



**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**

APM Business Unit

ANSAwise Training - topics and modules

Chris Mayers

Abstract

Customers need to find out more about distributed systems; APM are mounting a training programme (ANSAwise) to meet this need.

APM wish to reduce the effort in course development and improve the quality of the materials by making the courses modular.

The document gives a one-page summary for each module (nominally a 1-hour presentation). This document is written as slides just for convenience in producing and reviewing these outlines.

The document may also be supplier to course partners and resellers.

APM.1273.00.04

Draft

2nd November 1994

Request for Comments (confidential to ANSA consortium for 2 years)

Distribution:

Supersedes:

Superseded by:



ANSAwise - Course Roadmaps

The following diagrams show the structure of the ANSAwise courses. Each box is one module (nominally an hour). The modules are grouped by theme; this grouping is not rigid. Course open/closing sessions are not shown, but must have time allowed for them.

The actual running order of modules within a course must allow for the dependencies between the course material (shown on these diagrams), but is also chosen for balance and variety in each day. A typical timetable for each course is at the end of this document.

A description of each module follows.



ANSAwise - Key to Roadmaps

Core module

**Optional
module
(sponsor only)**

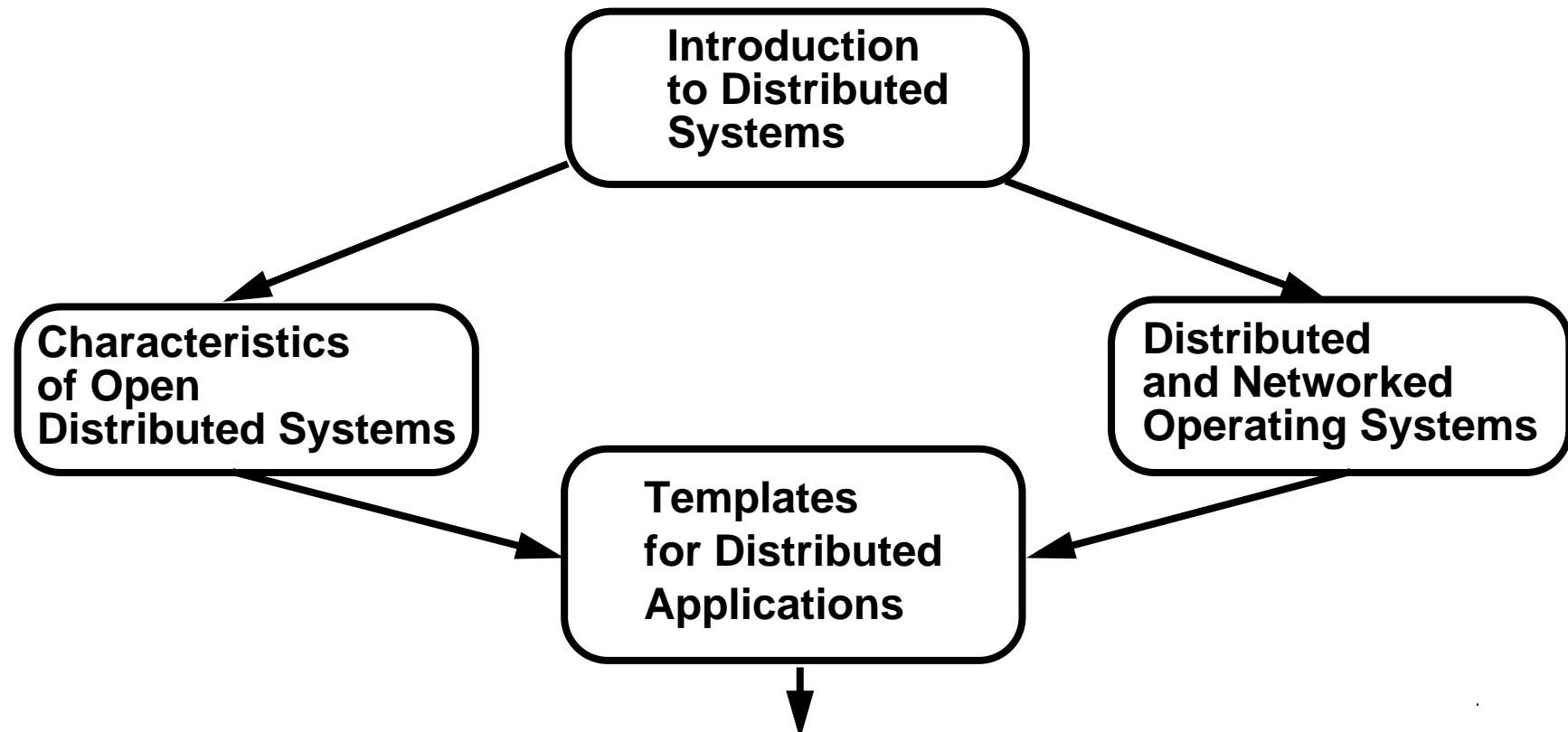
**Optional
module**

Some optional modules are available only to sponsors, because they contain non-public material from the ANSA Phase III programme.

For customized courses, customers can select among optional modules, subject to course length



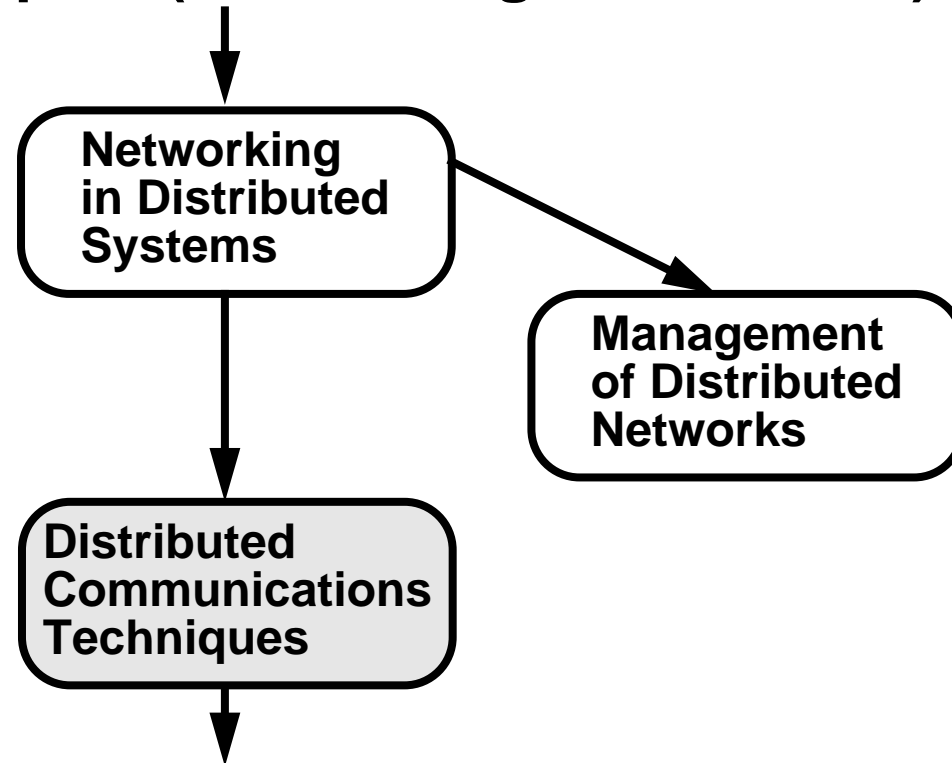
Understanding Distributed Systems Architecture Course Roadmap - 1 (Business and Technical Context)





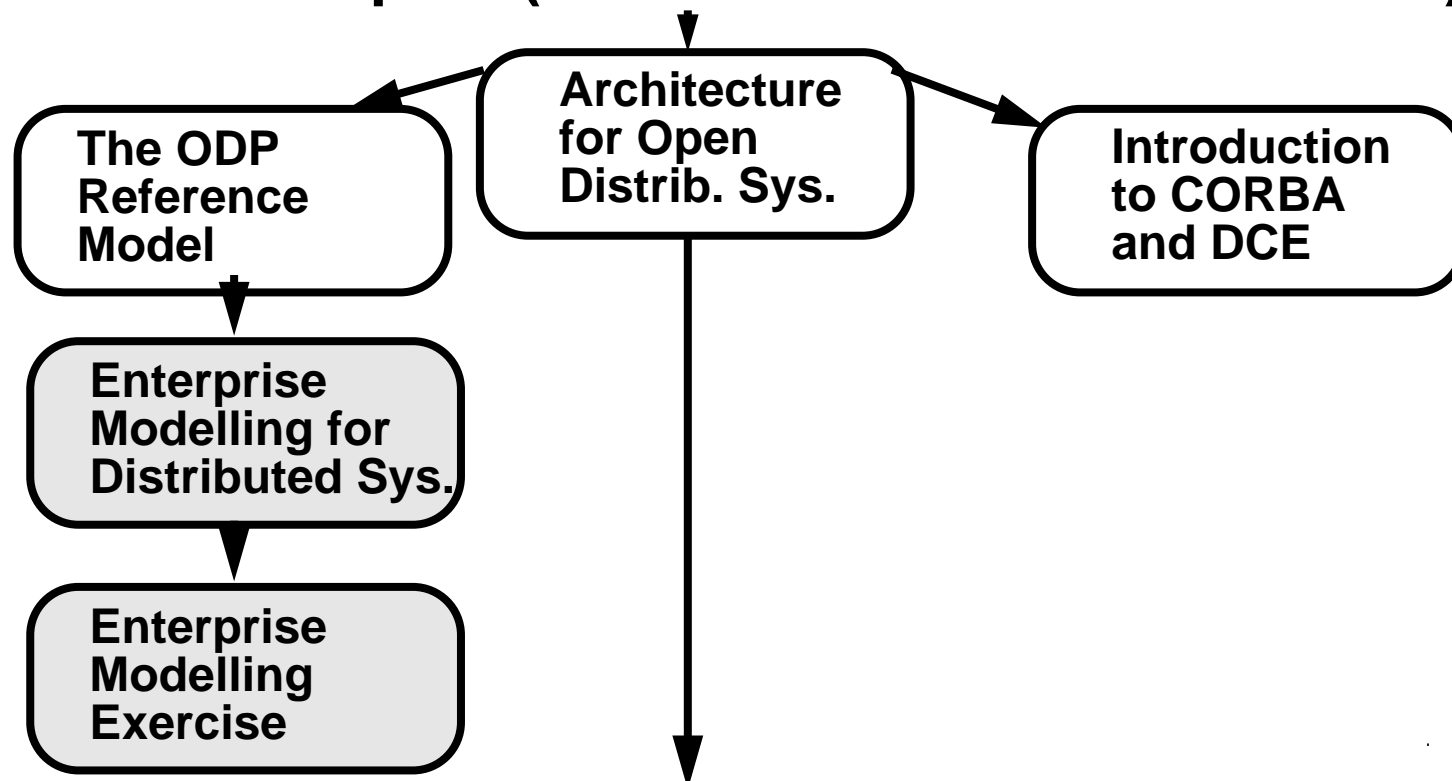
Understanding Distributed Systems Architecture

Course Roadmap - 2 (Networking and Control)



Understanding Distributed Systems Architecture

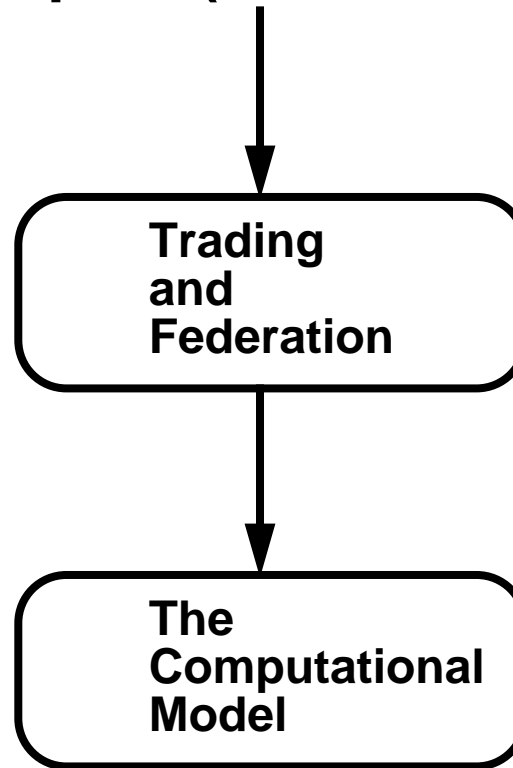
Course Roadmap - 3 (Architecture and Frameworks)





Understanding Distributed Systems Architecture

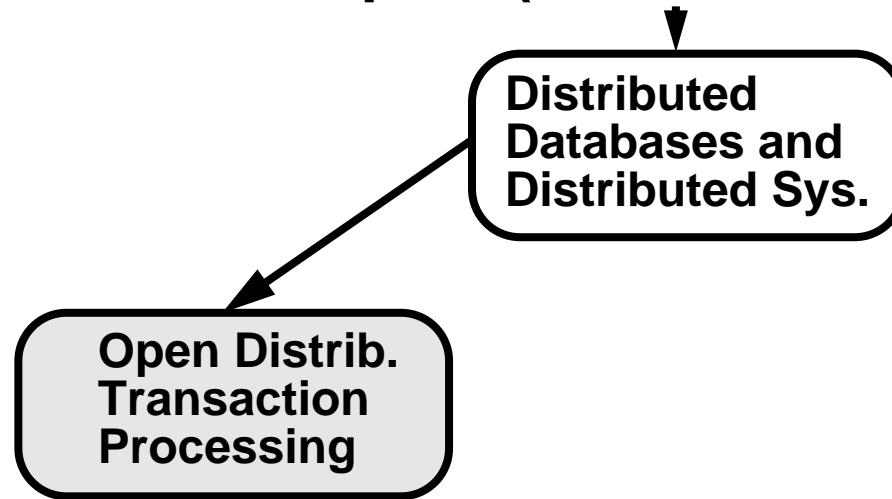
Course Roadmap - 4 (Models and Specification)





Understanding Distributed Systems Architecture

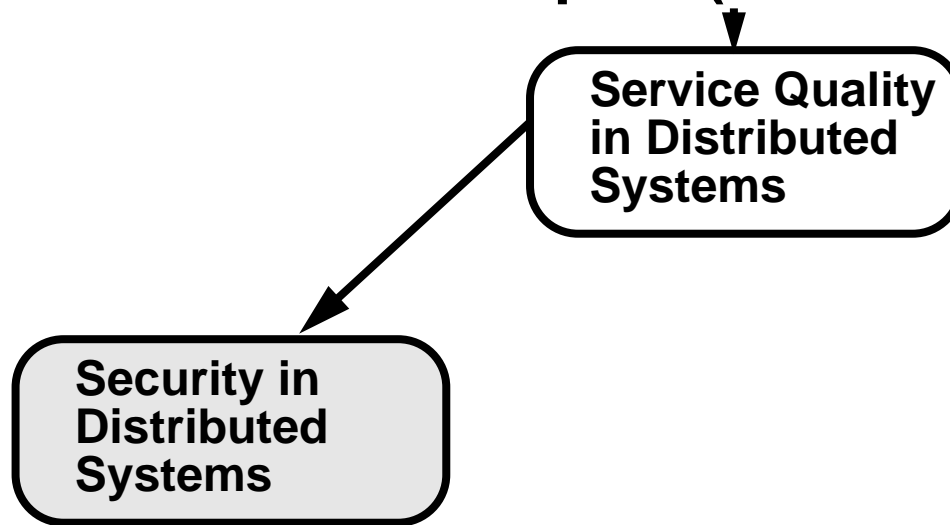
Course Roadmap - 5 (Database and Legacy Systems)





Understanding Distributed Systems Architecture

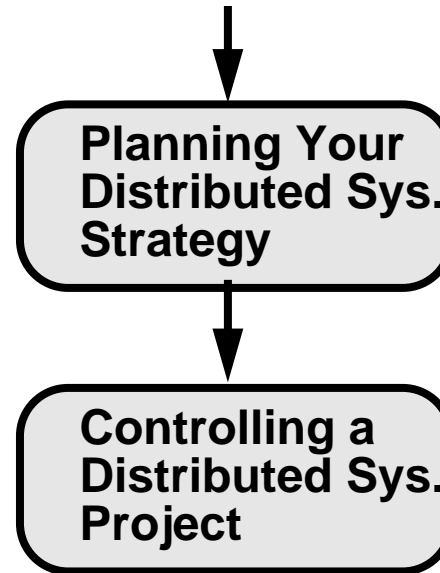
Course Roadmap - 6 (Service Dependability)





Understanding Distributed Systems Architecture

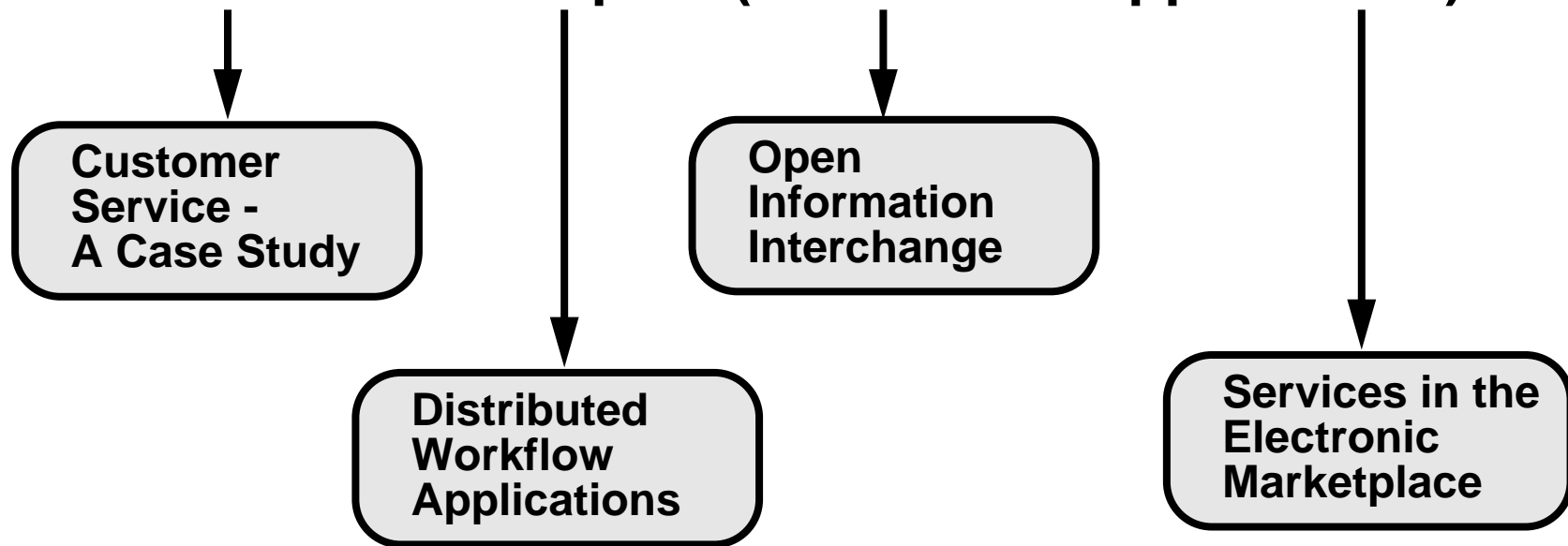
Course Roadmap - 7 (Planning)





Understanding Distributed Systems Architecture

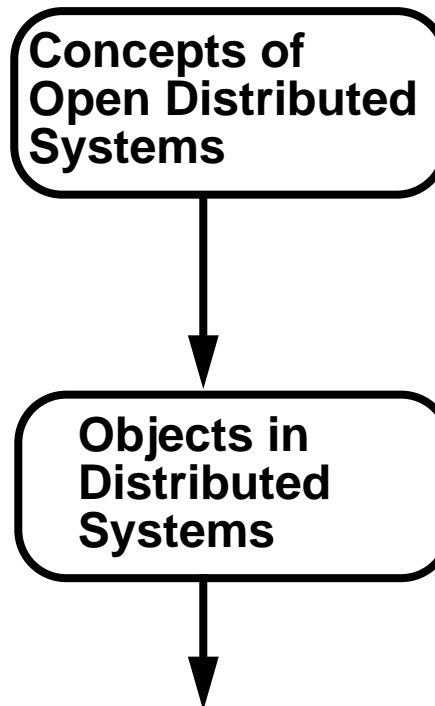
Course Roadmap - 8 (Distributed Applications)





Building Applications with Distributed Objects

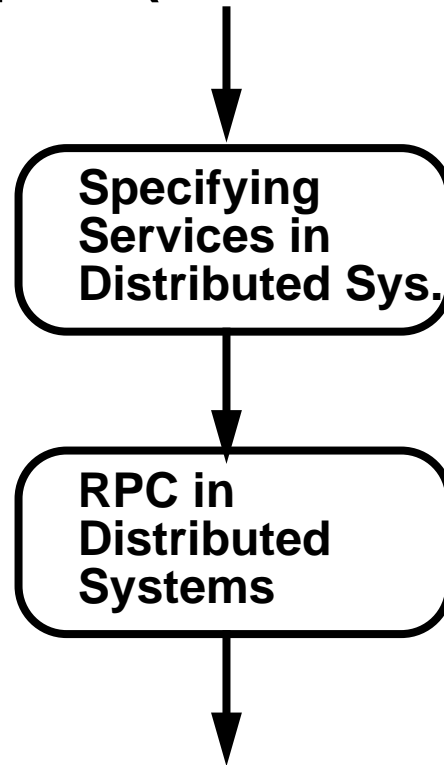
Course Roadmap - 1 (Technical Context)





Building Applications with Distributed Objects

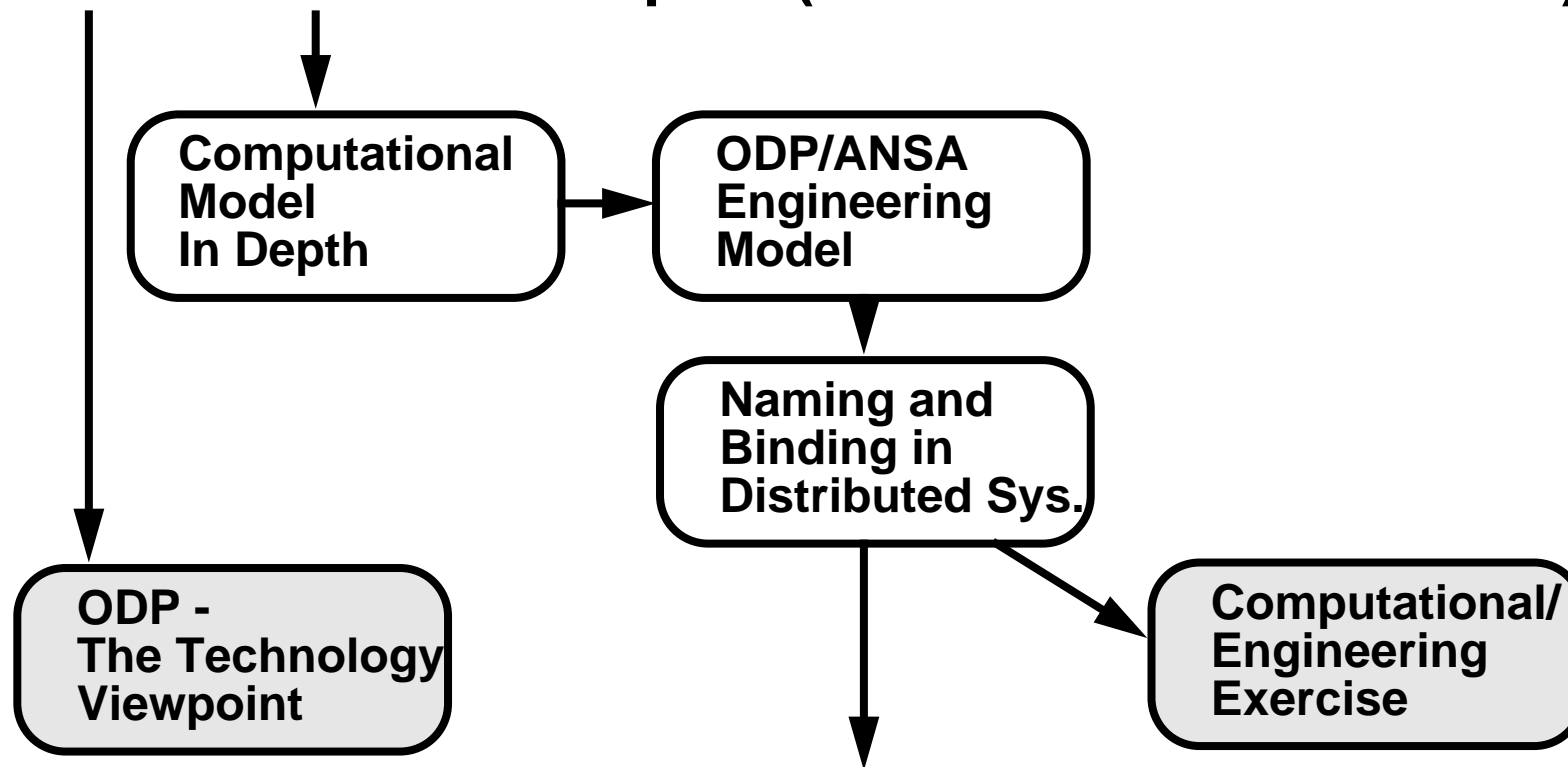
Course Roadmap - 2 (Service Specification and Design)





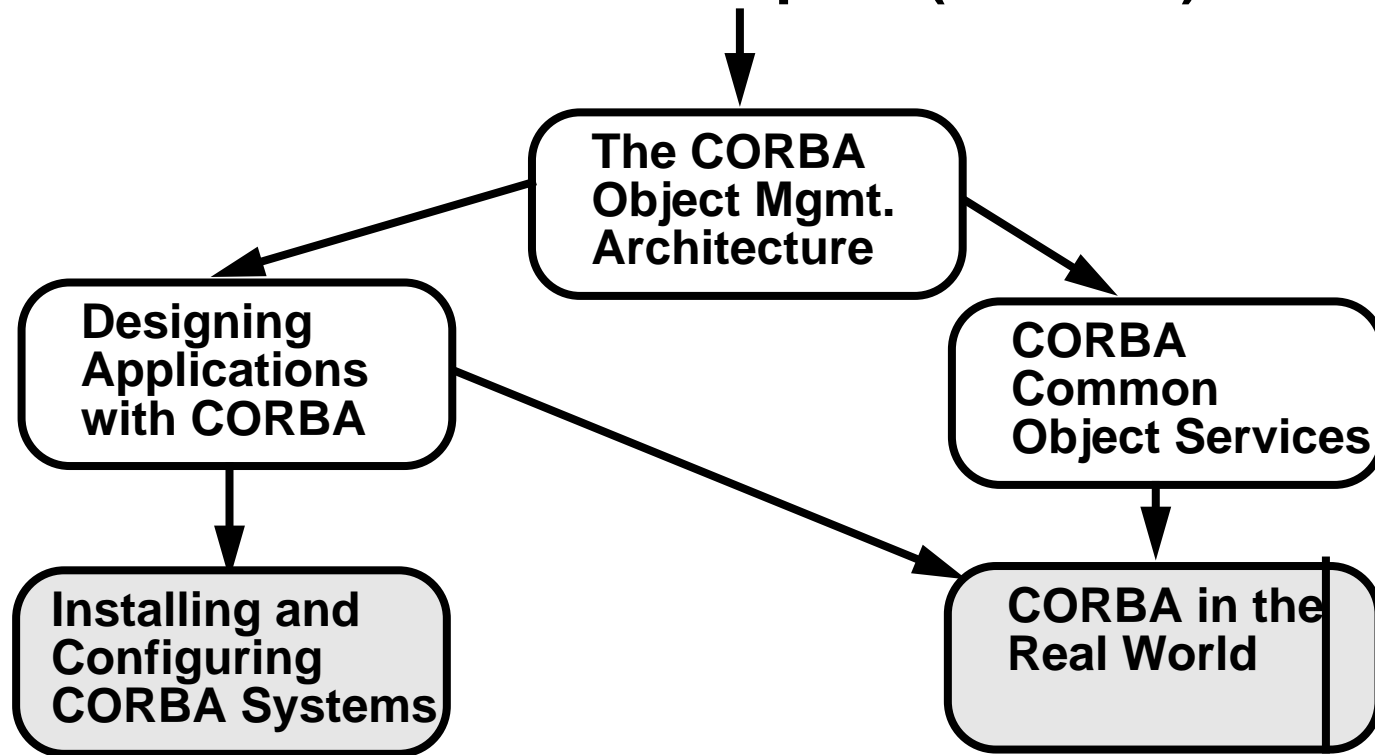
Building Applications with Distributed Objects

Course Roadmap - 3 (Architecture and Models)



Building Applications with Distributed Objects

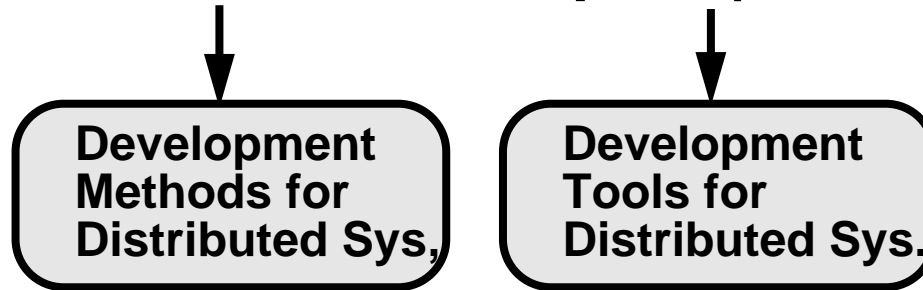
Course Roadmap - 4 (CORBA)





Building Applications with Distributed Objects

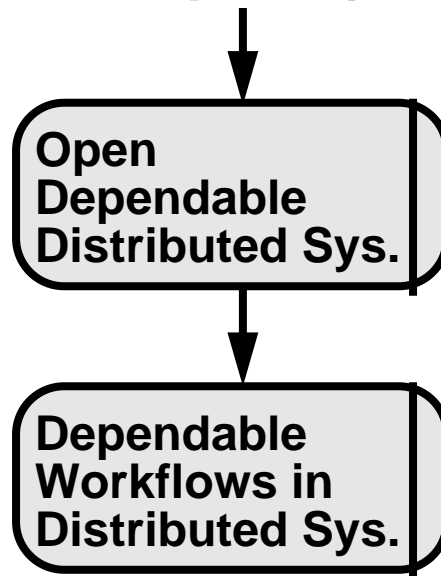
Course Roadmap - 5 (Methods and Tools)





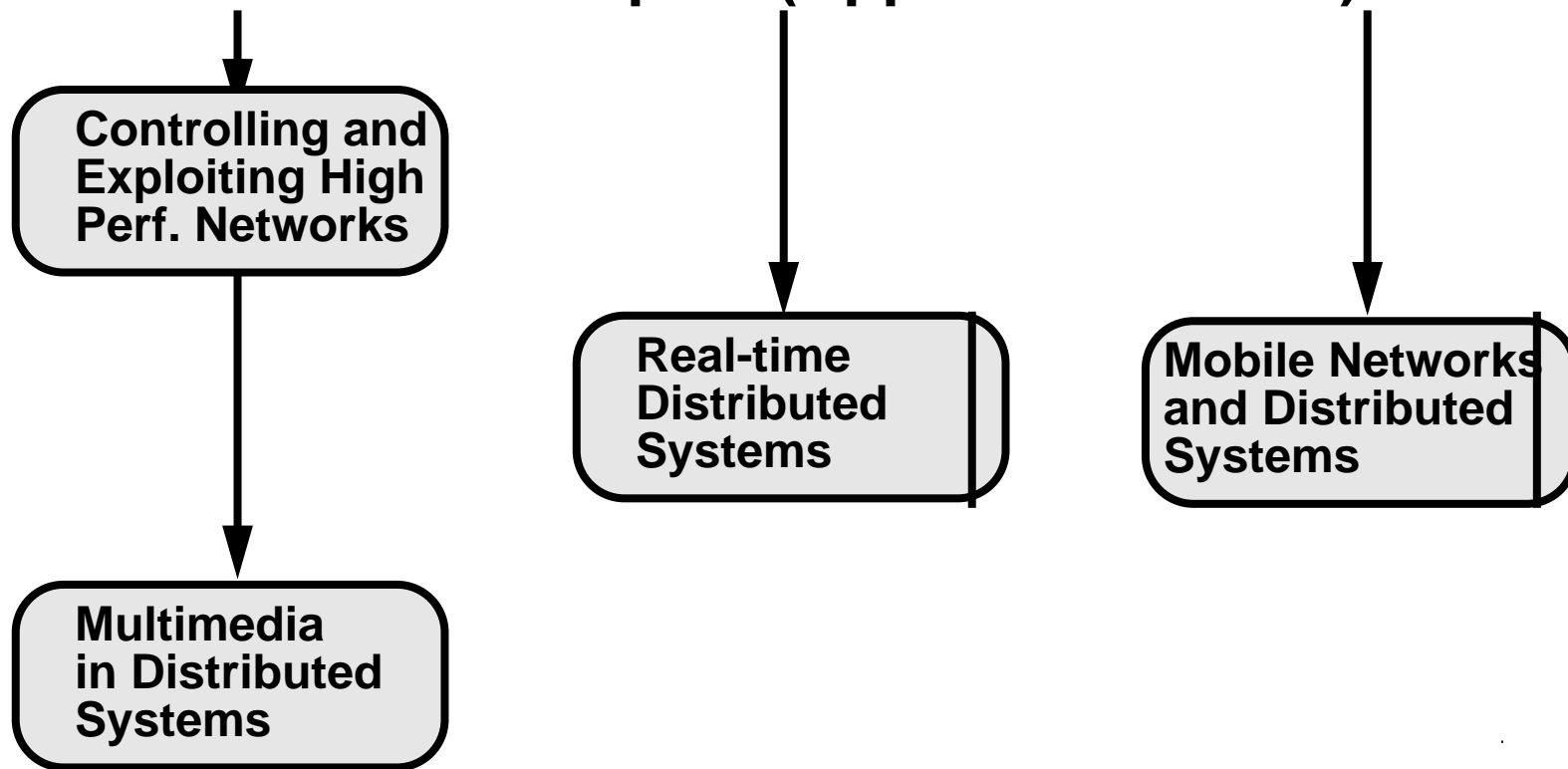
Building Applications with Distributed Objects

Course Roadmap - 6 (Dependability)



Building Applications with Distributed Objects

Course Roadmap - 7 (Application Areas)





ANSAwise - Topics not included

Although a comprehensive programme, ANSAwise cannot cover all aspects of distributed systems. We do not plan to cover these topics in these standard courses; please ask us about course customization

- *Protocol specification and verification (including security protocols); formal specification languages and techniques (e.g. LOTOS)*
- *Distributed algorithms*
- *Distributed file system service design and implementation*
- *Distributed time synchronization service design and implementation*
- *X Window System and distributed user interfaces*
- *Parallel systems and parallel algorithms*
- *Parallel and distributed programming languages (occam, CSP, Ada)*
- *Safety in distributed systems*



Introduction to Distributed Systems

This module briefly covers the business case for distributed systems. The emphasis is on partnership/business integration by overcoming organizational inertia. This module introduces the idea of heterogeneity.

- *The roles of information systems*
- *pressures for change: IS and IT in the 90s*
- *tackling complexity and change*
- *downsizing and rightsizing: legacy systems*
- *growth and evolution*
- *end-to-end processes and business process engineering*



Characteristics of Open Distributed Systems

This module lets the participants discover, through a series of brief paper exercises, the characteristics of open distributed systems. It covers the costs as well as the benefits. It concludes with the idea of reversed assumptions. It does not cover standards or products; these are covered in later modules. This module does not use the idea of “architecture”; this is covered in a later module.

- *today's open systems: how we got here*
- *what do we mean by 'open'?*
- *what do we mean by 'distributed'?*
- *who is involved, and what are the trends?*
- *the advantages of open distributed systems*



Distributed and Networked Operating Systems

This module explains why network operating systems and experimental distributed systems are insufficient for distributed systems needs, although they have useful components and concepts.

- ***LANs and Network Operating Systems (NOS)***
 - PC legacy networks: Microsoft/IBM, Novell
 - Unix and other peer-to-peer systems
 - Mainframe interconnect
- ***New generation operating systems***
 - PC (Windows 95, Apple Mac OS)
 - Micro-kernel (Mach, Chorus, Windows NT)
 - Embedded and special-purpose; support for distribution
- ***New generation hardware platforms: parallel and MPP***
- ***Positioning these into your distributed systems strategy***



Templates for Distributed Applications

This module describes different ways of structuring distributed applications; “what goes where”. It introduces the idea of middleware as distributed systems glue for applications integration. It explains how the LAN has been put to progressively more sophisticated uses. This module does not cover service quality issues for each template; this is covered in a separate module.

- *The LAN and WAN: sharing resources, information, and processes*
- *First-generation and second-generation client-server*
- *Partitioning: user interface, business logic, information management*
- *Client-server and peer-peer templates*
- *Industry-specific initiatives*



Networking in Distributed Systems - Principles and Functions

This module is a review of the networking Lower Layers (1-3), plus the transport layer (4). The emphasis is on the ideas of interoperability, layering, and the diversity of options on each layer; distributed systems ideas per se are not discussed. The different protocols are not discussed in detail. Network management is covered in a separate core module. Protocols for mobility are covered in a separate optional module. Network services are covered in a separate core module. The focus is on network architecture, not technical detail.

- *Reference Models and Layering*
- *Network Architectures and Protocols*
- *LAN and WAN: inter-networking*
- *Broad-band communications*
- *Who's involved - standards and products*



Management of Distributed Networks

This module reviews the concepts and standards of network management. It also covers the management of software (applications, objects), right down to the desktop (where there are many interesting ad-hoc solutions). This module does not cover network management protocols (SNMP, CMIP). This module does not cover the use of distributed systems to monitor and manage a network (AR.010, TR.039, and ANSA Phase III dependability work); this could be covered in a separate module if required. It does not cover TMN. None of the frameworks is covered in detail. Again, these could be offered as separate modules if required.

- *Overview of OSF/DME, OSI/ITU-T, Internet, and IEEE frameworks*
- *Network management and systems management: functional areas*
- *Applications management and management applications*
- *Management of distributed objects*
- *Management of the desktop*
- *Related initiatives (Omnipoint)*



Architecture for Open Distributed Systems

This module explains why an architectural approach is necessary for open distributed systems, and the costs and benefits this implies.

- *architecture as framework*
- *templates, patterns, and reference designs*
- *architecture for interoperability*
- *choice, ownership and control of your architecture*
- *matching architecture to your enterprise*



The ODP Reference Model

This module is an introduction to the Reference Model of Open Distributed Processing. After this module, participants would be able to study Part 1 of RM-ODP independently.

- *Structure of the ODP Reference Model*
- *Key ideas*
- *Viewpoints and models*
- *Relationship to other standards*
- *ODP in action: case studies*
- *Finding out more*



Enterprise Modelling for Distributed Systems

This module explains why enterprise modelling is a valuable approach, and introduces an exercise.

- *Why use (enterprise) modelling for your organization?*
- *Enterprise modelling for complex systems: health-care, telecommunications*
- *Policy and mechanism*
- *Boundaries, communities, and autonomy: contracts*
- *Agents artefacts, and resources*
- *Political, social, economic, and technological approaches*
- *Use cases and scenarios*
- *New techniques and method extensions: EMM/ODP*



Enterprise Modelling Exercise

This module is an exercise in enterprise modelling. It is carried out in syndicate groups of 3 or 4. The focus is firmly on enterprise modelling, but inevitably some information issues will arise too. Notation is also required (this will use the standard ANSA notation unless the customer is already using an enterprise modelling technique).

- *Exercise setting*
- *The AAR technique*
- *Exercise review*



Introduction to CORBA and DCE

This module is a look at two major open standards in terms of capabilities, strengths, weaknesses, and future plans, concentrating on CORBA. This module aims to convey the current state of the industry (ownership and control) rather than the theoretical concepts; these concepts are covered elsewhere. ANSA and ODP are covered elsewhere.

- *The Object Management Group (OMG) and Open Software Foundation (OSF); their visions*
- *The structure of CORBA, and the structure of DCE*
- *Who else is participating (X/Open, SIGs)*
- *Example winning applications*
- *Interoperability: a key issue*
- *Finding out more (publications, contacts, user groups)*
- *Work in progress: the future of CORBA and DCE*



Trading and Federation

This module describes the ODP/ANSA Trading and Federation Model

- *Client and server roles*
- *The need for trading*
- *The interactions in trading*
- *Type conformance*
- *Federation and trading contexts*
- *Trading in ANSA, ODP, and CORBA*



The Computational Model

This module describes the ODP/ANSA Computational Model.

- *Encapsulation and the problem of shared state*
- *Multiple interfaces*
- *Interface types*
- *Service configurations*
- *Interfaces and operations*
- *Type conformance*



Service Quality in Distributed Systems

This module covers how to define requirements for the non-functional characteristics of a system.

It does not cover quality/metrics for the development life-cycle; this will be covered in another optional module. as the need arises.

- *Service Quality and QoS/GOS*
- *QoS frameworks and taxonomy*
- *QoS and trading in distributed systems*
- *QoS in telecommunications*
- *QoS methods and mechanisms*
- *QoS management, metrics, and monitoring*



Security in Distributed Systems

This module introduces basic security concepts, and explains the limitations inherent in most current approaches. This module is not primarily concerned with security planning and cost/benefit (this could be a preliminary module based on BS 7800 and CRAMM). A comparison with (non-distributed) security architectures (e.g. ECMA TR/46, ISO 7498-2) would also be in a separate module. Assessment, certification, and legal issues (e.g. data protection, harmonization) are not covered.

- *Security aims and policies*
- *Controlling the risks (current and future)*
- *What's different about distributed object-based systems security*
- *How objects help*
- *Interoperability*
- *Key standards and their limitations*
- *What's possible now*



Open Information Interchange

This module examines how distributed systems approaches complement the document-centric approach of the desktop operating systems. It also explains why interchange formats and replication strategies are inherently limiting. Document reprographics as an electronic service is also given as an enterprise example.

- *Documents as components: OLE, OpenDoc*
- *Document management and exchange*
- *Document distribution*
- *Hypertext and hypermedia*
- *Standardization and its limits*
- *Distributed systems and electronic publishing*



Services in the Electronic Marketplace

This module explains the technological limits to electronic services, and the new approaches they require. The focus is more on federation than on dependability. Approaches to remuneration are discussed in detail

- *Electronic services: today and tomorrow*
- *Service federation and organizational boundaries*
- *Remuneration*
- *Intelligent agents*



Distributed Workflow Applications

This module concentrates on the applications/enterprise/information aspects of workflow, and explains why distributed systems are a natural match for inherently distributed workflow. Investments in office systems have not yielded expected benefits, and process automation is necessary. It explains why e-mail cannot offer necessary guarantees. Computational and engineering aspects are covered in a separate module.

- *exploiting your investment in office systems*
- *workflow for business process automation: collaborative work*
- *human issues and the enterprise viewpoint: empowerment*
- *workflow and workgroup computing: the end and the means*
- *fundamental workflow activities*
- *workflow for rapid service provision*
- *the distributed systems solution: why e-mail is not enough*



Customer Service - A Distributed Systems Case Study

This module describes a successful transition to a distributed system achieved through a sound management approach backed by ANSA technology and a partnership project. Legacy systems were a key part of the problem.

- *Application: Customer identification and information*
- *Critical success factors (CSFs) and business objectives*
- *Technical objectives: resilience and heterogeneity*
- *System architecture*
- *Future plans*



Concepts of Open Distributed Systems

This module explains the basic concepts of open distributed systems. It relates these concepts to the desired characteristics (e.g. scalability).

- *services*
- *encapsulation and objects*
- *interfaces*
- *client and server roles*
- *trading and federation*
- *transparencies: supporting the characteristics*
- *existing closed systems and compromises*



Planning Your Distributed Systems Strategy

This module identifies the project planning issues in introducing distributed systems as part of an IT strategy.

- *Assessing your current strengths*
- *Buy versus build: reuse?*
- *Entry costs and cost estimation: assessing the benefits and the return*
- *Maintenance, support, and enhancements*
- *Strategy and tactics: what is usable now?*
- *Standards: which and why?*
- *Staffing the project: people and organization*
- *Controlling the risk: internal and external*
- *Trends and directions*



Distributed Communication Techniques

This module compares and contrasts the various inter-process communication paradigms, and explains how they relate to other middleware ideas. Sample applications are given for each. No paradigm is discussed in detail; RPC and RDA are covered in detail in other modules. Continuous media are covered in other modules. distributed file systems

- *distributed messaging and OLTP*
- *distributed function (RPC) and distributed objects*
- *distributed database (RDA)*
- *standardization initiatives*



Controlling and Exploiting High Performance Networks

This module surveys high performance networking standards, concentrating on ATM. The emphasis is not on the networking technology, but what distributed systems can do for networks, and vice-versa. That is, how distributed systems will use network features, and conversely how the networking technology will use distributed systems for management. Wireless technologies are covered in a separate optional module on mobility.

- *Full circle: today's network assumptions*
- *High performance: speed isn't everything*
- *Asynchronous Transfer Mode (ATM): features for distributed systems*
- *Performance specification*
- *Wide Area Network design challenges*



Mobile Networks and Distributed Systems

This module surveys the facilities that mobile networks can provide for distributed systems, and the challenges they offer to distributed applications. Specific mobile data networking products are not discussed.

This is an optional module.

- *The challenge of mobility for distributed systems*
- *Mobile network and transport protocols*
- *Intermittent connections: a key challenge*
- *Transparency in mobile networks*
- *Mobile applications*
- *The future: ubiquitous computing*



Distributed Databases and Distributed Systems

This module explains why distributed databases are insufficient for building distributed systems and applications, and positions the two with respect to each other. Transaction procession (TP, OLTP) is covered in a separate module, and only mentioned briefly. Commercial database products are only used as illustrations. Legacy systems are covered in a separate module

- *the data-centric view and its limitations*
- *distributed databases and their limitations*
- *SQL and the role of database middleware*
- *integrating databases with object systems*



Open Distributed Transaction Processing

This module explains why transaction processing is important, but often misunderstood. A brief paper exercise is planned. Different application examples are included to span the audience's fields.

- *Transaction processing myths*
- *The ACID properties reviewed*
- *Transaction processing for applications*
- *Distributed object-based transaction processing*
- *X/Open and ISO models*
- *Transactions in the real world: extended frameworks*



Objects in Distributed Systems

This module explains the different kinds of object technology available. It delineates the object technology that is relevant to distribution, and introduces the idea of a distributed object. It recaps some of the ANSA Computational Model material in the course. “Understanding Distributed Systems Architecture”). It also mentions some of the pitfalls.

- *The benefits of objects: rapid development, reuse*
- *Object-oriented languages, operating systems, databases,...*
- *Business objects and technology objects*
- *Object concepts: types, inheritance, polymorphism*
- *Objects in a distributed system*
- *Distributed objects: vendor strategies (CORBA, OpenDoc, OLE)*



Specifying Services in Distributed Systems

This module explains what is involved in writing an IDL interface specification, and the typical syntax and semantics involved. APIs are mentioned, but not covered in detail. Other kinds of specification are not covered.

This module briefly introduces the type concepts of ANSA Computational Model (covered in a later module).

This also explains how enterprise specifications (including QoS definitions) could be mapped into a future extended IDL specification. This module may include a paper exercise in writing IDL (CORBA or ANSA), depending on the audience.

- *Interface Definition Languages (IDLs) and APIs*
- *Specifying policies: some examples*
- *Feature interactions and specifications*
- *Organizing object types*
- *Good and bad specifications: pitfalls*
- *Extending IDL*



RPC in Distributed Systems

This module explains in detail the function of the remote procedure call in distributed systems. Note that this module precedes the ANSA Engineering Model module; it is a good idea to cover this material twice. IDL is covered in another module.

- *RPC protocol compilers, stubs, and marshalling*
- *Heterogeneity*
- *The client and server view*
- *Synchronous and asynchronous call: FRPC*
- *Call semantics: at-least-once, at-most once, exactly-once*
- *Idempotence and statelessness*
- *Error handling*
- *RPC federation: OSF/DCE RPC, ANSAware RPC, SunOS RPC*



Development Methods for Distributed Systems

This module explains how it is possible to extend development methods to cope with distributed systems. This module is not prescriptive; it simply aims to prove that existing development methods can be extended, and that new methods are not needed.

- *The vision: a factory for distributed objects*
- *Extending a non-object-oriented method (SSADM/LSDM)*
- *Extending an object-oriented method (OMT)*
- *Business objects: the development life-cycle for reuse*
- *Objects from elsewhere: implications*



Controlling a Distributed Systems Project

This module explains the differences a project manager and team leader will find when managing a distributed systems project. The differences are less significant than the similarities, but do exist; and there are different kinds of dependencies to consider.

- *The balance of work in a distributed systems project*
- *Multi-disciplinary teams for the three system elements*
- *Changing roles*
- *Retraining*
- *Containing the risk*



Development Tools for Distributed Systems

This module explains why efficient development of distributed systems requires new tools and a changed development environment. It also explains how to make the transition technically. The planning issues are covered in a separate module.

- *The development environment for distributed systems*
- *The right tool for the job*
- *Tool support for middleware*
- *Configuration control and repository for reuse*
- *OO tools*
- *GUI builders (frontware)*
- *Testing tools*
- *Groupware: supporting the team*



Naming and Binding in Distributed Systems

This module explains the design issues in an application's use of name services. The approach is comparative, and emphasises the advantages of the ANSA approach. Trading and federation will have been covered in an earlier module; contexts are covered again here. This module does not cover the explicit binding work of ANSA Phase 3; this is covered in a separate module.

- *Directory services: OSI, OSF/DCE, ODP, and CORBA*
- *“White pages” and “Yellow pages” service; the Internet Domain Name Service (DNS)*
- *Early and late binding*
- *Naming, contexts, and trading*
- *Naming and security*
- *Name management and federation*
- *Explicit and implicit binding*



ODP/ANSA - The Engineering Viewpoint

This module describes the ODP/ANSA Engineering model.

- *Engineering as infrastructure*
- *Engineering for trade-offs*
- *Engineering mechanisms*
- *Functions*
- *Transparencies*
- *Resource allocation: policies and mechanisms*
- *Engineering viewpoint compared with the computational viewpoint*



ODP - The Technology Viewpoint

This module explains the purpose of the ODP technology viewpoint, and how it is used to establish testability. The emphasis is on assurance concepts (ISO 9000) rather than customer satisfaction (TQM) concepts. The CORBA conformance also shows the role of X/Open

- *Service quality: convincing the sceptic*
- *Assuring confidence in distributed systems*
- *Compliance and conformance*
- *Conformance reference points*
- *Viewpoint transformation*
- *The assessment process (ICS, PCO, IXIT)*
- *Conformance in CORBA and DCE: a comparison*



The CORBA Object Management Architecture

This module describes the CORBA OMA in detail. The rationale for the OMA has already been given. This module does not cover the CORBA specification itself. Terminology and definitions are dealt with.

- *CORBA Building Blocks - ORB, object services, common facilities, application objects*
- *The Core Object Model*
- *Language mappings*
- *ORB 1.1 and ORB 2 differences; state of play*



Designing Applications with CORBA

This module explains the ORB architecture, and gives some straightforward examples of use. It explains the development process for CORBA-based applications. An exercise may be included, depending on the audience.

- *The CORBA Interface Definition Language (IDL): syntax and semantics*
- *Object identifiers*
- *Types and operations: Substitutability*
- *Example mapping: C language*
- *Interface Repository*
- *Basic Object Adapter*
- *IDL (Interface Definition Language) versus DII (Dynamic Invocation Interface)*



CORBA Common Object Services

This module describes each of the CORBA Common Object Services.

- *General design principles*
- *Naming*
- *Events*
- *Life-cycle*
- *Persistence*
- *Future services, and anticipated changes (COSS 2, 3, 4)*



CORBA in the Real World

This module explains the complexities of putting together a real system, and the decisions you will need to take when deploying a CORBA systems.

This module is available to sponsors only.

- *What you will need: which components from where*
- *Compatibility in the real world: platforms and protocols*
- *Feasible limits*
- *Product maturity*
- *What's missing from CORBA*
- *Right and wrong applications*
- *ORB product survey*



Installing and Configuring CORBA Systems

This module explains the issues to be considered when installing and configuring a CORBA system. It is not tied to a particular CORBA product (see separate module for this information). The performance measurement aspects emphasised in the light of observed CORBA performance.

This module is available to sponsors only.

- *selecting a CORBA product: pricing implications*
- *development, pilot, and production environments*
- *testing and performance measurement (benchmarking)*
- *system configuration: transports and protocols*
- *cut-over and configuration control*
- *upgrading a CORBA system*



Open Dependable Distributed Systems

This module defines dependability, and explains the computational viewpoint and engineering mechanisms to achieve it. This module does not cover security, predictability, transactions, or multimedia; these are covered in separate modules. This module explains group protocols, but not their design and implementation.

This optional module is available only to sponsors. (It might be possible to provide a variant of this module to non-sponsors.)

- *what is dependability?*
- *dependability in business and real-time systems*
- *failure models*
- *distribution for dependability: some advantages, some disadvantages*
- *dependability techniques: replication and interface groups*
- *dependability engineering: transparencies and group protocols*
- *dependability toolkits (ISIS)*



Real-time Distributed Systems

This module explains the distinctive needs of distributed real-time applications, and explains the changes to the Computational and Engineering models to satisfy these needs. Multimedia is covered in a separate module

This is an optional module, available only to sponsors.

- *real-time applications: process control, command-and-control*
- *predictable performance*
- *synchronous computational models*
- *engineering for resource separation*
- *real-time policies: rate-monotonic and deterministic scheduling*
- *real-time protocols*
- *explicit binding*



Multimedia in Distributed Systems

This module explains why multimedia makes special demands on distributed systems. Quality-of-Service is covered here, but only in the context of multimedia.

This is an optional module.

- *distributed multimedia applications: more than video-conferencing*
- *Streams: support for continuous media*
- *demands on the system: real-time orchestration*
- *demands on the network: multi-cast protocols, and QoS*
- *LAN and WAN capabilities: what's possible now*
- *telecomms trends and directions: who pays, who benefits?*



Dependable Workflows in Distributed Systems

This module explains why new architectural components are needed to support dependable workflows. It briefly recaps earlier modules on Open Distributed Transaction Processing. Information Publishing is covered in a separate module. This is an optional module.

- *Architectural components for dependable workflows*
- *Federated workflows*
- *Workflow scripting*
- *Event management*
- *Toolkit for assembling applications*
- *API standards*



Understanding Distributed Systems Architecture DS13/1

Table 1: Day 1 of standard 3-day course (DS13)

Time	Module
9:30	<i>Welcome</i>
9:45	Introduction to Distributed Systems
10:45	Distributed and Networked Operating Systems
11:45	<i>Break</i>
12:00	Characteristics of Open Distributed Systems
13:00	<i>Lunch</i>
14:00	Templates for Distributed Applications
15:00	Architecture for Open Distributed Systems
16:00	<i>Break</i>
16:15	Networking in Distributed Systems
17:15	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate



Understanding Distributed Systems Architecture DS13/2

Table 2: Day 2 of standard 3-day course (DS13)

Time	Module
9:00	<i>Review of Day 1</i>
9:15	Introduction to CORBA and DCE
10:15	The ODP Reference Model
11:15	<i>Break</i>
11:30	Management of Distributed Networks
12:30	<i>Lunch</i>
13:45	Enterprise Modelling for Distributed Systems
14:45	Distributed Databases and Distributed Systems
15:45	<i>Break</i>
16:00	The Computational Model
17:00	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate



Understanding Distributed Systems Architecture DS13/3

Table 3: Day 3 of standard 3-day course (DS13)

Time	Module
9:00	<i>Review of Day 2</i>
9:15	Trading and Federation
10:15	Service Quality in Distributed Systems
11:15	<i>Break</i>
11:30	Distributed Workflow Applications
12:30	<i>Lunch</i>
14:00	Security in Distributed Systems
15:00	Services in the Electronic Marketplace
16:00	Course Roundup
16:30	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate



Understanding Distributed Systems Architecture DS12/1

Table 4: Day 1 of standard 2-day course (DS12)

Time	Module
9:30	<i>Welcome</i>
9:45	Introduction to Distributed Systems
10:45	Distributed and Networked Operating Systems
11:45	<i>Break</i>
12:00	Characteristics of Open Distributed Systems
13:00	<i>Lunch</i>
14:00	Templates for Distributed Applications
15:00	Networking in Distributed Systems
16:00	<i>Break</i>
16:15	Introduction to CORBA and DCE
17:15	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate.



Understanding Distributed Systems Architecture DS12/2

Table 5: Day 2 of standard 2-day course (DS12)

Time	Module
9:00	<i>Review of Day 1</i>
9:15	Trading and Federation
10:15	Management of Distributed Networks
11:15	<i>Break</i>
11:30	The Computational Model
12:30	<i>Lunch</i>
13:45	Distributed Databases and Distributed Systems
14:45	Security in Distributed Systems
15:45	<i>Break</i>
16:00	Services in the Electronic Marketplace
17:00	Course Roundup
17:30	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate.



Building Applications with Distributed Objects DS23/1

Table 6: Day 1 of standard 3-day course (DS23)

Time	Module
9:30	<i>Welcome</i>
9:45	Concepts of Open Distributed Systems
10:45	Objects in Distributed Systems
11:45	<i>Break</i>
12:00	Specifying Services in Distributed Systems
13:00	<i>Lunch</i>
14:00	RPC in Distributed Systems
15:00	Computational Model in Depth
16:00	<i>Break</i>
16:15	ODP - The Technology Viewpoint
17:15	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate.



Building Applications with Distributed Objects DS23/2

Table 7: Day 2 of standard 3-day course (DS23)

Time	Module
9:00	<i>Review of Day 1</i>
9:15	ODP/ANSA Engineering Model
10:15	The CORBA Object Management Architecture
11:15	<i>Break</i>
11:30	Controlling a Distributed Systems Project
12:30	<i>Lunch</i>
13:45	Naming and Binding in Distributed Systems
14:45	Open Dependable Distributed Systems
15:45	<i>Break</i>
16:00	Designing Applications with CORBA
17:00	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate



Building Applications with Distributed Objects DS23/3

Table 8: Day 3 of standard 3-day course (DS23)

Time	Module
9:00	<i>Review of Day 2</i>
9:15	CORBA Common Object Services
10:15	Development Methods for Distributed Systems
11:15	<i>Break</i>
11:30	High Performance Networks
13:00	<i>Lunch</i>
14:00	Real-time Distributed Systems
15:00	Multimedia in Distributed Systems
16:00	Course Roundup
16:30	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate



Building Applications with Distributed Objects DS22/1

Table 9: Day 1 of standard 2-day course (DS22)

Time	Module
9:30	<i>Welcome</i>
9:45	Concepts of Open Distributed Systems
10:45	Objects in Distributed Systems
11:45	<i>Break</i>
12:00	Specifying Services in Distributed Systems
13:00	<i>Lunch</i>
14:00	RPC in Distributed Systems
15:00	Computational Model in Depth
16:00	<i>Break</i>
16:15	The CORBA Object Management Architecture
17:15	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate.



Building Applications with Distributed Objects DS22/2

Table 10: Day 2 of standard 2-day course (DS22)

Time	Module
9:00	<i>Review of Day 1</i>
9:15	ODP/ANSA Engineering Model
10:15	Open Dependable Distributed Systems
11:15	<i>Break</i>
11:30	High Performance Networks
12:30	<i>Lunch</i>
13:45	Designing Applications with CORBA
14:45	CORBA Common Object Services
15:45	<i>Break</i>
16:00	Development Methods for Distributed Systems
17:00	Course Roundup
17:30	<i>Close</i>

This timetable shows a typical running order, and may be altered for customization. All timings are approximate.