



**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 Training - case studies and exercises

Chris Mayers

Abstract

Attendees to the ANSAwise training programme require high-quality examples of real-life situations to bring in focus the rest of the course material.

Devising these examples is not easy. There is a lack of real-life case study material to draw on, and many of the academic examples are inappropriate for this audience.

This document summarizes all the significant case studies and exercises that are possibilities for use in the ANSAwise training programme. It will be extended as new possibilities arise.

APM.1401.00.01

Draft

20th January 1995

Briefing Note

Distribution:

Supersedes:

Superseded by:

ANSAwise Training - case studies and exercises



ANSAwise Training - case studies and exercises

Chris Mayers

APM.1401.00.01

20th January 1995

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by Architecture Projects Management Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

Architecture Projects Management Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK
INTERNATIONAL
FAX
E-MAIL

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

Copyright © 1995 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.

Architecture Projects Management Limited takes no responsibility for the consequences of errors or omissions in this Report, nor for any damages resulting from the application of the ideas expressed herein.

Contents

1	1	Introduction
1	1.1	Audience
1	1.2	Scope
1	1.3	Status
2	2	Overview
2	2.1	About this document
2	2.2	Other sources for material
4	3	Simple Bank
4	3.1	Overview
4	3.2	Purpose
4	3.3	Current usage
4	3.4	Source
4	3.5	Comments
5	4	Echo
5	4.1	Overview
5	4.2	Purpose
5	4.3	Current usage
5	4.4	Source
5	4.5	Comments
6	5	Netinfo
6	5.1	Overview
6	5.2	Purpose
6	5.3	Current usage
6	5.4	Source
6	5.5	Comments
7	6	Phone Book
7	6.1	Overview
7	6.2	Purpose
7	6.3	Current usage
7	6.4	Source
7	6.5	Comments
8	7	Appointments Calendar
8	7.1	Overview
8	7.2	Purpose
8	7.3	Current usage
8	7.4	Source
8	7.5	Comments

9	8	Theatre Booking System
9	8.1	Overview
9	8.2	Purpose
9	8.3	Current usage
9	8.4	Source
9	8.5	Comments
10	9	NASA Astrophysics Data System
10	9.1	Overview
10	9.2	Purpose
10	9.3	Current usage
10	9.4	Source
10	9.5	Comments
11	10	Shared Whiteboard
11	10.1	Overview
11	10.2	Purpose
11	10.3	Current usage
11	10.4	Source
11	10.5	Comments
12	11	Hydro-Electric
12	11.1	Overview
12	11.2	Purpose
12	11.3	Current usage
12	11.4	Source
12	11.5	Comments
13	12	DEVETIR
13	12.1	Overview
13	12.2	Purpose
13	12.3	Current usage
13	12.4	Source
13	12.5	Comments
14	13	Road traffic information and management system
14	13.1	Overview
14	13.2	Purpose
14	13.3	Current usage
14	13.4	Source
14	13.5	Comments
15	14	Prison Administration system
15	14.1	Overview
15	14.2	Purpose
15	14.3	Current usage
15	14.4	Source
15	14.5	Comments
16	15	Auction system
16	15.1	Overview
16	15.2	Purpose
16	15.3	Current usage

16	15.4	Source
16	15.5	Comments
17	16	Distributed Card Game
17	16.1	Overview
17	16.2	Purpose
17	16.3	Current usage
17	16.4	Source
17	16.5	Other references
17	16.6	Comments
18	17	Daemon Game
18	17.1	Overview
18	17.2	Purpose
18	17.3	Current usage
18	17.4	Source
18	17.5	Comments
19	18	DEMON
19	18.1	Overview
19	18.2	Purpose
19	18.3	Current usage
19	18.4	Source
19	18.5	Comments
20	19	IN (Intelligent Network) Freephone
20	19.1	Overview
20	19.2	Purpose
20	19.3	Current usage
20	19.4	Source
20	19.5	Comments
21	20	Active Badge Server
21	20.1	Overview
21	20.2	Purpose
21	20.3	Current usage
21	20.4	Source
21	20.5	Comments
22	21	Grapevine name service
22	21.1	Overview
22	21.2	Purpose
22	21.3	Current usage
22	21.4	Source
22	21.5	Comments
23	22	Internet Domain Name Service (DNS)
23	22.1	Overview
23	22.2	Purpose
23	22.3	Current usage
23	22.4	Source
23	22.5	Comments

24	23	Distributed File System
24	23.1	Overview
24	23.2	Purpose
24	23.3	Current usage
24	23.4	Source
24	23.5	Comments
25	24	Virtual Private Network (VPN)
25	24.1	Overview
25	24.2	Purpose
25	24.3	Current usage
25	24.4	Source
25	24.5	Comments
26	25	Video-on-demand (VoD)
26	25.1	Overview
26	25.2	Purpose
26	25.3	Current usage
26	25.4	Source
26	25.5	Comments
27	26	Video Mail
27	26.1	Overview
27	26.2	Purpose
27	26.3	Current usage
27	26.4	Source
27	26.5	Comments
28	27	Remote Temperature Sensor
28	27.1	Overview
28	27.2	Purpose
28	27.3	Current usage
28	27.4	Source
28	27.5	Comments
29	28	Vending Machine
29	28.1	Overview
29	28.2	Purpose
29	28.3	Current usage
29	28.4	Source
29	28.5	Comments
30	29	Lift/Elevator Control System
30	29.1	Overview
30	29.2	Purpose
30	29.3	Current usage
30	29.4	Source
30	29.5	Comments
32	30	Airport Shuttle System
32	30.1	Overview
32	30.2	Purpose
32	30.3	Current usage

32	30.4	Source
32	30.5	Comments
33	31	Simple Railway System
33	31.1	Overview
33	31.2	Purpose
33	31.3	Current usage
33	31.4	Source
33	31.5	Other references
33	31.6	Comments
34	32	Prussian Generals Problem
34	32.1	Overview
34	32.2	Purpose
34	32.3	Current usage
34	32.4	Source
34	32.5	Comments
35	33	Two-way channel with disconnect
35	33.1	Overview
35	33.2	Purpose
35	33.3	Current usage
35	33.4	Source
35	33.5	Comments
36	34	Simple network service
36	34.1	Overview
36	34.2	Purpose
36	34.3	Current usage
36	34.4	Source
36	34.5	Comments
37	35	Mixing synchronous with asynchronous input
37	35.1	Overview
37	35.2	Purpose
37	35.3	Current usage
37	35.4	Source
37	35.5	Comments
38	36	Ray tracing
38	36.1	Overview
38	36.2	Purpose
38	36.3	Current usage
38	36.4	Source
38	36.5	Comments

1 Introduction

1.1 Audience

APM Training

1.2 Scope

This document covers all courses in the ANSAwise training programme, including customized and public courses.

1.3 Status

This is a draft document for review.

2 Overview

2.1 About this document

The following chapters of this document are case studies and exercises that have been considered for use in the ANSAwise training programme, including those that have been rejected. The exercises include paper-only desk exercises and hands-on exercises.

The list naturally overlaps with case study marketing material supplied by APM.

Generally speaking, all the case studies and exercises must have a distribution aspect to them, and should focus on applications design (rather than infrastructure design).

In CORBA terms, any examples of service designs should be CORBA Application Objects, rather than Common Facilities or Object Services.

Note: Perhaps include all demonstrations in here too?

Note: The focus here is on the specific rather than generic. So, a a specific application of the Management Engine would be ideal.

2.2 Other sources for material

The following need to be checked:

- other training courses (and academic computing courses)

Note: Check recent Cambridge Dip Comp Sci, and also the other universities with which we have close links

- ANSA Phase 3 material (particularly scenarios); also ANSA ARs/TRs; also old ANSA documents?
- demonstration, benchmark, and sample applications in distributed systems products (CORBA, DCE)

Note: Check here with mdrb on CORBA products

- examples in books on distributed systems products (CORBA, DCE, distributed OLE)
- examples in books on databases, and transaction processing

Note: there is probably material Jim Gray's book

- examples in distributed systems specifications (CORBA, DCE)
- examples in (OO) programming/design/analysis books and journal/conference papers (practice and experience journals? MSJ?)

Note: see OMT book. Jacobsen's book has a telephony call control example. OO Brewery full paper? Also general software engineering books (e.g. Programming Pearls)

- shareware networked applications

Note: And those bundled with (for example) compilers (e.g. WSOCK). Windows Sockets applications? WORMHOLE? Also those provided with Distributed OLE? Also anything on ftp.microsoft.com?

- programming competitions
- public case study presentations (e.g. from VISUAL)
- ANSA project members personal experience and previous projects

Note: Check my filing pile for any example I might have forgotten about

3 Simple Bank

3.1 Overview

An example application; a centralized network of bank cashpoint machines

3.2 Purpose

Objects, multiple interfaces.

3.3 Current usage

Used in all ANSAware courses, including [Writing Distributed Applications using ANSA].

Also described in outline in [CORBA in the Real World].

3.4 Source

Supplied with ANSAware, and described in detail in [Application Programming in ANSAware].

At ICL, mdrb also ported Simple Bank to DCE.

A slightly different specification also appears as Problem 9 in [Denvir et al.], but there were no solutions offered. [Orbix] has a similar example.

Note: Check with mdrb whether Orbix has sample source code

3.5 Comments

The most detailed example in ANSAware.

ANSAware includes a GUI (X) front-end (but unfortunately, not one for Microsoft Windows).

This example needs care when explaining it, because of different banking terminology in different countries (e.g 'current account'/'checking account'). This makes it rather tiresome to prepare slides.

[Coulouris et al.] says:

“[Bacon 1993] gives a useful summary of a typical design for such a system. It is a highly specific solution, requiring the development of many levels of application-specific software support. This approach has been employed successfully in many commercial applications, but it suffers from limitations that restrict the scope for system growth, reliability, and security.”

4 Echo

4.1 Overview

An example service; echoes back a string sent to it

4.2 Purpose

About the simplest possible client-server application.

Shows:

- simple example of IDL
- ANSAware 'managed service' (using factory and node manager services)

Also used as a test program for ANSAware installations.

4.3 Current usage

Used in all ANSAware courses, including [Writing Distributed Applications using ANSA].

Also used in [Specifying Services] as a comparison between CORBA, DCE, and ANSAware IDL.

4.4 Source

Supplied with ANSAware, and described in detail in [Application Programming in ANSAware].

Note: Check whether mdrb did a DCE version, for comparison

4.5 Comments

Echo has not been ported to Microsoft Windows.

It makes a pleasant change from Hello World. (Unwisely, the [Introduction to DCE] includes a DCE Hello World program. It goes on for several pages...)

5 Netinfo

5.1 Overview

Example service; simple name lookup service providing a RPC interface to the TCP/IP sockets `get*byname()` interfaces.

5.2 Purpose

Shows:

- example of name service
- example of ANSAware IDL CHOICE construct

Also used as a test program for ANSAware installations.

5.3 Current usage

None.

5.4 Source

Supplied with ANSAware, but only briefly mentioned in [ANSAware System Manager's Guide].

5.5 Comments

Netinfo has not been ported to Microsoft Windows.

Netinfo is no longer an ideal server example in itself, because most sites already run the Internet Domain Name Service (DNS) which provides these name lookup facilities already. The Windows Sockets interface provides the same facilities from Microsoft Windows. See also comments on the Internet Domain Name Service (DNS) as an example below

The Windows Sockets Sample Application (WSOCK) in Microsoft Visual C++ is a client example using `gethostbyname()`. The Windows Sockets Test program is a comprehensive client example.

Given the interest in network/system management from CNET, it might be worth making this into a more substantial example. This is the territory being fought over by Microsoft SMS (codename Hermes), and Novell.

6 Phone Book

6.1 Overview

Example service; a simple phone book.

6.2 Purpose

Shows:

- RPC-style access to a flat-file database (alternative to RDA/SQL approaches, simple illustration of approach to 'wrapping a legacy file format')
- simple concurrency control
- use of ANSAware Storage facilities and the ANSA Storage model and migration

Note: Check how good the error handling is...

6.3 Current usage

None.

6.4 Source

Supplied with ANSAware, but only briefly mentioned in [ANSAware System Manager's Guide].

6.5 Comments

Not ported to Microsoft Windows.

Uses the ANSA factory.

Output from ISA project; README implies that the version supplied with ANSAware has bugs.

Work on the ANSA storage model was never written up. Perhaps now overtaken by Arjuna?

Does not use replication or support database recovery.

Maybe too simple compared with the PC tool-based environments (e.g. Powerbuilder)?

Might be more convincing if it interfaced with an industry-standard file format (e.g. delimited text/CSV). Also if there was a GUI front-end.

7 Appointments Calendar

7.1 Overview

Example application; an appointments calendaring system.

7.2 Purpose

Shows:

- Use of name server
- Concurrency control

7.3 Current usage

None.

7.4 Source

Supplied as part of Arjuna, which is part of the standard ANSAware tape. It is in contrib/Arjuna/Applications/Calendar

7.5 Comments

Not ported to Microsoft Windows (and no plans to support Arjuna on Microsoft Windows - and the software is not owned by APM.)

From a quick glance at the source, it does not appear to have a GUI (X) interface. No documentation supplied, and we have not tested it.

The combination of concurrency control and persistence is not by itself particularly interfacing (any shareware PC database will give you this). What is interesting is that you get location transparency and access transparency

Calendaring systems are widespread; some people have negative feelings about them which might colour their views on this example.

8 Theatre Booking System

8.1 Overview

Example application; a theatre booking system.

8.2 Purpose

Shows:

- Concurrency control
- Persistence

8.3 Current usage

None.

8.4 Source

Supplied as part of Arjuna, which is part of the standard ANSAware tape. It is in contrib/Arjuna/Applications/Theatre

8.5 Comments

Booking systems are a staple example on all kinds of computer training courses. Theatre booking is perhaps safer than car rental, airline reservations, hotel reservations, and training course booking systems...

Not ported to Microsoft Windows (and no plans to support Arjuna on Microsoft Windows - and the software is owned by APM.)

From a quick glance at the source, it does appear to have a GUI (X) interface. No documentation supplied, and we have not tested it.

The combination of concurrency control and persistence is not by itself particularly interfacing (any shareware PC database will give you this). What is interesting is that you get location transparency and access transparency too.

9 NASA Astrophysics Data System

9.1 Overview

Case study of legacy system and data integration on a very large scale

9.2 Purpose

ANSA/APM marketing/promotional; major project.

9.3 Current usage

Check ajh's and dme's marketing material
Not currently used in training courses.

9.4 Source

See the [NASA ADS]. Note that we are no longer using this brochure as marketing material.
We do have a couple of pages of example source code and some diagrams.

9.5 Comments

We seem to have individual scattered slides, not all in electronic form. Some of these supplied by NASA are beginning to fade and crack; they are not in colour.
The design and implementation was done by Ellery Systems.
See dme for paper file of background information.

10 Shared Whiteboard

10.1 Overview

A research application consisting of a shared whiteboard on which users can scribble, leave notes, and so on; each user sees the same image - WYSIWIS (what you see is what I see).

10.2 Purpose

This application was built to compare the infrastructure support and programming support provided by different distribution infrastructures, including ANSAware.

Shows:

- Use of groups for reliability through replication
- Causality and concurrency issues

10.3 Current usage

The paper [O'Connell et al.] is a handout on the 'Building Applications with Distributed Objects' course; see [Replication Techniques].

10.4 Source

Described in [O'Connell et al.]. This work was carried out at HP Labs.

10.5 Comments

The ANSAware version used ANSAware 3.0, which did not have group support. It is therefore not a particularly good example for ANSAware.

If we wanted to consider using the source code, we would have to approach HP.

We don't know what the user interface would look like.

Perhaps somewhat futuristic for commercial courses, although there are already products like this already being sold (both as part of video-conferencing products, and as stand-alone PC products).

The current usage doesn't cover the other comparisons in [O'Connell et al.]. For instance, the comparison of object lifecycle approaches is relevant to CORBA Object Services issues.

11 Hydro-Electric

11.1 Overview

A case study for customer service application, integrating computing and telecommunications

11.2 Purpose

Overview of the OASIS project based on DAIS.

11.3 Current usage

Diagram in [Catching Up With the Future].

11.4 Source

See [Middleware for ODP] for a very few (but crisp) slides and sales script.

See also Hydro-Electric single sheet in DAIS information pack.

dme suggests contacting Graham Hook of ICL for more detail.

Was presented jointly by David Hutcheson and Tony Drahota at ANSAworks '94 (see paper copies of slides). Also mentioned in [Legacy Systems] presented at ANSAworks '94.

11.5 Comments

No real technical details yet available.

See dme for paper file of background information.

iam wrote this one up as a case study.

Perhaps work in a groupware angle on this?

12 DEVETIR

12.1 Overview

A case study of business process re-engineering at the Department of Employment, Vocational Engineering, Training, and Industrial relations of Queensland, Australia.

12.2 Purpose

To show an Open Systems based architecture is practical and realistic, in achieving optimum resource utilisation, integration of systems, responsiveness to change, and database rationalization. To report on the lessons learned.

12.3 Current usage

None.

12.4 Source

Via ICL. See [DEVETIR]. The Prototyping section shows the common technical architecture based on ANSA, but the main focus of this paper is the business issues.

See also DEVETIR single sheet in DAIS information pack.

dme suggests contacting Graham Hook of ICL for more detail.

12.5 Comments

The DEVETIR prototype included the ProcessWise Workbench. If we do further sessions on workflow, this may be relevant.

See dme for paper file of background information. This case study was assigned to di; it is not known if the write-up was completed.

13 Road traffic information and management system

13.1 Overview

A demonstration of integrated road traffic information and management, via a simulator.

13.2 Purpose

Demonstration of MOTOS project work.

13.3 Current usage

Promotional.

13.4 Source

MOTOS consortium partners.

13.5 Comments

The demonstration was complex to set up and configure, requiring several bodies from the MOTOS consortium partners. It also required some software/hardware being brought by the partners.

A (largely unrelated) part of the demonstration was of WWW technology, see [Searching for traffic data].

14 Prison Administration system

14.1 Overview

Note: See mdrb

14.2 Purpose

Demonstration of DAIS product.

14.3 Current usage

None.

14.4 Source

ICL DAIS demonstration - see mdrb.

14.5 Comments

The British prison service had a bad time during 1994...!

15 Auction system

15.1 Overview

A demonstration application based on an auction.

15.2 Purpose

Demonstration of DEC's ACAS/ObjectBroker product

15.3 Current usage

None.

15.4 Source

Has been seen by some people at BT Labs, but not by anyone at APM.

Note: Check with mdrb whether it's included with the evaluation of ObjectBroker

Note: There may be some details in a marketing information pack

15.5 Comments

Keep an eye open for this demonstration at exhibitions (e.g. Object World).

16 Distributed Card Game

16.1 Overview

Mentioned in the Objects in Europe trade paper:

“At San Francisco’s Object World Exhibition, Sun IONA Technologies, Postmodern Computing, and other delighted visitors with a distributed multimedia card game on a specially constructed network of Unix and Windows machines.”

16.2 Purpose

A CORBA interoperability demonstration

16.3 Current usage

16.4 Source

Unknown - but Iona use it, apparently. Try and see if we can get access to it.

16.5 Other references

16.6 Comments

Would have to have a convincing GUI. (People’s expectations of game demos are influenced by the consumer video games market.)

The networked Hearts program supplied with Microsoft Windows for Workgroups is a slick implementation, against which any distributed card game demonstration could be judged. (It uses network DDE as the networking mechanism, but I don’t think the Hearts DDE interface is published.)

17 Daemon Game

17.1 Overview

An exercise in designing for a simple multi-player game in which players have to guess whether a counter value is odd or even.

17.2 Purpose

Original purpose was to contrast specifications (formal description techniques) in SDL, LOTOS and Estelle.

17.3 Current usage

In [Designing Applications with CORBA], it is a paper exercise in:

- writing simple IDL
- considering object boundaries and their implications for failure and activation

17.4 Source

[ISO/IEC TR 10167], modified in [Designing Applications with CORBA].

17.5 Comments

Worked well on public course (10 people); not so well at BT (25 people). The game itself is very dull, but this did not seem to matter.

The write-up could be improved, and a worked solution should be provided.

Entirely unrelated to the DEMON project (see below).

18 DEMON

18.1 Overview

18.2 Purpose

18.3 Current usage

18.4 Source

Esprit visualization project; part of the work done by MARI?
ajh suggests discussing this with yh

18.5 Comments

19 IN (Intelligent Network) Freephone

19.1 Overview

A demonstration consisting of a simulated IN-based Freephone network, with integrated call control, billing and charging. Simulates both basic and enhanced Freephone service.

19.2 Purpose

A demonstrator scenario of the DOMAINS management architecture.

19.3 Current usage

None.

19.4 Source

A demonstrator produced as part of the DOMAINS project (Esprit Project 5165), task 4. Part of this was implemented in ANSAware and tcl/tk by rtor. The source is available; see rtor.

See [Freephone Demonstrator Notes], also other contents of DOMAINS paper file.

Note: Check if there was an old one-page marketing flyer on this DOMAINS demonstrator

19.5 Comments

The history of the project is worth discussing with rtor.

There is a fair amount of background material here on Freephone/telephony. But there is some doubt about whether the telephony call model here is accurate; and IN specifications and thinking may have evolved in the last 3 years.

The material here is worth trawling for pieces for relevance to telecomms customers (e.g. fragments of ANSAware IDL).

The separation of Service Subscriber, Service Provider, and Network Operator may be a helpful example for Enterprise modelling issues.

The demonstrator is a substantial ANSAware application.

20 Active Badge Server

20.1 Overview

A research application consisting of a naming database for an active badge system.

20.2 Purpose

Shows:

- Mobility
- Persistence/Storage
- Concurrency specification

20.3 Current usage

None.

20.4 Source

Zihue Wu's Phd thesis (~ajh/misc/old/phd/wu-thesis/*.ps.Z), Chapter 9; suggested by ajh. This includes sample source code.

20.5 Comments

This thesis appears to be mainly concerned with persistent extensions to C++ (PC++). This also provides location transparency. It does not support an exception handling mechanism.

The concurrency specification (through conflict relations) is not the same as in the ANSA Concurrency Model.

Active Badge used to use ANSAware, but no longer?

Yet another naming service.

Too futuristic for a commercial course.

21 Grapevine name service

21.1 Overview

A research service; a distributed, replicated application-orientated database service, incorporating a name service

21.2 Purpose

Shows naming issues

21.3 Current usage

None.

21.4 Source

ajh suggests looking at Mike Schroeder's original paper.

Mentioned in [Mullender], in a chapter by Roger Needham which is a good summary of naming issues.

21.5 Comments

Yet another name service.

Now rather old (1982), but sound.

22 Internet Domain Name Service (DNS)

22.1 Overview

The Internet Domain Name service

22.2 Purpose

Arguably the most widely-used distributed service, but hardly the stuff to fire the blood...

22.3 Current usage

None.

22.4 Source

Note: RFC TBD

Note: O'Reilly book

22.5 Comments

Not an application service.

Normally used via APIs rather than RPC.

23 Distributed File System

23.1 Overview

Example service; distributed file system

23.2 Purpose

ajh says:

“A good example to test distribution is a simple replicated file system in which data is written to and read from multiple files. Files are treated as a sequence of blocks. This involves the programmer in a lot of thread bashing and synchronization. It requires some thought about caching, logging and recovery. There are obvious direct solutions and scope for optimization. It can encourage the student to look at the dependable file server literature. The lecturer can provide a “flaky” server and a benchmark client. You can offer a prize to the student who runs the benchmark fastest, without error. The smart student will implement a log based file system with lots of client caching.”

23.3 Current usage

None.

23.4 Source

See also [Mullender]

See [Coulouris et al.], which has one complete chapter on a model for a file service, and another chapter on case studies of the Sun Network File System (NFS), Andrew File System (AFS), and Coda File System. The exercises at the end of each chapter look relevant, and refer back into each chapter for the solutions.

23.5 Comments

This is about designing distributed file systems; usually perceived as a systems programming exercise. It might be rather difficult, even for an advanced course.

Implementation would be impractical.

Perhaps inappropriate, because it is unlikely that anyone other than researchers or the very few OS vendors would be likely to implement it.

Inappropriate for a commercial course anyway

24 Virtual Private Network (VPN)

24.1 Overview

Case study; virtual private network

24.2 Purpose

Interesting illustration of enterprise and network/system management issues.

24.3 Current usage

None.

24.4 Source

Note: APM scenarios?

Note: ROSA?

Note: Cassiopeia main example?

24.5 Comments

Some demand from ANSA telecomms sponsors for an example here (BT asked for one based on call control). Would require quite a lot of telecomms knowledge.

We have no implementation knowledge.

25 Video-on-demand (VoD)

25.1 Overview

Case study; video-on-demand

25.2 Purpose

Shows:

- real-time issues
- federation (billing/charging issues)

25.3 Current usage

None.

25.4 Source

See [Information Model for Federation]

Note: Other federation references? Also work done by ECP (see rvdI)?

25.5 Comments

A much-debated application for distributed multimedia.

Much of this work is part of ANSA Phase 3.

Too futuristic for a commercial course.

26 Video Mail

26.1 Overview

Case study; video mail service

26.2 Purpose

Shows:

- multimedia issues
- enterprise issues

26.3 Current usage

None

26.4 Source

Note: I have a TINA '93 paper on this somewhere...

26.5 Comments

Too futuristic for a commercial course.

27 Remote Temperature Sensor

27.1 Overview

Example application: sensing the temperature of a vessel in a production plant.

27.2 Purpose

Shows:

- need for real-time response
- use of replication (to take average of N sensor readings)

27.3 Current usage

None.

27.4 Source

Note: In some OO paper?

27.5 Comments

A good example of a naturally distributed system.
Real-life system would have a safety angle.
APM does not have application knowledge.

28 Vending Machine

28.1 Overview

Example application; food or drink vending machine

28.2 Purpose

Shows

- sequencing/state machine design
- error handling

28.3 Current usage

None

28.4 Source

A common example in embedded systems/structured design courses.

28.5 Comments

Doesn't have distributed systems angle (unless you include remote access, for example, the infamous Internet Coke Machine!)

Inappropriate for a commercial course?

29 Lift/Elevator Control System

29.1 Overview

Example application; a building with M floors and N lifts/elevators. The exercise is to design all the software to control the buttons that summon the lifts/elevators, schedule them without starvation, and so.

29.2 Purpose

Shows:

- concurrency
- scheduling

ajh says:

“Can’t offer you a detailed description of the solution. However the scheduling aspect is the same as for disc arm movement and the standard heuristic is keep travelling in the same direction until you have no more requests in that direction, then reverse. This avoids starvation, which could happen if you went to the nearest request every time.”

29.3 Current usage

None.

29.4 Source

A famous example used in computer science courses

See also [Guidon et al.]; this is a study of how designers tackle complex design problems and notes the specific difficulties they encounter. An Appendix contains the problem description.

29.5 Comments

A correct, efficient, and distributed solution requires considerable care. See [Guidon et al.], which shows how software engineers approach this problem. We do not appear to have access to such a solution.

It may be too difficult a problem for a course like this.

dme says that ICL ran this as a competition once (with no results). He worked on it himself, and thinks that UCL and Loughborough also did. Bill O’Riordan (ICL Chief of Research) would be the contact.

Real-life lift/elevator systems also have the safety aspect.

**This example also suffers from the 'lift'/elevator' terminology problem.
Inappropriate for a commercial course?**

30 Airport Shuttle System

30.1 Overview

An exercise consisting of a public transportation system having 4 passenger terminals connected by a railway network served by up to 3 trains.

30.2 Purpose

As the reference implies, this example was deliberately built to test out an approach to building distributed systems (DIADEM) using Ada. It concentrates on distributed concurrency control.

30.3 Current usage

None.

30.4 Source

A simplified version of the Simple Railway [Denvir et al.]; see below. A description of a demonstration design appears in Chapter 10 of [Ada for Distributed Systems]; the Airport Shuttle demonstration was driving real (model) trains.

30.5 Comments

A fully-worked solution would be rather a lot of work.

Although the example is based on Ada, similar issues for any distributed systems design.

The exercise rather loses its impact without the real (model) trains...

31 Simple Railway System

31.1 Overview

An exercise consisting of a railway system made from lengths of single track joined at crossing points, a length of track consisting of one or more sections. No section of track can have more than one train on it at once.

Suggested layout is three crossing points with two lines of track connecting each pair.

31.2 Purpose

A classical resource sharing/mutual exclusion problem.

31.3 Current usage

None.

31.4 Source

Problem 4 in [Denvir et al.]. Formal solutions also appear here.

31.5 Other references

31.6 Comments

See also the comments on the Airport Shuttle System (see above).

Too general for ANSAwise courses.

32 Prussian Generals Problem

32.1 Overview

Example problem; deterministic distributed agreement.

32.2 Purpose

To demonstrate that (perfect) distributed agreement is impossible using a deterministic protocol.

32.3 Current usage

As a brief exercise.

32.4 Source

A classic problem (need a reference here).

32.5 Comments

This merits a more careful explanation than that currently presented, perhaps as a handout.

This would be better dealt with in the replication and transaction sessions.

33 Two-way channel with disconnect

33.1 Overview

An example protocol; two parties can communicate with each other until one of them disconnects.

33.2 Purpose

A simple concurrency/communications example

33.3 Current usage

None.

33.4 Source

Problem 1 in [Denvir et al.] Formal solutions also appear here.

33.5 Comments

Probably inappropriate, since it is a communications protocol example.

34 Simple network service

34.1 Overview

An example protocol; a simplified form of the OSI Network layer service.

Note: Not in the sense of an object service

34.2 Purpose

A concurrency/communications example.

34.3 Current usage

None.

34.4 Source

Problem 2 in [Denvir et al.] Formal solutions also appear here.

34.5 Comments

Probably inappropriate, since it is a communications protocol example.

35 Mixing synchronous with asynchronous input

35.1 Overview

An example protocol; mixing asynchronous and synchronous input.

35.2 Purpose

To show the problems in mixing synchronous and asynchronous input.

“The object has two inputs and one output. The output and one of the inputs respectively send and receive data in packets at regular intervals. The remaining input is asynchronous, i.e. data appears at undetermined times.

The data packets which arrive at the synchronous input may be full or empty, and the object may only output data by forwarding packets from the synchronous input or filling an empty packets with data from the asynchronous input. All packets have the same size.”

35.3 Current usage

None.

35.4 Source

Problem 8 in [Denvir et al.] Formal solutions also appear here.

35.5 Comments

An important issue for some applications, (e.g. token-based network protocols), but presented in abstract form it may be unappealing. A real-world example would be preferable (perhaps a management application?)

36 Ray tracing

36.1 Overview

Example application; networking parallel image processing

36.2 Purpose

Shows:

- parallelism (client-side concurrency)

36.3 Current usage

None.

36.4 Source

Described in detail in Chapter 6 and Appendix B of [Power Programming], including sample source code for Sun's ONC RPC and rationale and exploration of alternative approaches.

36.5 Comments

Probably the best fully-worked design and implementation example of a distributed application.

This is the only example of client-side concurrency anybody ever seems to use.

Sample implementation is not OO.

Alas, it is also inappropriate for this audience.

References

[NASA ADS]

ANSA in Action - The NASA Astrophysics Data System (APM brochure)

[Middleware for ODP]

Middleware for Open Distributed Processing (ICL sales presentation)
Version 1.0

[DEVETIR]

OPENframework in Action at DEVETIR - Ian Craig
ICL Technical Journal, May 1993

[Application Programming in ANSAware]

Application Programming in ANSAware
RM.102

[ANSAware System Manager's Guide]

ANSAware 4.1 System Manager's Guide
RM.100

[Introduction to DCE]

OSF DCE 1.0 - Introduction to DCE
Revision 1.0

[Orbix]

Orbix Programmer's Guide
IONA Technologies

[Coulouris et al.]

Distributed Systems - Concepts and Design (2nd ed.) - Coulouris, Dollimore,
and Kindberg
Addison-Wesley

[Mullender]

Distributed Systems - ed. Mullender
Addison-Wesley

[Bacon 1993]

Concurrent Systems - Bacon, J.
Addison-Wesley

[ISO/IEC TR 10167]

Guidelines for the application of Estelle, LOTOS and SDL

[Denvir et al.]

The Analysis of Concurrent Systems, Cambridge 1983 Proceedings
Spring-Verlag Lecture Notes in Computer Systems, 207

[O'Connell et al.]

A review of four distribution infrastructures - O'Connell et al.
Distrib. Syst. Engng 1(1994) 202-211

[Moucharis and Chang]

Video Mail Service
TINA '93 Conference Proceedings (?)

[Guidon et al.]

Breakdowns and Processes during the Early Activities of Software Design by
Professionals -reprinted in:
Software State of the Art - Selected papers - ed. DeMarco and Lister

[Ada for Distributed Systems]

Ada for distributed systems - ed. Atkinson et al.
Cambridge University Press

[Power Programming]

Power Programming with RPC - John Bloomer
O'Reilly and Associates

[Freephone Demonstrator Notes]

Freephone Demonstrator Notes
RC.xxx

[Legacy Systems]

Legacy Systems - The Way Forward
APM.1181

[Information Model for Federation]

Information Model for Federation
APM.1229

[Writing Distributed Applications using ANSA]

Writing Distributed Applications using ANSA and ANSAware
APM.1261

[Distributed Communications Techniques]

Distributed Communications Techniques
APM.1340

[Specifying Services]

Specifying Services for Distributed Systems
APM.1348

[Designing Applications with CORBA]

ANSAware - Designing Applications with CORBA
APM.1352

[Replication Techniques]

ANSAware - Replication Techniques for Distributed Systems
APM.1358

[CORBA in the Real World]

ANSAware - CORBA in the Real World
APM.1379

[Searching for traffic data]

Searching for traffic data using ANSA trading
APM.1381

[Catching Up With the Future]

Catching Up With the Future
APM.1396

