocation
  - migration
  - failure
  - persistence
  - replication
  - transaction

CORBA

- access transparency
- location transparency

Realisation of ODP binding

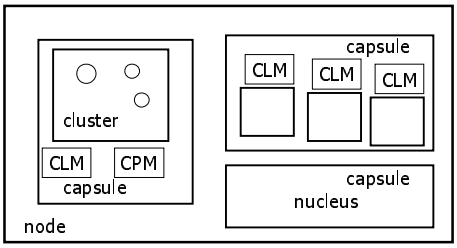
ODP

- trading of services
- type conformance ensured by negotiation during late binding process
- heterogeneity
- dynamic type repository

CORBA

- trading of object instances
- type conformance ensured by inheritance
- polymorphism
- static interface repository

Engineering model for control



Engineering viewpoint functions

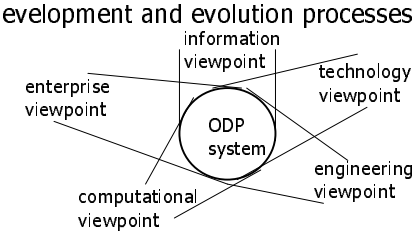
- Node/Object/Cluster/Capsule management
- Coordination functions:
  - event notification, checkpointing and recovery, de/reactivation
  - group, replication, migration, tracking
  - transaction
- Security functions
  - Access control, audit, authentication, integrity, confidentiality, nonrepudiation, key management

Usage

- Viewpoints can be used in software engineering process; part of MDA
- functions like trading developed in liaison with OMG -> CORBA includes
- framework standards direct middleware standards in future

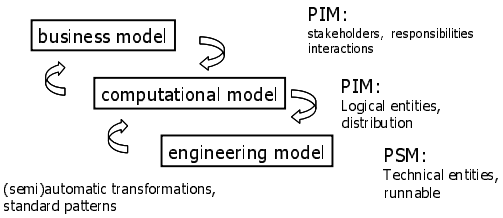
Usage

- Viewpoints for system specification, management and tracing effects of development and evolution processes



Usage

- MDA process (Model driven architecture) by OMG



Where to read more?

- Some tutorials:
  - IS10746-1 (not quite consistent w. Part 2&3)
  - Linington-95, Raymond-95, Blair/Stefani-97
  - Putman: Architecting with RM-ODP
  - CORBA, TINA, ANSA, etc. **DO NOT** share the set of concepts nor goals
- ODP-group in Finland
  - Lea.Kutvonen@cs.Helsinki.FI