



---

**Poseidon House  
Castle Park  
Cambridge CB3 0RD  
United Kingdom**

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (01223) 515010  
+44 1223 515010  
+44 1223 359779  
apm@ansa.co.uk**

---

## **Training**

# **ANSAwise - Object-Oriented Methods for Distributed Systems**

**Chris Mayers**

### **Abstract**

Organizations wish to deploy services rapidly; software reuse and distributed processing appear to offer a way forward.

Object-oriented methods promise reuse, but provide little (if any) support for distribution of the applications being modelled.

This module of the ANSAwise training programme reviews the Object Modelling Technique (OMT), and shows how it can be used with ODP and applied to telecommunications applications. It was originally prepared for the CNET course "Introduction to Design of Distributed Systems". The approach suggested is based on work done in European projects by CNET and other telecommunications companies.

---

APM.1596.00.02

**Draft**

13th March 1996

Briefing Note

---

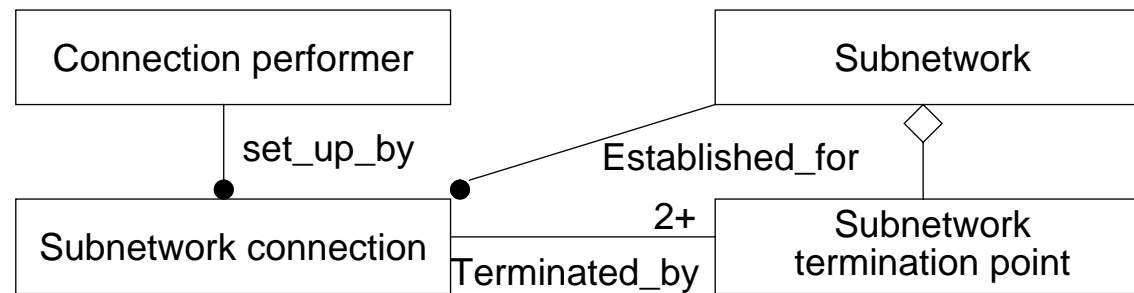
**Distribution:**

**Supersedes:**

**Superseded by:**



# Object-Oriented Methods for Distributed Systems





## In this session

- **Show how an object-oriented method can help develop distributed applications**
  - despite limited support for distribution within the method
- **Show the alignment of the concepts in**
  - **Object Modeling Technique (OMT)**
  - **Open Distributed Processing (ODP)**



---

## Object Modelling Technique (OMT)

- Like most object-oriented methods, OMT analyzes a system from three viewpoints
- From each viewpoint you produce an analysis model of your system
  - Object model
  - Dynamic model
  - Functional model



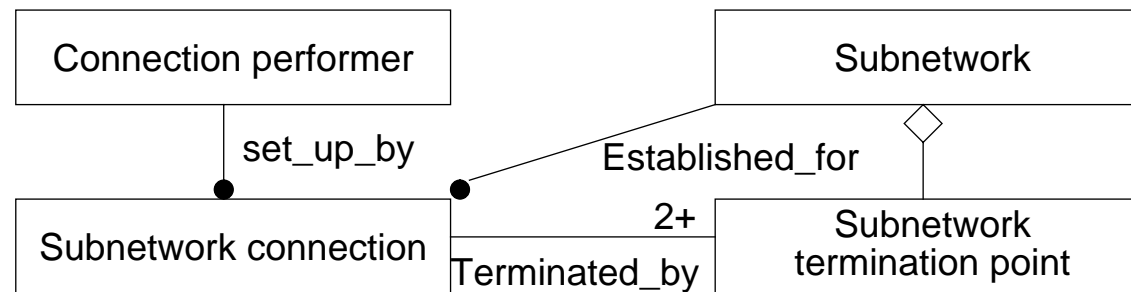
---

## A telecommunications network management example

- Shows the OMT modelling techniques
- Shows the OMT models of the structure of a telecommunications network
- Shows how the management of this network can be distributed

## OMT Object Model

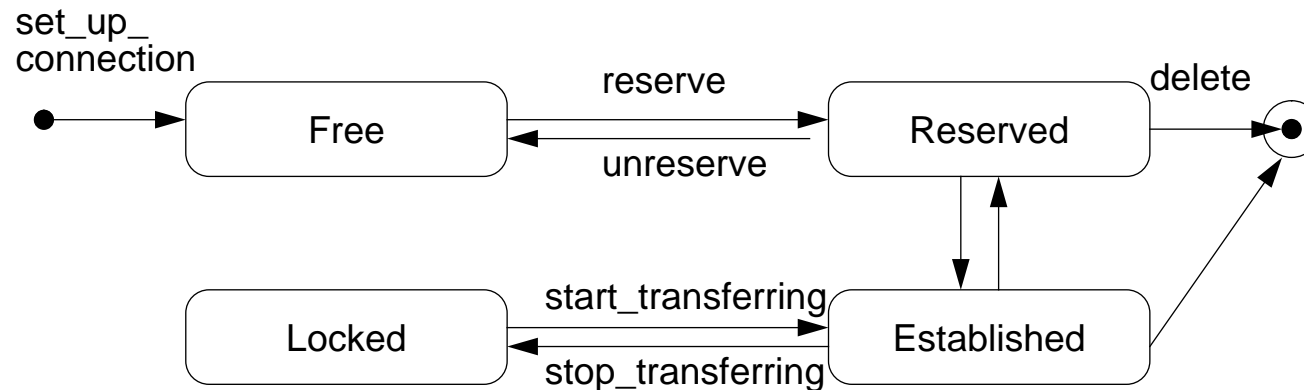
- Represents the static, structural 'data' aspects of the system...



- ... the information relationships

## OMT Dynamic Model

- Represents the temporal 'control' aspects of the system...

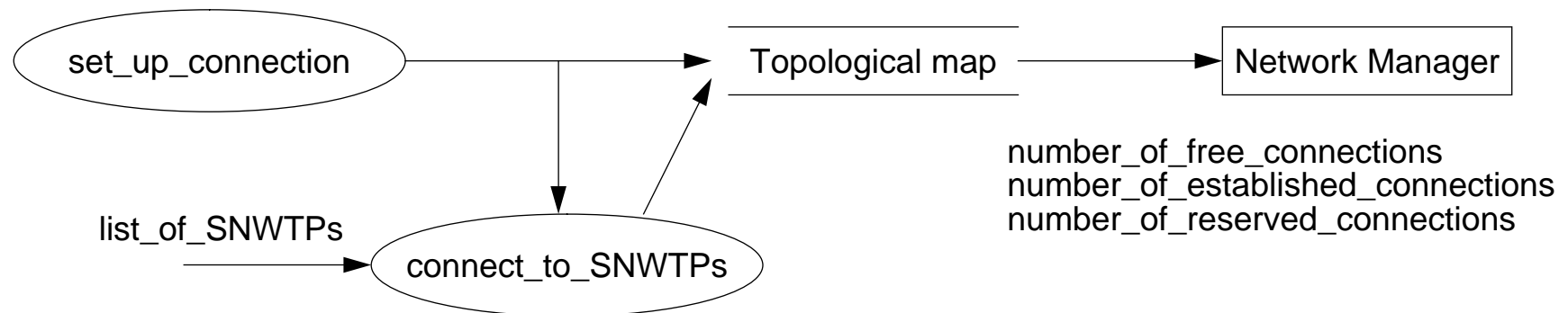


- ...the state transitions



## OMT Functional Model

- Represents the ‘transformational’ aspects of the system



- ... the data flows



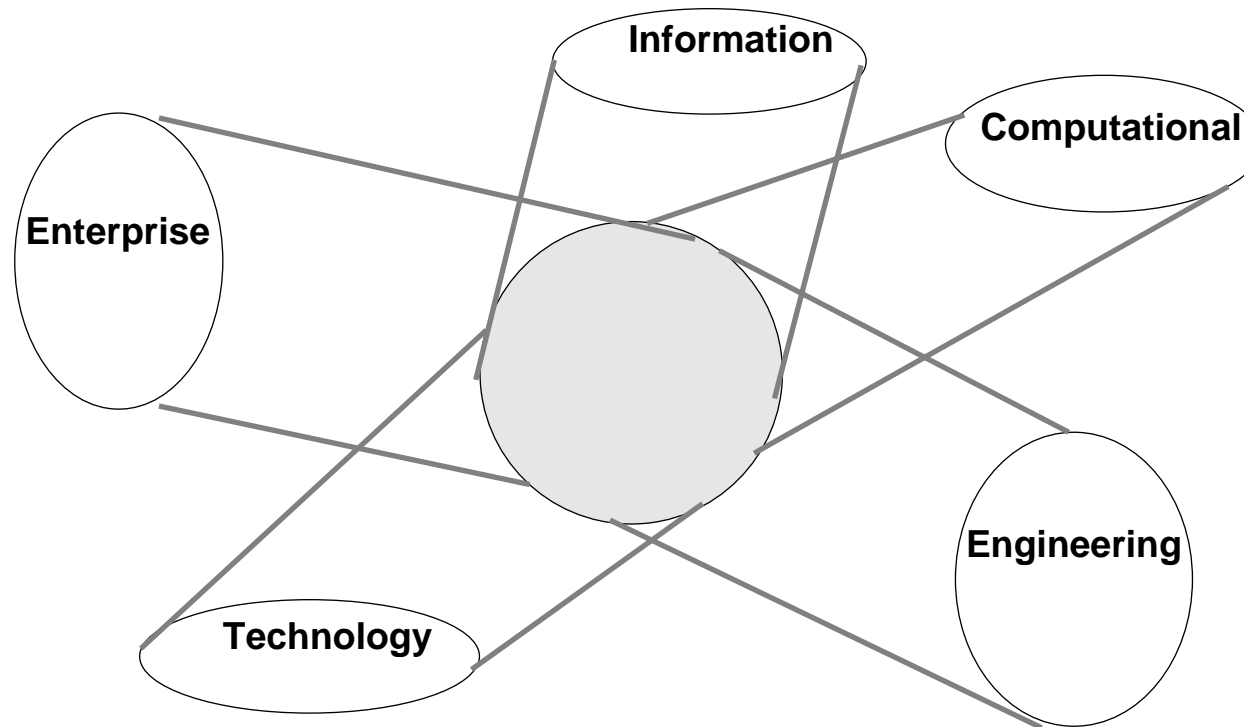
---

## The OMT Methodology

- **Analysis**
  - define the problem
  - build an object model, dynamic model, and functional model
- **System Design**
  - organize the system into subsystems
  - define policies/trade-offs for data storage, concurrency, resource management,...
- **Object Design**
  - design algorithms
  - design class structure, data structures, attribute representation,...

## The ODP Viewpoints

- These are of the same system and are not 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,...

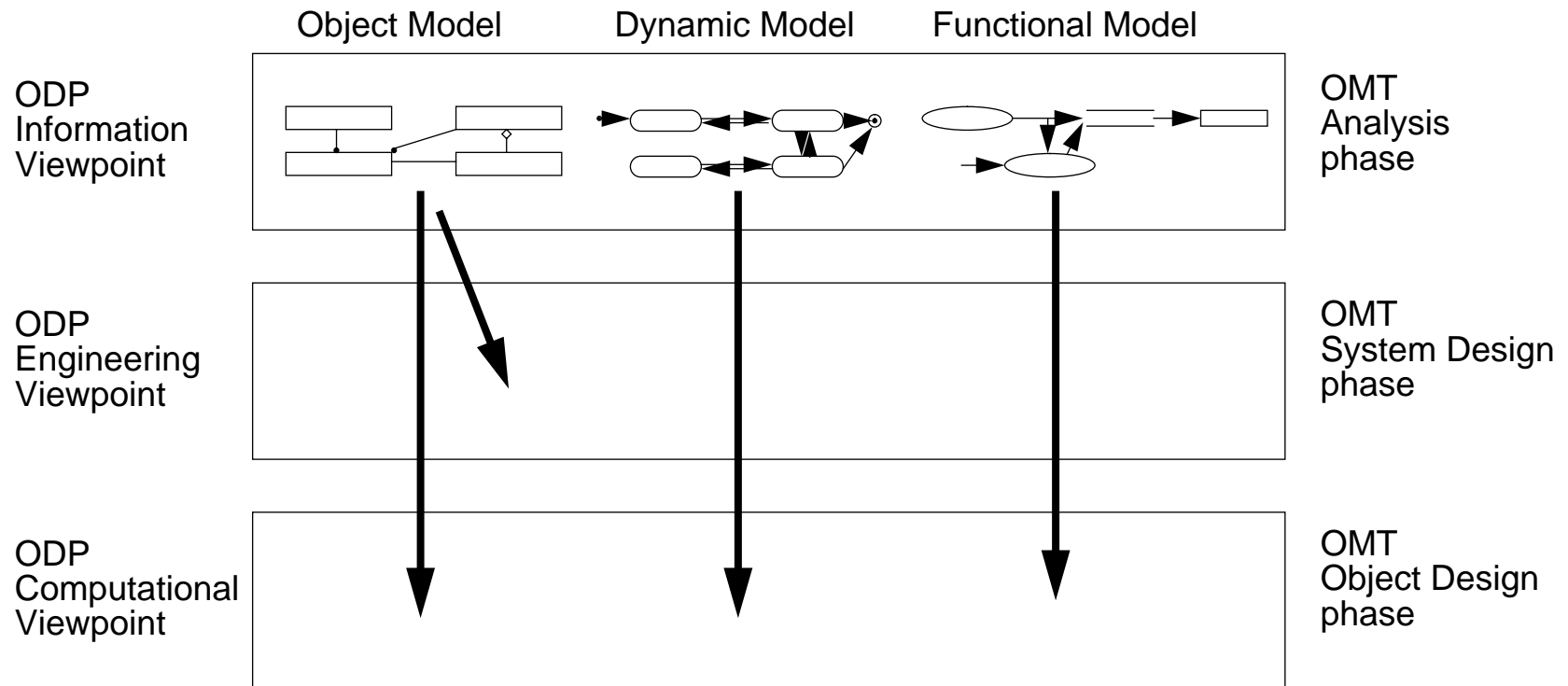


---

## Linking OMT and ODP Development

- **The OMT Analysis phase yields an Analysis Document**
  - its OMT Object Model corresponds to the ODP Information viewpoint
- **The OMT System Design phase yields a System Design Document**
  - no specific OMT modelling techniques are used; the issues correspond to some of those in the ODP Engineering viewpoint
- **The OMT Object Design phase yields a Design Document**
  - containing Detailed OMT Object, Dynamic and Functional Models

# OMT and ODP Development





## Issues with linking OMT and ODP Development

- **There is no OMT equivalent to the ODP Enterprise, Computational and Technology viewpoints**
- **The OMT Methodology requires Analysis before System design...**
  - ... ODP Information viewpoint before Engineering viewpoint
  - ... ODP allows the viewpoints to be specified in any order
- **The OMT Object Design is an elaboration of the Analysis**
  - detailed Object, Dynamic, and Functional Models...
  - ... no separation of Information, Computational, and Technology concerns



---

## Hidden OMT assumptions

- **OMT assumes the entire system can be specified and built as one project**
  - ...under the control of one organization
  - ...delivered as one product
  - ...installed once and never changed
- **OMT assumes that distribution is a system design issue only**





---

## Limitations of the OMT approach - elaboration

- **OMT's elaboration from Analysis to Object Design is 1-1**
  - assumes that one analysis object corresponds to one design object...
  - ...this limits design flexibility
  - in practice, analysis objects often need to be distributed, so cannot be a single design object
- **OMT's System Design allocates software at the granularity of the subsystem**
  - this granularity is usually too coarse



---

## Limitations of the OMT approach - distribution

- **OMT does not support distribution**
  - no distinction between objects with interfaces for local use and remote use...
  - ... must either distribute all objects, or none
- **OMT system design does not model the distribution infrastructure explicitly**
  - as the ODP engineering model does



## Limitations of the OMT approach - transparencies

- **No concept of distribution transparencies**
- **No support in the method for concurrency or transactions**
- **No support for relocation or other transparencies**
- **Interoperability is not explicitly addressed**



## Avoiding the limitations

- **Add extra steps to the OMT System Design phase**
  - **model the distribution infrastructure**
  - **identify transparency requirements**
  - **specify ODP computational interfaces derived from the OMT Object Model**



## OMT Object Model and ODP Computational Model

- **The OMT Object Model also lacks some important specification concepts**
  - **multiple interfaces to an object**
  - **stream interfaces**
  - **quality-of-service**



---

## PREPARE (Pre-Pilot in Advanced Resource Management)

- **An EU RACE II project (R2004)**
- **Focusing on co-operation between private and public management systems**
  - **including standardization of inter-domain management interfaces**
- **Building a broadband testbed demonstrator with Virtual Private Network (VPN) services**



## PREPARE - Mapping between GDMO and OMT models

OMT	GDMO
associations (object model)	association attributes
aggregations (object model)	name bindings
inheritance (object model)	derivations
states (dynamic model)	state attributes
events (dynamic model)	operations
actions and activities (dynamic model) processes (functional model)	behaviour

- **This is not the only possible mapping**
  - **this is under study in the CORBA Telecom SIG**



## The future of OMT

- **OMT was originally developed by James Rumbaugh of GE Labs**
  - who has now joined Rational, the developers of the Booch method...
  - ...along with Ivar Jacobson, the developer of the Objectory/OOSE method
- **Rational have announced version 1.0 of the 'unified method'**
  - "expected to be released in early 1996".
- **The unified method will include the best features of these and other methods**
  - in particular, 'use cases' from Objectory/OOSE...
  - ...but no support for distribution, nor for ODP?



## Methodologies in use

Methodologies in use	Percentage of organizations using
Rumbaugh OMT	30+
Booch OOA&D	9+
Jacobson Objectory/OOSE	9+
Martin/Odell OOIE	4
Object Designers Syntropy	4
Yourdon OOD, OOA, OOP	4
'Bits of several'	22
None in particular	22

Source: Andersen Consulting



## Summary

- **ODP and OMT are broadly compatible and complementary**
- **Vendor tool support for ODP+OMT is not likely in the near future**
  - **you will need to adopt your own techniques for using these tools**
- **TINA is pioneering these integrated techniques**
- **Much more work is needed to support ODP+OMT over the whole development life-cycle**
  - **from system requirements to system maintenance**



## More information?

- For the background to this example
  - see *Analysis and design of a management application using RM-ODP and OMT* by Erik Colban and Fabrice Dupuy (in ICODP '94)
- For a technical description of OMT
  - see *Object-Oriented Modeling and Design* by Rumbaugh et al. (Prentice Hall)
- For another example of OMT used with ODP...
  - ... a multimedia conferencing system
  - ... see the Part 1 of the ODP standard itself (ISO/IEC 10746-1, X.901)
- For information on TINA
  - see <http://www.tinac.com>