Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

Training

ANSAwise - The ODP Reference Model

Yigal Hoffner

Abstract

Organizations wish to deploy IT systems that include old and new IT software and hardware, and use IT services that span organizational boundaries.

Interoperability of heterogeneous distributed systems poses many technical difficulties. Even determining which standards to use is a problem, because of the complexity of the issues from all viewpoints.

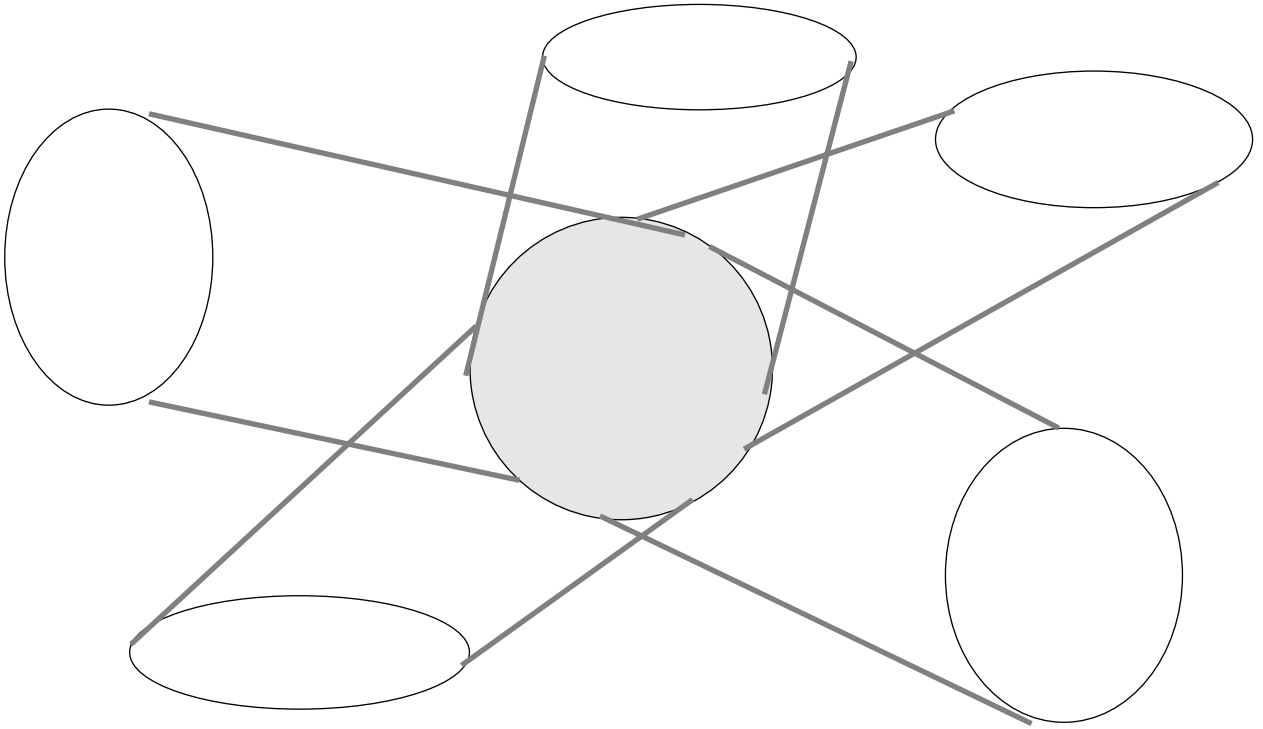
The solution being offered is the use of the ODP Reference Model for specifying and procuring open distributed systems.

APM.1336.01 **Approved** 24th October 1994

Briefing Note

Distribution:
Supersedes:
Superseded by:

The ODP Reference Model



In this session

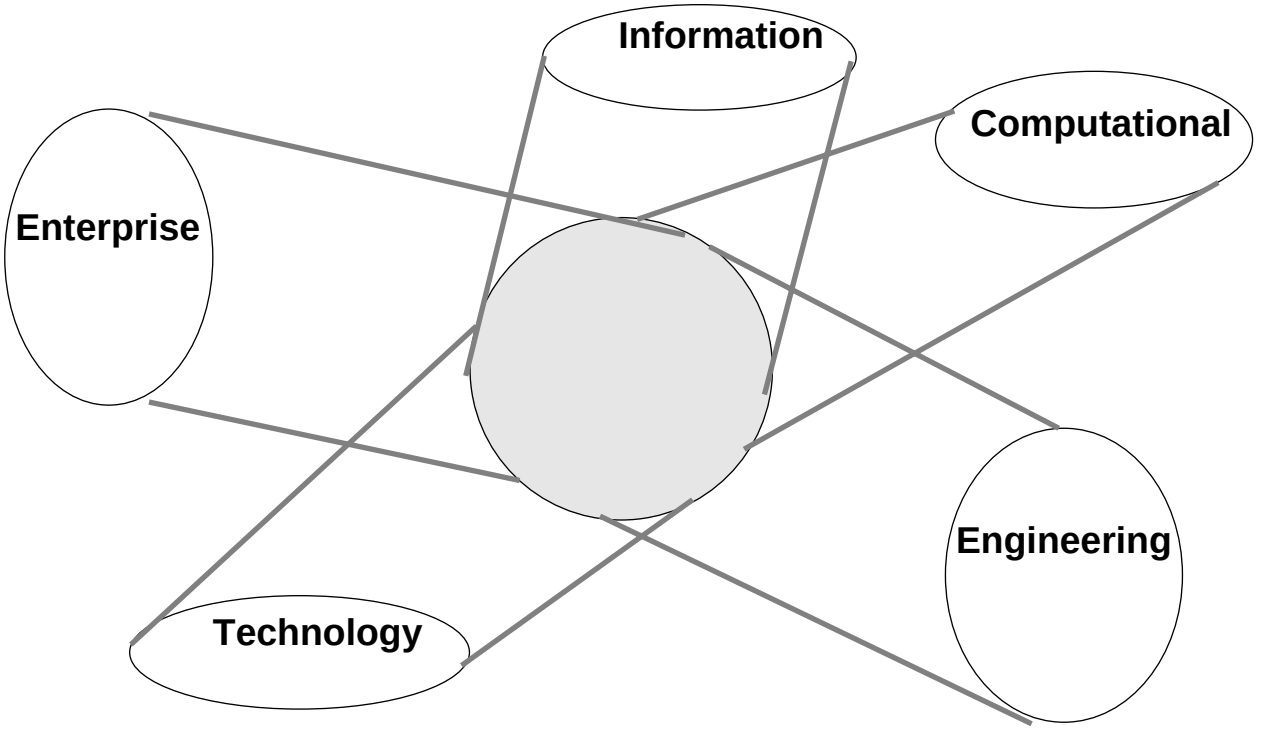
- *Show how separating the viewpoints of a system help you build open distributed systems*
- *Explain the significance of the ODP Reference Model*
- *Explain the goals of ODP*
- *Explain the key concepts of ODP*
- *Show the relationship to other standards*
- *Enable you to find out more*

Distributed Systems have many aspects

- ***Distributed systems involve many different people (the stakeholders)***
  - **business managers, users, IT managers, IT developers,...**
- ***These people are concerned with different aspects of the system***
  - **they see the system from a different viewpoint**
  - **each viewpoint is important**
- ***We need to be able to separate out these concerns when describing distributed systems***
  - **so that each stakeholder can see that their needs are satisfied...**
  - **... without being overwhelmed by descriptions of aspects that are irrelevant to them**

Five different viewpoints

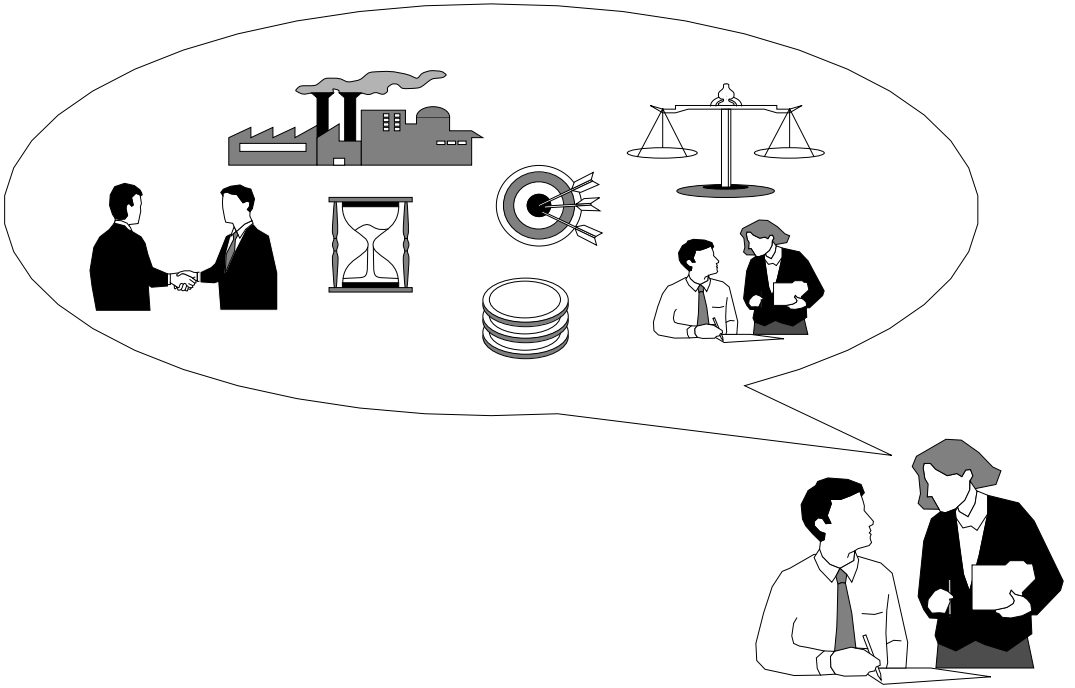
- ***These are of the same system and are <u>not</u> layered***



Content of the five viewpoints

- ***Enterprise*** **- the *purpose* of the enterprise and the system within it**
- ***Information*** **- the *meaning* of the information within the enterprise**
- ***Computational*** **- the *execution* as a model of distributed processing**
- ***Engineering*** **- the *mechanism* for realising the computational model**
- ***Technology*** **- the *conformance* of hardware, operating systems, compilers,...**

The Enterprise viewpoint

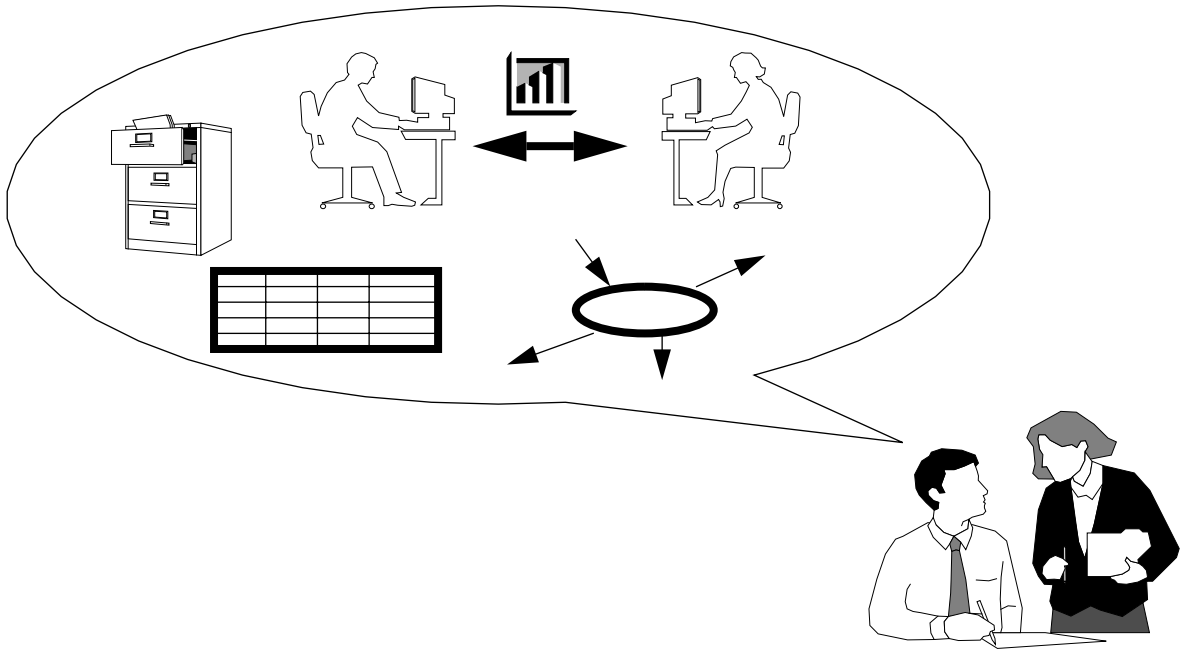


- ***Describes agreements, targets, people, time, money,...***

About the Enterprise viewpoint

- ***More specifically, the Enterprise viewpoint is concerned with***
  - **roles of people, organizations, and systems**
  - **rights, responsibilities, and obligations**
  - **resources**

The Information viewpoint

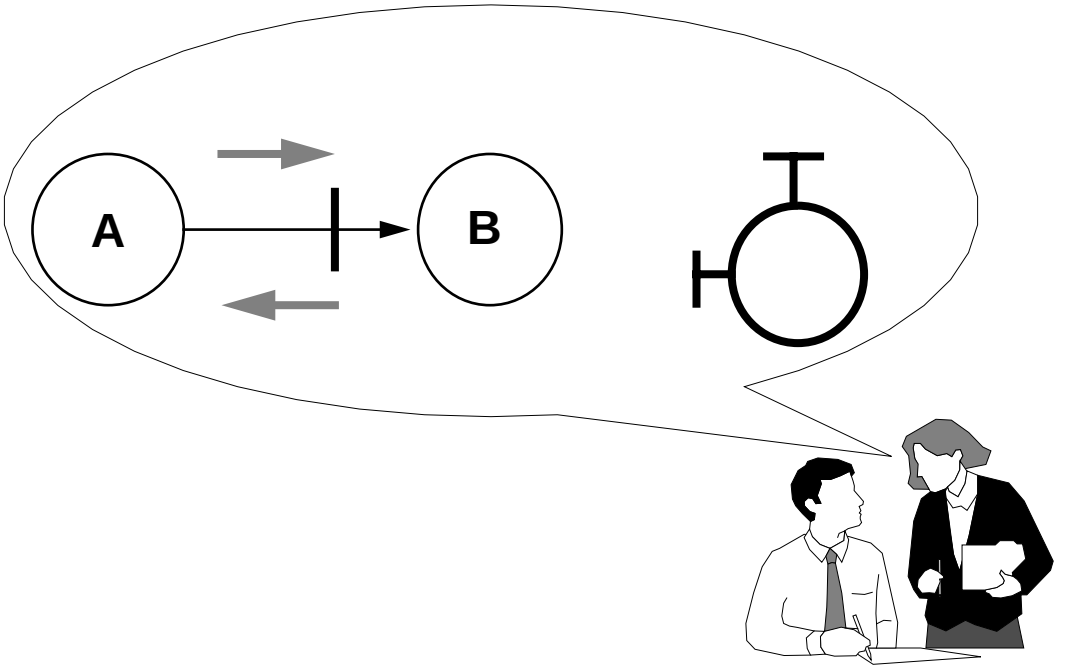


- ***Describes information flows, information stores, information users,...***

About the information viewpoint

- ***The information viewpoint describes objects***
  - **not interfaces**
- ***This is familiar territory for the business analyst or database specialist...***
  - **...schemas, entities,...**

The Computational viewpoint

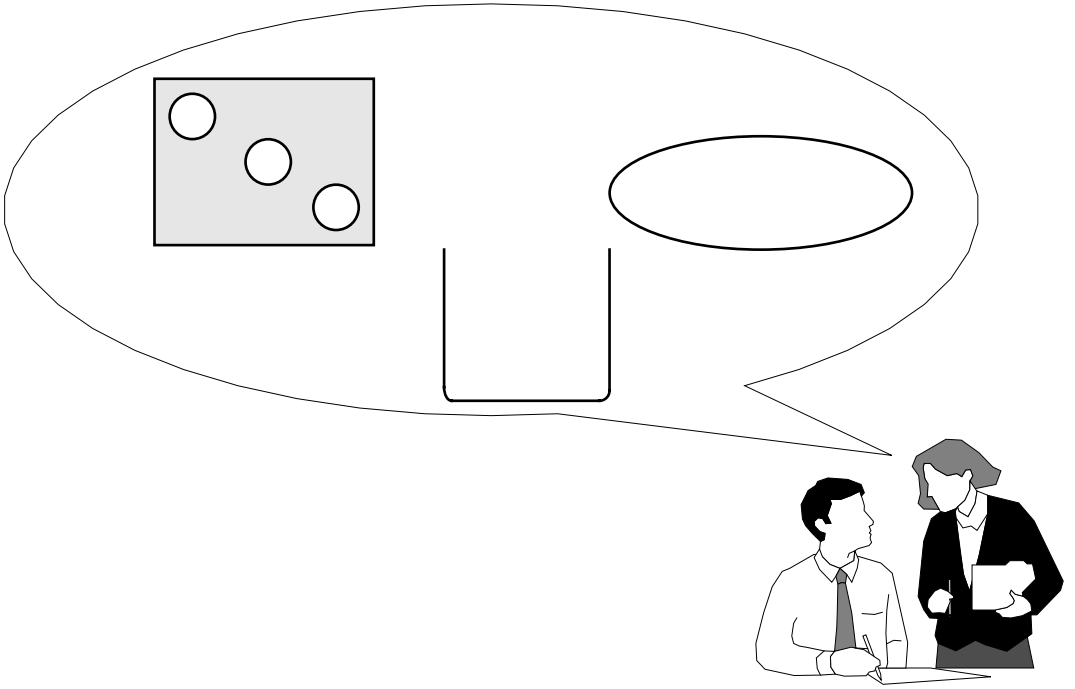


- ***Describes objects, interfaces, operations,...***

About the Computational viewpoint

- ***The distribution of a distributed system is ignored by the Computational viewpoint***
  - **it is transparent to the Computational viewpoint**
- ***From the Computational viewpoint***
  - **resources are always available when needed**
  - **communication between objects is transparent**

The Engineering viewpoint

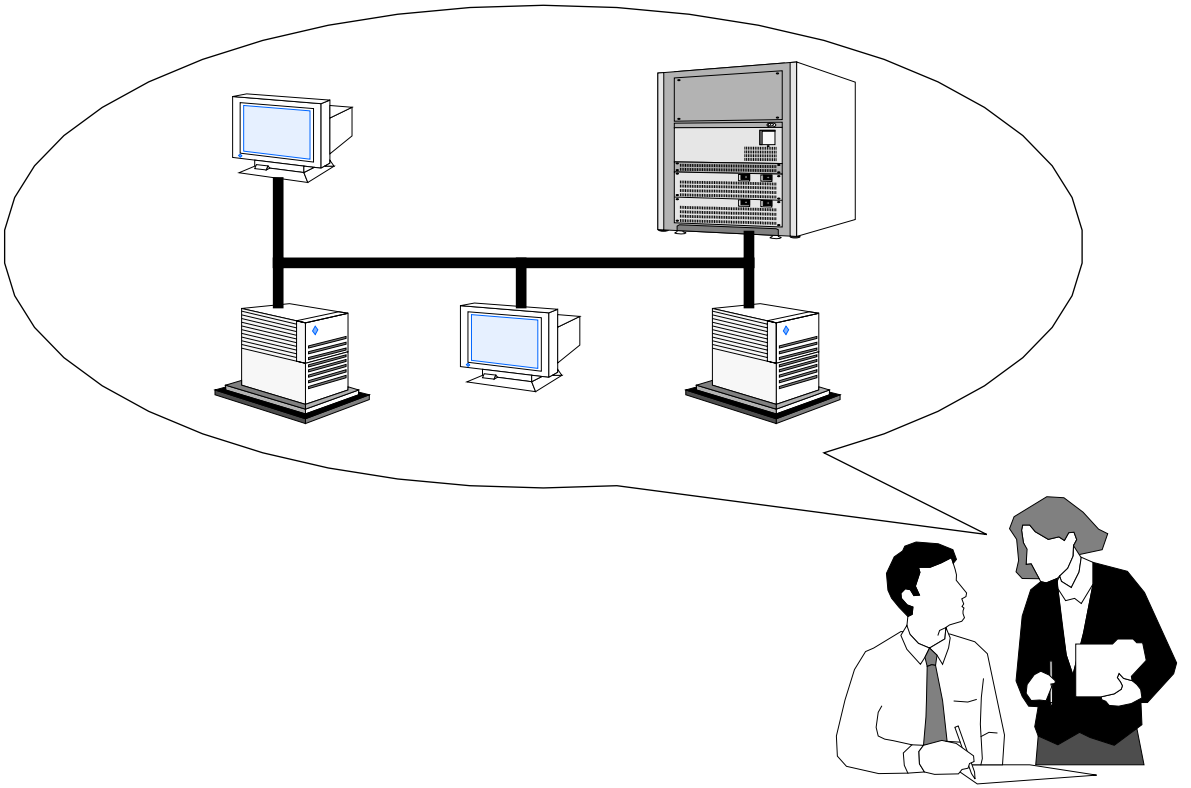


- ***Describes clusters, nodes, channels,...***

About the Engineering viewpoint

- *The Engineering viewpoint describes the infrastructure for the Computational viewpoint*
- *The infrastructure deals with*
  - communication channels between objects
  - resource management
- *It provides transparency mechanisms that hide the distribution from the Computational viewpoint*

The Technology viewpoint



- ***Describes how the system design uses the actual technology***

About the Technology viewpoint

- *The Technology viewpoint is mainly concerned with conformance to standards of actual hardware and software*

- *There are few rules in the Technology viewpoint*

  - **rules will be implementation-dependent**

Interoperability in the five viewpoints

- ***Successful interoperability requires all the viewpoints to work together***
  - **a mismatch in any one can prevent interoperation**
- ***The aim is detect and resolve this mismatch at specification time***
  - **one day there will be software tools to do this automatically**

Technology mismatch

- *Two departments wish to interconnect their LANs*
  - one uses Ethernet...
  - ...the other uses Token Ring

Engineering mismatch

- ***An engineering organization wishes to use an existing database system to store information from a real-time control system***

  - **the real-time control system delivers a periodic data feed...**

  - **... the database can't guarantee to respond in time with an acknowledgement**

Computational mismatch

- ***A company's Marketing department wishes to use the R&D department's document management system to store the master copies of its literature***

  - **the Marketing application accesses documents by filename...**

  - **...the R&D system accesses documents by reference number**

Information mismatch

- *An company wishes to integrate their Marketing and Accounts systems*

  - each keeps information about ‘customers’...

  - ... but their definition of a ‘customer’ is different

Enterprise mismatch

- *Two airlines wish to connect their reservation systems*
  - each has a policy on cancellations...
  - ...one gives automatic refunds
  - ...one automatically rebooks

What is ODP?

- ***Open Distributed Processing is a goal***
  - **the ability to create open distributed systems...**
  - **... connecting all kinds of IT systems**
  - **.... spanning organizational boundaries**
- ***Specifically, ODP aims to provide***
  - **interoperability of applications between distributed systems**
  - **portability of applications between distributed systems**
  - **... in a way that is *transparent* to the applications**

RM-ODP

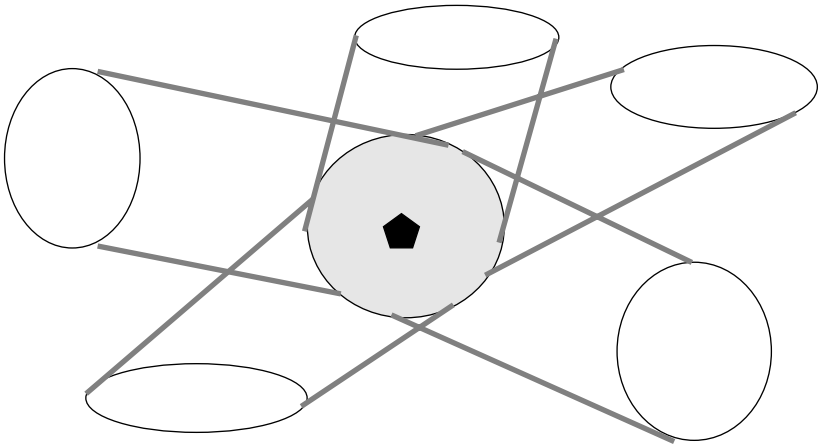
- ***The Basic Reference Model for Open Distributed Processing (RM-ODP) is...***
  - **an architectural framework for understanding the problems and concerns of distributed systems**
  - **a framework for assessing the conformance of a particular system**
  - **a forthcoming international standard**
- ***...a starting point for ODP standards***
- ***Standardization is essential for openness to be achieved***

Key concepts of RM-ODP

- ***Objects and interfaces***
- ***Transparency***
- ***Viewpoints***

The viewpoints are linked into a framework

- ***Because the viewpoints are views of the same system...***

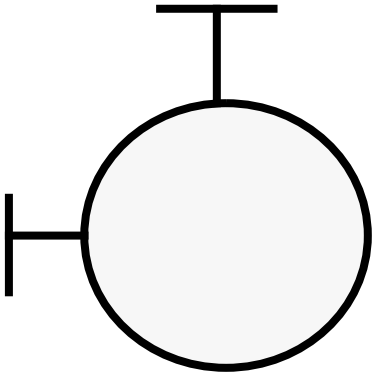


  - **to make sure the viewpoints are consistent with each other**

Service

- *To separate a system into parts, each part must offer a coherent service*
- *The service must be explicitly specified*
- *Specifications are declarative; what, not how*
- *The provider of a service agrees to meet the specification*
- *The provider does not reveal how the service is provided*
  - **it could be via a mainframe legacy system**
- *... in a word, encapsulation*

Objects for encapsulation



- ***Objects are encapsulated...***
  - **...all interactions are via defined interfaces**
  - **...all objects interact in the same way**

Examples of objects in the different viewpoints

- *Enterprise viewpoint*
  - a person, organization, or resource
- *Information viewpoint*
  - an information entity
- *Computational viewpoint*
  - an encapsulation of behaviour/state
- *Engineering viewpoint*
  - a channel controller
- *Technology viewpoint*
  - a machine

Distributed systems are different

- **Many traditional system design assumptions must be reversed**

Traditional	*Reversed*
Local	Remote
Sequential	Concurrent
Homogeneous Environment	Diverse Environment
Fixed Location	Mobile
Single Copy	Multiple Copies
Synchronous	Asynchronous
Direct	Indirect
Shared	Separate
Global	Context Relative
Complete Failures	Partial Failures
Early Binding	Late Binding

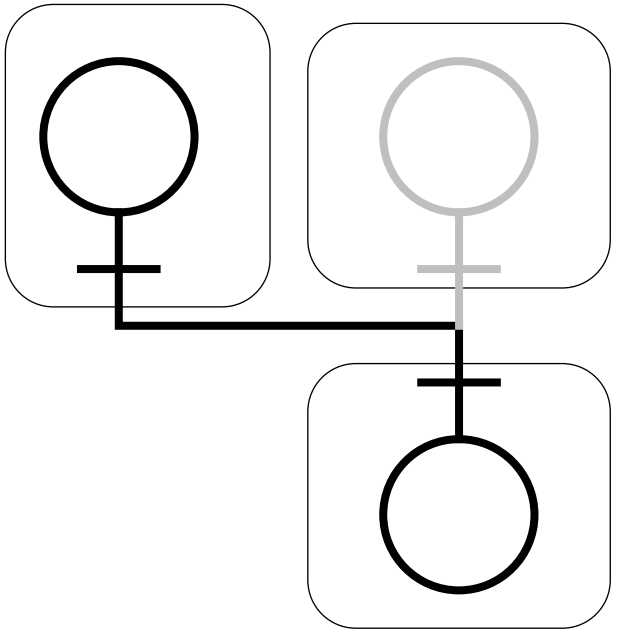
- ***A systematic approach is needed to avoid these assumptions***

The transparency approach

- ***Applications should not be burdened with the complexity of handling these reversed assumptions***
- ***Something else must handle this complexity...***
- ***... transparency mechanisms in the infrastructure***

Example Transparency - Migration

- **Migration Transparency**
  - **application need not know where the object has moved to**



Exploiting the reversed assumptions

- ***Exploit positive consequences***
  - **Consider, for example...**
  - **Late binding: Trading supports choice of Quality of Service**
  - **Multiple copies: Concurrency supports parallelism**
  - **Partial failure: Replication supports availability**
- ***Mask negative consequences***
  - **Use selective *transparency* mechanisms, for example...**
  - **Migration transparency: Isolates client from service relocation**
  - **Replication transparency: Isolates client from multiple copies of service**

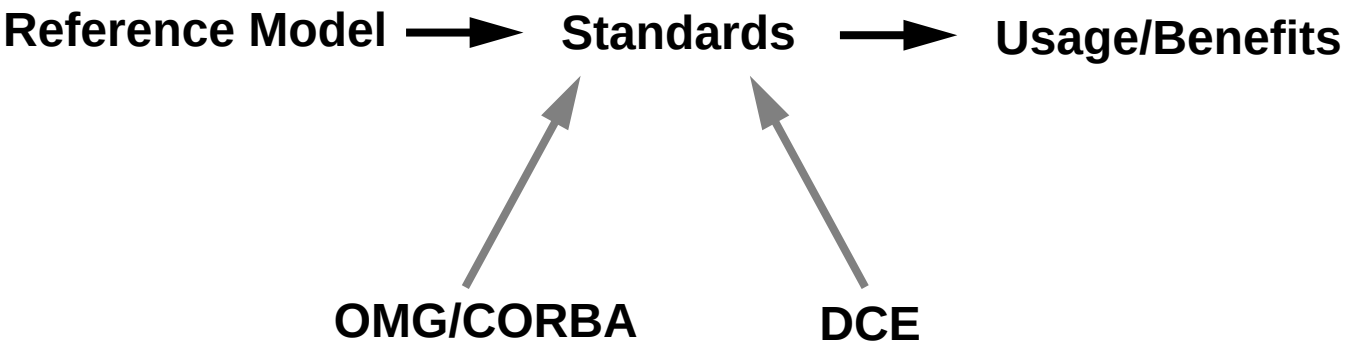
Handling the reversed assumptions - The Computational and Engineering viewpoints

- *Isolate specification of transparencies from their design*
  - Computational viewpoint defines the transparencies
  - Engineering viewpoint provides the mechanisms
  - Applications developers just state which transparencies they need
- *Automate the building of transparencies*
  - Software tools can construct transparencies from the engineering mechanisms

RM-ODP is a starting point

- *It is a framework for the development of other ODP standards...*
  - standards for specification, modelling and programming languages
  - language mappings (APIs) for ODP systems
  - functional components of ODP systems (specific services)
- *...a framework for relating the different ODP standards to each other*
- *RM-ODP contains the concepts and rules needed to write these standards*
- *Many relevant standards already exist*
  - they need to be fitted into the framework

Relationship to other standards



- ***Liaison is in place with OMG, and the framework is being populated***
  - **function correspondence has been identified**
- ***CORBA and DCE are working bottom-up...***
- ***...ODP is working top-down***

Summary

- ***The Basic Reference Model of ODP (RM-ODP) is a framework standard***
  - **more detailed standards are needed to populate the framework**
- ***RM-ODP simplifies the design of distributed systems***
  - **using viewpoints to separate the concerns of stakeholders**
  - **using objects and interfaces for encapsulation**
  - **using transparencies to mask distribution from applications**
- ***For more information on ODP***
  - **for more on transparency mechanisms, see *The Challenge of ODP* (TR.033.02)**
  - **for reading RM-ODP itself, suggestions are given below**

Organization of the RM-ODP Standard

- *The standard is in four parts*
  - Part 1: Overview and guide to use (ISO/IEC 10746-1, ITU-T X.901)
  - Part 2: Descriptive model (ISO/IEC 10746-2, ITU-T X.902)
  - Part 3: Prescriptive model (ISO/IEC 10746-3, ITU-T X.903)
  - Part 4: Architectural semantics (ISO/IEC 10746-4, ITU-T X.904)
- *Each part describes the Reference Model in a different way*
  - Part 1 is an informal overview and rationale in plain English
  - Part 2 is a definition of the concepts and analytical framework
  - Part 3 is a specification of the characteristics of an ODP system
  - Part 4 is a definition of the concepts in terms of other formal description techniques (LOTOS, SDL, Estelle, Z)

The general flavour of RM-ODP

- *The style of each of the RM-ODP Parts is different*
  - Part 1 contains examples of an ODP system described from each of the five viewpoints
  - Part 2 contains a list of definitions
  - Part 3 contains a list of rules for each viewpoint
  - Part 4 contains a formal description
- *RM-ODP is hard to grasp...*
  - ...not because of detail, or length, but because it is so abstract
- *Start with Part 1*

RM-ODP Part 1: Overview and guide to use

- *Contains an overview of the ODP, rationale, explanations of key concepts, and some examples*
- *The examples show how to use RM-ODP to identify where more detailed standardization is necessary*
  - at reference points for conformance identified in Part 3
- *A suggestion for understanding Part 1:*
  - Start by reading the first few sections, then look at the examples later on, to see how the five viewpoints are used...
  - ... or follow the suggestions given at the beginning of Part 1 itself

RM-ODP Part 2: Descriptive model

- ***Defines the ODP key concepts***
  - **The definitions are sufficient to support the formal semantics of Part 4**
  - **The definitions are sufficient to establish requirements for new specification techniques**
- ***These definitions are terse, highly abstract, and strongly inter-related; for example:***
  - **"*Failure*: Violation of a contract"**
- ***A suggestion for understanding Part 2:***
  - **Stick to one viewpoint at a time**
  - **Find a concept in Part 1 (or Part 3) that is of interest**
  - **Follow through the definitions in Part 2, and refer back to Part 1**

RM-ODP Part 3: Prescriptive model

- ***Specifies rules that a distributed system must follow if it is to be an ODP system...***
  - **'structuring rules' using the concept definitions of Part 2**
  - **conformance and reference points of an implementation at which these rules can be checked**
  - **consistency rules between specifications from different viewpoints**
  - **...these rules must also be followed by other ODP standards (outside the RM-ODP)**
- ***Specifies the ODP functions and transparencies***
- ***A suggestion for understanding Part 3:***
  - **Start by reading about the ODP functions and transparencies**

RM-ODP Part 4: Architectural semantics

- *Contains a formal description of the basic RM-ODP Part 2 concepts*
  - in LOTOS, SDL, Estelle, and Z
- *These formal descriptions map RM-ODP concepts to the corresponding concepts of LOTOS, SDL, Estelle, and Z*
  - sometimes there is no direct equivalent
- *A suggestion for understanding Part 4:*
  - Read the section that uses a formal description technique you already know

Status of RM-ODP

- *RM-ODP is being standardized jointly by ISO, IEC, and ITU-T (formally CCITT)*
  - RM-ODP is based on work pioneered by ANSA
- *Now at committee draft status*
- *Each Part is progressing separately*

Other ODP standards

- *ODP components*
  - Trader was chosen as the first component to be standardized
  - Type Manager next to come
- *Profile*
- *Management*
- *Security*

Finding out more about ODP

- *Via APM*
  - Andrew Herbert (editor of Part 3)
- *ISO/IEC JTC1/SC 21/WG7 Project 21.43*

  Secretariat:
  Standards Association of Australia,
  PO Box 1055, Strathfield,
  NSW, Australia 2135;

  Tel: +61 2 746 4830;
  Fax: +61 2 746 8450