



**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 in Distributed Systems

Chris Mayers

Abstract

A training brochure for the Business Unit.

APM.1237.00.03

Draft

5th October 1994

Marketing and Contracts

Distribution:

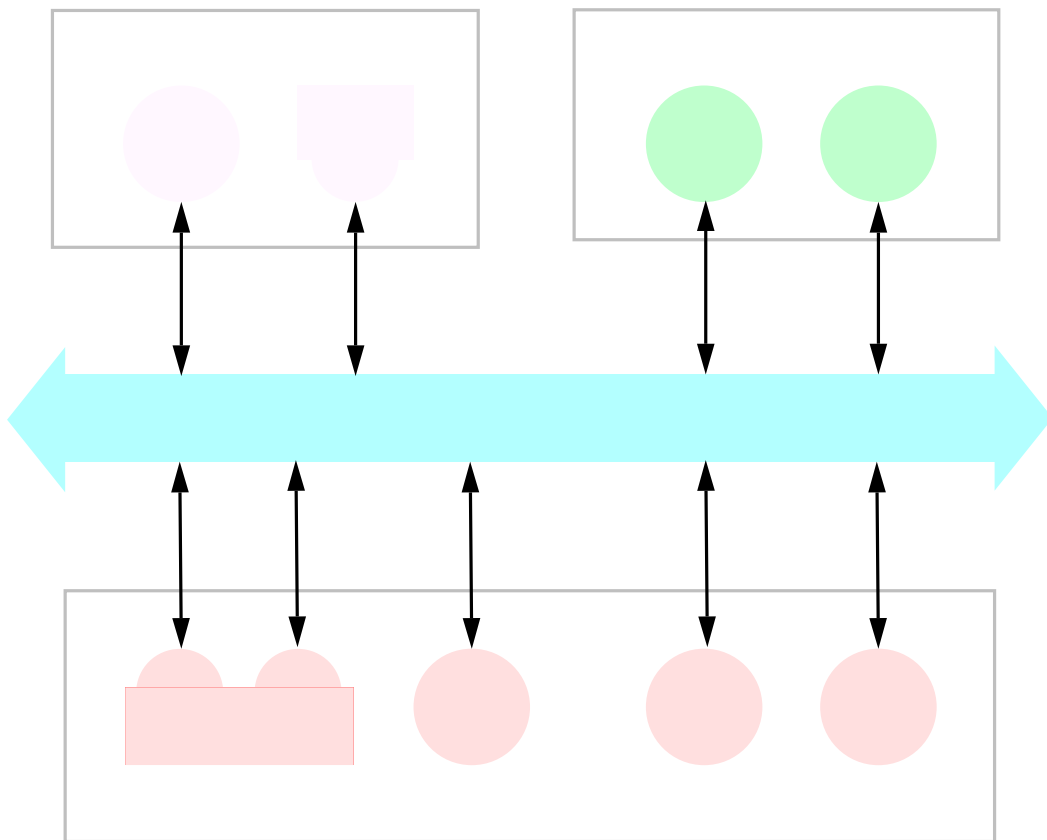
Supersedes:

Superseded by:

Copyright © 1994 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.



Training in Distributed Systems





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

TELEPHONE: Cambridge (0223) 515010
INTERNATIONAL: +44 223 515010
FAX: +44 223 359779
E-MAIL: apm@ansa.co.uk

Building client/server and distributed systems is easy. All you need is a front-end tool - choose one - some network software, and a compatible data-access back-end. And there you are. Or are you? As a demonstration it may work. As a basis for your business, who knows? As a basis for your business tomorrow, doubtful! Recently there has been a spate of reports of client-server applications not achieving their objectives, the issues being scalability, robustness, and compatibility.

Client/server and distributed systems platforms are rapidly moving into the main stream. Many organizations are applying them - with sometimes mixed results.

The key technologies of client-server, such as middleware, CORBA, and DCE cannot be ignored. But any new technology has its potential pitfalls. What are the benefits, costs, and risks (technical and business)? What issues are still unresolved? Who are the key players and what are their plans?

APM has over 8 years experience in the design of distributed systems - spanning both Open and Legacy systems. In fact the ANSA architecture developed by APM, makes no assumption about compatibility or otherwise of your basic systems. Our technology is in use at more than 150 installations.

APM Training focuses on training staff in the issues which are critical to today's business world. Business is not static, and neither is the underlying IT technology. Today's systems must be designed and built - from the outset - with flexibility and adaptability in mind - "openness" in the broadest sense.

Using our technology, customers have developed large-scale systems with over 6000 nodes. These systems have grown smoothly from only a few dozen nodes. All distributed systems should be able to achieve this - including yours.

Course Overview

About ANSAwise

ANSAwise is our name for a modular series of courses in distributed systems, designed to meet your training needs. ANSAwise teaches portable and transferrable skills. It is constructed to be independent of any particular vendor or vendor's products. It is generic and does not assume use of the ANSA architecture.

ANSAwise focuses on object-oriented approaches. It covers all stages in the development cycle from IT strategy through requirements and analysis into design and implementation. Relevant commercially-available products and design methods will be critically examined and discussed.

Case studies and examples are drawn from real-world experience, including telecommunications, financial, and command-and control systems.

The initial courses present overviews and detail appropriate for both IT management and technicians.

The later courses are more technical and provide hands-on experience. During these courses, technicians will construct client-server applications and a concurrent server over a heterogeneous network. These applications are a practical demonstration of the principles embodied in the course.

ANSAwise is presented by leading practitioners of the design and implementation of open distributed systems. It is overseen by Dr. Andrew Herbert, the Technical Director of APM, and an international authority on the design of open distributed systems.

Course structure

ANSAwise has four courses:

- Understanding Distributed Systems Architecture (DS13): a 3-day introductory course
- Building Applications with Distributed Objects (DS23): a second-level 3-day course
- Developing Manageable Distributed Applications (DS33): an advanced, hands-on, 3-day course
- Design and Engineering of Distributed Systems (DS43): an advanced, hands-on 3-day course

Additional courses

We arrange additional courses which cover specific distributed systems topics in depth. These topics include security, multimedia, mobile applications, and telecommunications applications. Please contact us with your specific requirements.

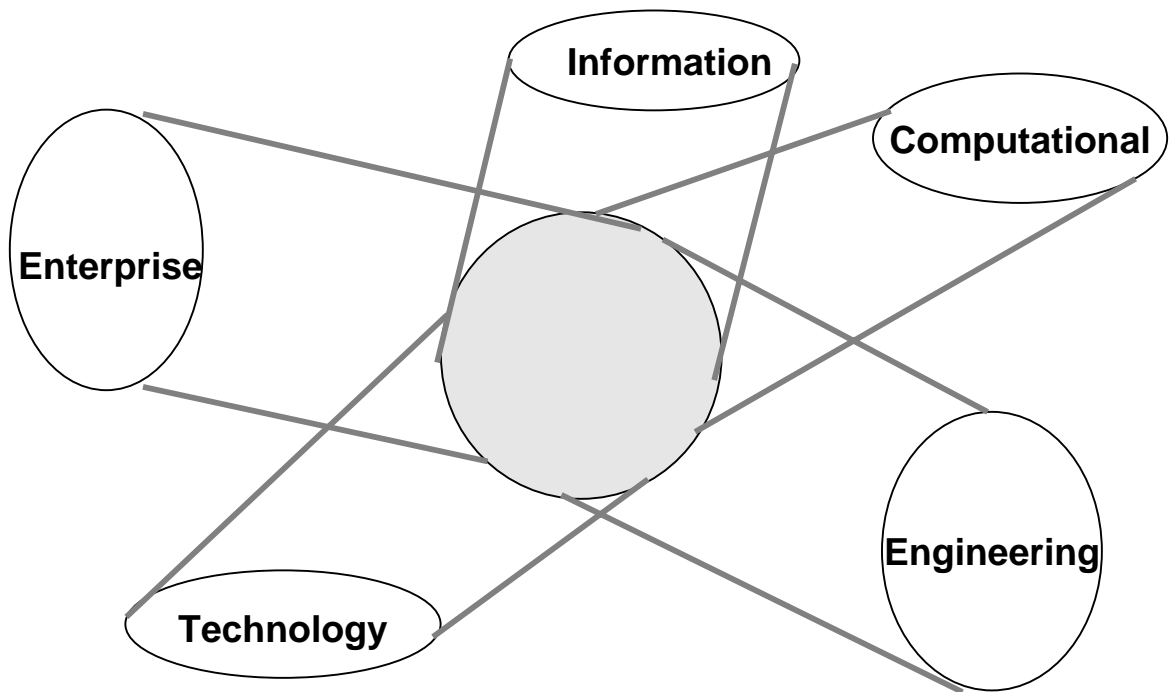
Tailored courses

APM's Business Unit can provide bespoke training, either at Cambridge or at your site. Please contact us with your specific requirements.

For further information

Please contact:

*Business Unit Administrator
Architecture Projects Management Ltd.
Poseidon House
Castle Park
Cambridge CB3 0RD
England
Tel: +44 223 515010
Fax: +44 223 359779
E-mail: apm@ansa.co.uk*



The ODP Reference Model

The five viewpoints separate the concerns of the various stakeholders in a distributed system

Understanding Distributed Systems Architecture

A 3-day introductory course: DS13

About this course

Every business needs to get the most out of their IT investments. What is difficult is to integrate them, to make them work together into an open distributed system which can grow and evolve to meet the needs of your enterprise.

This application integration must preserve the investment in existing legacy systems by seamless integration of applications, balancing demands of change and continuity.

Client/server and distributed systems platforms are rapidly moving into the main stream. Many organizations are applying them - with sometimes mixed results.

There is a strong market move towards distributed multi-vendor systems. What's missing is a coherent approach to IT systems that takes into account all the stakeholders: business management, end-users, IT managers, and IT staff. Similarly, telecommunications systems must satisfy service users, service providers, manufacturers, and regulatory authorities

This coherent approach is called an architecture.

By the end of this course, delegates will have an understanding of the principles and requirements of distributed systems, and an understanding of the consequences of distributed systems design decisions, related to commercial products. They will be able to understand the trade-offs involved in specifying distributed systems for mission-critical applications.

Who will benefit

Senior IT Management, Project Managers, Consultants, System Architects, IT Strategic Planners, System Programmers, anyone wanting to become familiar with concepts of distributed systems.

Topics covered include

Distributed systems and applications integration; typical applications.

Flexibility through diversity and scalability: wins and losses

Client-server, peer-to-peer, and hybrid models

Networking and distributed systems: how they relate.

Middleware and Software Integration Platforms (SIPs): form and function

Cooperating services (trading and federation): objects for sale

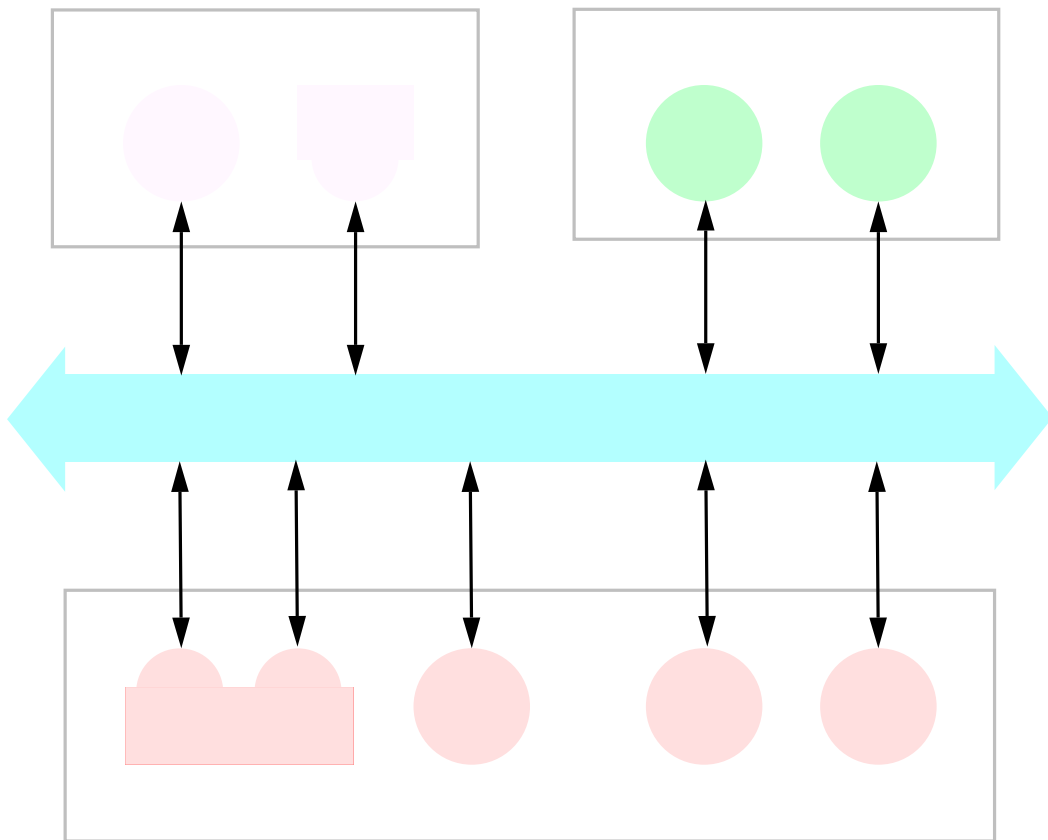
Security and integrity: today's limitations

Distributed systems architectures: examples and comparisons

Integrating legacy systems into distributed applications: preserving investment

Design assumptions and their consequences

Commercial product overview, standards, strengths and weaknesses.



The CORBA Object Management Architecture

The Object Request Broker (ORB) is a 'software bus' that provides clients access to servers. Everything else in the architecture is an object

Building Applications with Distributed Objects

A second-level 3-day course: DS23

About this course

Object-oriented techniques and distributed systems technology are two of the most powerful forces to emerge in the last ten years. But how can they be combined to deliver business services? What new demands do they place on development methods and infrastructures?

What are the market trends, and the strengths and weaknesses of current products? Who controls the key standards, and which are really 'open'?

This course provides a technical view of the central ideas of distributed systems. It clarifies which technology is in actual use, being launched, or in prototype form (the state of the practice, the state of the market, and state of the art.)

Different applications have different needs. This course explains how to choose the right criteria to select appropriate products.

Delegates will acquire the knowledge of the practical tools for constructing open distributed systems, an understanding of the key relevant standards and an awareness of common pitfalls in constructing distributed systems.

Who will benefit

IT Management, Project Managers, Consultants, System Architects, IT Strategic Planners, System Programmers, anyone wanting to become familiar with implementation and methods of building distributed systems.

Topics covered include

How applications operate in a distributed environment.

Tools for constructing open distributed systems

Overview of DCE, CORBA and ODP/ANSA

The programming interface: API and IDL.

Distributed objects and distributed databases

Remote Procedure Calls (RPC), Remote Data Access (RDA), and Reliable Messaging.

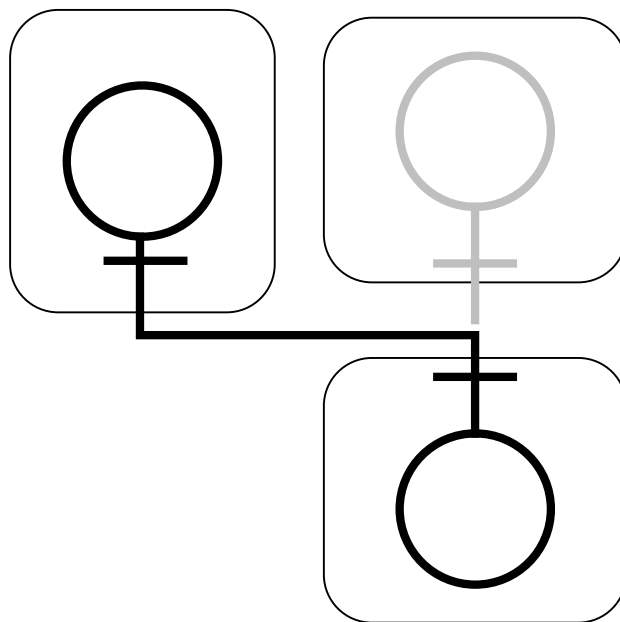
Portability, interoperability, and interchange standards

Object services and transparency mechanisms

Common pitfalls in building distributed systems.

Delegate prerequisites

Delegates must have attended the previous course *Understanding Distributed Systems Architecture*, or already be aware of the material covered by it.



Object relocation

Objects may have to be relocated. With location transparency, users of the object are unaffected

Developing Manageable Distributed Applications

An advanced, hands-on, 3-day course: DS33

About this course

Those who specify distributed systems are faced with many options and decisions. Different applications have different requirements for throughput, availability, and flexibility. Cost-effective, practical implementations mean trade-offs. Which trade-offs are reasonable and what are their implications?

A large part of the cost of a distributed system is the management of the applications that use it. Management interfaces built with object technology reduce that cost.

Applications built from distributed objects offer considerable flexibility. Objects can be upgraded and enhanced separately from the applications that use them. How can this be made transparent to the applications? What systems management tools are available to control this? What are the specific needs of large-scale systems?

This course provides *hands-on* experience in developing applications that tackle these issues. The emphasis is on the pragmatic application of rules and techniques for the construction of distributed systems.

During this course, delegates will develop and test a real-world managed client-server application on a distributed systems platform. Delegates will be able to take with them the code which they have developed during the course.

Who will benefit

IT Specialists, Consultants, Project Managers, System Architects, IT Strategic Planners, System Programmers and Analysts, anyone wanting in-depth knowledge of the construction and principles of open distributed systems.

Topics covered include

Specifying and managing distributed applications

Object types, naming, and binding

IDL specification and use

Transparency mechanisms: selection and use

Management frameworks (for example OSI, SNMP)

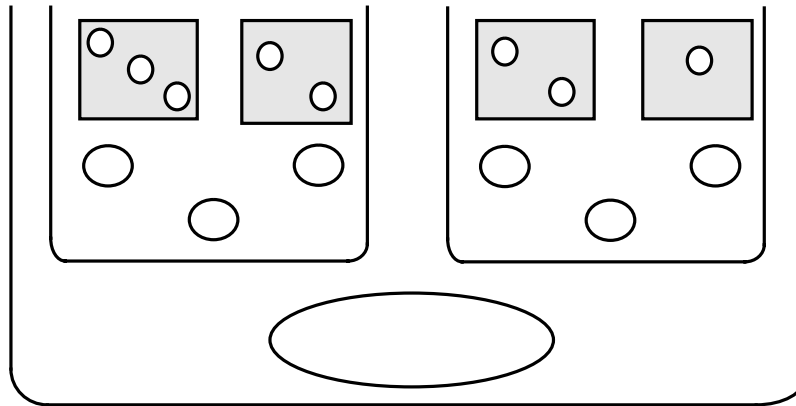
Implementing error handling and recovery strategies

Practical trade-offs and compromises

Testing and debugging distributed applications

Delegate prerequisites

Delegates must be fluent with C/C++. Delegates must have attended the previous course *Building Applications with Distributed Objects*, or be familiar with the material covered by it.



Object Engineering Infrastructure

Within a node, the infrastructure provides inter-process communication. Processes can support many objects.

Design and Engineering of Distributed Systems

An advanced, hands-on, 3-day course: DS43

About this course

Designing and implementing distributed systems poses new technical challenges. Robust objects must maintain tight control over the resources they use. Demanding throughput requirements mean using scalable multi-threaded concurrency techniques.

In the past these engineering techniques were in the specialist domain of the systems programmer. No longer! Nowadays any experienced software engineer must be prepared to employ them. And the system programmer must now cope with complexities of distributed access and multiprocessing.

The traditional programming activities of testing and debugging require new methods. These methods must be applied at the design stage.

This course provides *hands-on* illustrations of typical design problems, and shows how to solve them. It also shows which techniques work for particular systems - and which do not.

During this course delegates will design, build and test a concurrent server. Delegates will be able to take with them the code which they have developed during the course.

Who will benefit

IT Specialists, Consultants, Project Managers, System Architects, IT Strategic Planners, System Programmers and Analysts, anyone wanting in-depth knowledge of the construction and principles of open distributed systems.

Topics covered include

Deriving engineering requirements from a performance specification

Separation of policy and mechanism

Robust use of resources

Controlled memory management

Concurrency specification and control

Processes, tasks and threads

DCE/POSIX pthreads, and other interfaces

Synchronization mechanisms

Systems support for concurrency, (including Unix and Windows NT)

Multiprocessor systems

Testing and debugging of concurrency

Distributed real-time and multimedia systems

Delegate prerequisites

Delegates must have attended the previous course, *Developing Distributed Applications*.

