



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

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (0223) 323010
+44 223 323010
+44 223 359779
apm@ansa.co.uk**

APM Business Unit

Proposal for Training in Distributed Systems [BNR]

Chris Mayers

Abstract

Proposal for a training course in Distributed Systems for BNR Europe Ltd.

Notes:

1. The Appendix is not a a FrameMaker document. It is two Postscript files, generated from Microsoft Project for Windows and Quattro Pro for Windows. Refer to the author for these files; they are not held within the APM doc system.
2. This document has non-standard front matter in front.doc. The page with the APM masthead is intended as the second page; the other page is the first page (bound inside the front cover). A template for new proposals should sort this out. (The APM doc system overwrites frontispiece.doc when you check out a document.)
3. Although the main document uses a left/right page format, it should be printed single-sided. (The layout is not ideal.) There are also a couple of blank pages at the end of the Table of Contents and in the dummy Appendix.

APM.1254.00.04

Draft

25 April 1995

Marketing and Contracts

Distribution:

Supersedes:

Superseded by:



Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE:
INTERNATIONAL:
FAX:
COMPANY No:
ARPA Internet:

Cambridge (0223) 323010
+44 223 323010
+44 223 359779
2300071
apm@ansa.co.uk

**Proposal
for
Training in Distributed Systems
for
BNR Europe Ltd.**

Proposal
for
Training in Distributed Systems
for
BNR Europe Ltd.

Contents

1	1	Executive Summary
1	1.1	Requirement
1	1.2	Proposal
1	1.3	Costs
2	1.4	Timescales
2	1.5	Benefits
2	1.6	Key assumptions
4	2	Analysis of the Requirement
4	2.1	Overall requirement
4	2.2	Principles and practice
5	2.3	Technical content of training programme
5	2.4	Practical content of training programme
6	2.5	Self-study
6	2.6	Technical facilities and logistics
7	2.7	Course review, feedback, and updating
7	2.8	BNR involvement in course development
9	3	The Proposal
9	3.1	Overview
9	3.2	Programme scope and structure
10	3.2.1	Prerequisites
11	3.2.2	Understanding Distributed Systems Architecture (2 days)
11	3.2.3	Building Distributed Applications (3 days)
12	3.2.4	Specifying and Managing Distributed Systems (2 days)
12	3.2.5	Resource Control and Concurrency in Distributed Systems (3 days)
13	3.3	Follow-on Course
14	3.4	Project Control
14	3.5	Project Plan Structure and Deliverables
15	3.6	Assumptions and Dependencies
15	3.7	Progress Reporting
15	3.8	Timescales and Costs
16	3.9	Terms and Conditions
16	3.10	Confidentiality, Rights, and Liabilities
18	4	Experience and Qualifications
18	4.1	General
19	5	Appendix - Proposed Timescales and Cost Breakdown

1 Executive Summary

1.1 Requirement

BNR Europe Ltd. (“BNR”) have a requirement for a customized training programme for between 60 and 100 staff in the principles and practical realization of distributed systems. This proposal from Architecture Projects Management Ltd. (“APM”) is a detailed response to that requirement, prepared at the request of Ian Cole.

Architecture Projects Management Ltd. thank BNR Europe Ltd. for the opportunity to submit this proposal.

1.2 Proposal

APM propose that:

- APM will develop a custom training programme consisting of 2 courses of 2 modules each
- APM will develop all the course materials including practical examples based on BNR’s Open Distributed Systems (ODS) Platform
- APM will present each course once to BNR, to 10-15 people
- APM will develop self-study materials for use by course participants, and will provide support for this
- APM will package and hand over the complete training programme to BNR

This proposal covers the training of 10-15 people. APM will submit on request a separate proposal for training additional BNR staff to meet BNR’s requirement.

This proposal does not cover the training of BNR staff to give the course presentations themselves. APM will submit a separate proposal for this on request.

This proposal is based on APM’s best understanding of BNR’s requirements.

1.3 Costs

APM propose that the work be carried out on the basis of time plus materials with an overall ceiling which will not be exceeded without BNR’s authority and negotiation. For budgetary purposes, the materials costs should be allowed as 10% additional to the time cost.

The costs are summarized as follows:

Table 1.1: Programme Time Costs

Element	Cost
Programme Definition and Introductory Course Development	£28,800
Advanced Course Development	£30,400
Self-Study Development and Support	£9,600
Course Presentations	£5,600
Programme Handover	£2,400
<i>Total</i>	<i>£76,800</i>

1.4 Timescales

One month has been allowed for pre-project negotiation, so the work in the plan starts at Month 2.

Table 1.2: Programme Timescales

Deliverable	Date
Proposal delivered	Month 1
Work start	Month 2
Programme definition complete	Month 3 + 2 weeks
Introductory Course complete	Month 5 + 2 weeks
Advanced Course complete	Month 8 + 2 weeks
Handover complete	Month 8 + 4 weeks

A more detailed breakdown appears in the Appendix.

1.5 Benefits

Implementing the training programme would have the following benefits:

- effective transfer of specialist knowledge and skills to BNR
- rapid and efficient transfer of this knowledge and skills to a fast-growing team
- relieving the pressure on key BNR staff to train others

1.6 Key assumptions

This proposal makes the following key assumptions:

- BNR will provide APM with the necessary software, licenses, documentation, and support for APM's use of the BNR ODS platform at APM's premises for the purposes of course development only
- BNR will provide workstations and networking facilities for the course presentations on-site at BNR
- BNR will review the course material as allowed for in the Project Plan, and provide feedback promptly so as to avoid project delay

- BNR will provide assistance with the definition of examples for use in the course
- BNR will nominate a person as a single point of contact with the APM project manager; this person will be empowered to agree any decisions relating to this training programme.

2 Analysis of the Requirement

2.1 Overall requirement

Training in distributed systems is currently hard to obtain. This is because:

- ANSA's research in this field over the last 9 years is only just beginning to emerge into general commercial exploitation
- the industry experience acquired in building distributed systems is only now becoming a coherent body of practical knowledge

Thus we now see research and practical experience converging into widespread exploitation.

Currently-offered commercial training courses are of low technical content and usually focus upon a particular specific and narrow technology, trading out the more general principles in favour of thinly-disguised attempts to sell or promote a particular solution. These would have three disadvantages for BNR. First, the examples used in those courses would not match BNR's needs. Second, the training would have too narrow a focus; it would concentrate just on specific products rather than cover overall principles. Third, BNR would not be able to retain control over the course material, and so not be able to give repeat presentations of the course to new staff.

Both parties are therefore agreed that the need is for a training programme which first provides knowledge and insight into general principles of distributed systems and then supplements this with detailed technical examples which are directly relevant and based upon BNR's needs and experience.

2.2 Principles and practice

A training programme that simply aimed to teach distributed systems from a single viewpoint (as one would teach a new programming language) would not meet BNR's need. This is because:

- there are new concepts to absorb, and old concepts to be reconsidered
- technology continues to race ahead; it is unclear which technology will predominate, if any
- it is important to evaluate a range of real products to understand their strengths and weaknesses, and the different levels at which they operate
- actual distributed systems will contain legacy software, new in-house applications, and new off-the-shelf software; this issue is an essential component of the programme

A single training programme must therefore cover both principles and practice, while still carefully distinguishing between them.

2.3 Technical content of training programme

Distributed systems are not an isolated technical topic. Rather, they span most computing disciplines:

- data structures and algorithms
- programming languages
- architecture
- operating systems
- software methodology and engineering
- databases and information retrieval

It is therefore vital that the course material draw on relevant topics from these disciplines. This will ensure that course participants who already have a background in these disciplines can build on their existing knowledge. (On the other hand, course participants who do not have such a background need only absorb the material that is directly relevant.)

Examples of topics from these disciplines are:

- concurrency
- memory management
- naming and addressing
- debugging and testing

It is APM's understanding that course participants will come from a range of technical backgrounds, with different blends of knowledge and skills. One way of tackling this is to relate the course material to participants' other knowledge, even if this lies outside the scope of the programme itself.

For example, the programme should briefly describe low-level communication protocols, an area that most participants will be familiar with. This approach must be carried through the entire programme, as there may be participants to whom many of the ideas are new.

2.4 Practical content of training programme

Because many of the topics will be new to the participants, the training programme must have a large practical element. This has two benefits:

- it demonstrates that the ideas can be applied
- it ensures that the ideas are retained.

The practical content must consist of examples that relate directly to BNR's activities and applications as this will result in more effective training. The practical sessions should be a high percentage of the training programme.

BNR's Open Distributed Systems (ODS) Platform will be used for the main practical examples. This will also build confidence and fluency with the tools that will be used in the participants' future work.

In order to maintain focus upon the needs of BNR, one main example, drawn from BNR's activities, will be used as a theme for all the major practical work. This has two benefits; it maintains the focus of the programme, and it reinforces the concept of viewing a system from different viewpoints. This

main example should be identified and refined at the beginning of programme development.

2.5 Self-study

A single continuous training programme that covered all the topics required by BNR would occupy 5 weeks, at least. It would be highly intensive, participants' attention would not be held, and the need for such a long period away from the desk would disrupt the BNR product programme. A modular programme is proposed, as follows:

- Understanding Distributed Systems Architecture
- Building Distributed Applications
- Specifying and Managing Distributed Systems
- Resource Control and Concurrency in Distributed Systems

This has the benefit that it can be repeated and that staff who miss one item can be scheduled to attend a repeat course covering that topic.

A modular programme can be advantageous, if participants have the opportunity for self-study between modules, by permitting the programme to be flexibly scheduled. Self-study material could include:

- basic self-assessment exercises to check understanding of course material
- self-paced tutorials
- more advanced exercises to explore particular topics more deeply
- reference books and further reading

Self-study material would supplement the course material; it would not be a substitute for it.

Of these, the basic self-assessment exercises appear most appropriate. They would be objective, require least effort to devise, and are a cross-check on the quality of the course material and its presentation. They can also be used as a "refresher" before participating in a new module. They must be sufficiently challenging that participants are motivated to take time over them.

The answers to the exercises can double as ready-reference material. This will reinforce the new modes of problem-solving that the course covers.

Effective self-study also requires someone to consult when in doubt. Electronic mail would be an ideal medium for asking such questions. It provides a permanent record of questions and answers, allowing both the self-study material and the main course material to be fine-tuned. This approach is recommended.

2.6 Technical facilities and logistics

Previous experience shows that two people can share one workstation without problems. This is in fact an advantage for a technical course of this kind, as it allows a person lacking a specific technical expertise to be paired with someone who already has it. It also builds confidence by allowing participants to sort out simple problems between themselves without needing to involve the course presenter.

For 10-15 people, up to 10 networked workstations will be needed, allowing for a machine for the course presenter, and one spare. Preferably these should be on a separate Ethernet segment to avoid potential disruption to other users; it can normally be left connected to the main network.

ODS supports the following systems:

- Sun4 with SunOS 4.1.2
- HP9000s700 with HP/UX 8.0
- ICL DRS6000 with DRS/NX SVR4.0

The Sun4 and HP9000s700 platforms are recommended. To minimise disruption, the same machines should be used for each course presentation.

2.7 Course review, feedback, and updating

As with any project, reviews should be planned during programme development. These should concentrate on checking the balance of programme content, consistency of style, and technical accuracy. The following review checkpoints are recommended:

- a review of the detailed course content
- a dry run of each practical example
- after the first presentation of each course module, a review of course feedback from the participants
- before the presentation of a course module, a review of any feedback from self-study after the previous course module

Both BNR and the course developer will then:

- analyse the course feedback
- identify corrections to be made during the programme development
- identify corrections to be deferred

This review process requires a person nominated by BNR to agree these decisions.

Review comments should be incorporated promptly in time for the next course presentation. Course updates are necessary to take account of:

- course feedback as described above
- new technical material as appropriate
- new audiences

Allowing for 60-100 people to be trained, with 10-15 people per course presentation, this means between 4 and 10 presentations of each module. It would be wise to anticipate one detailed update of each module over this time. Such an update would allow topic material to be transferred between modules, for instance to adjust the course schedule. This is the subject of a separate proposal.

2.8 BNR involvement in course development

As with any customized training course, BNR must be involved so as to maximize course relevance, but at minimum cost in effort.

BNR's involvement will therefore cover:

- promoting the programme within BNR
- reviewing the detailed programme outline
- defining a main example for practical sessions
- providing technical information and support for BNR's Open Distributed Systems (ODS) Platform
- providing workstations and networking facilities for the course presentations

BNR will need to provide a person as a single point of contact for these activities, who can take any decisions necessary.

3 The Proposal

3.1 Overview

The training programme will consist of:

- briefing sessions on distributed systems
- hands-on sessions using BNR's ODS Platform
- self-study material for use by participants

Although the practical sessions will use particular technology, the programme will distinguish between principles and practice; there will also be capacity within the course to include briefing sessions on third-party products and technologies (for example, Microsoft Windows NT), to compare and contrast them, and to illustrate ideas from them.

Each course will be suitable for groups of 10-15 participants; 12 is a recommended number, striking a balance between course throughput and individual attention.

3.2 Programme scope and structure

The focus will be on proven concepts and techniques.

Since BNR staff have diverse experience with hardware and software backgrounds, the programme will be modular, allowing flexibility in planning staff participation.

The programme has two courses:

- Introductory course
- Advanced course

On completing the Introductory course, participants will understand the basic principles of distributed systems, and will have written a simple client/server application. They will be able to apply this knowledge to write more complex client software for distributed applications.

On completing the Advanced course, participants will understand how to specify, manage, and develop services, and will have written and tested a simple server application incorporating concurrency features. They will be able to apply this knowledge to design and implement advanced server software for distributed applications.

The courses are built from the following modules:

Table 3.1: Course Structure

Module	Title and Content Summary	Course	Duration
ARCH-INTRO	Understanding Distributed Systems Architecture - Characteristics of Distributed Systems - Benefits of Distributed Systems - Distributed Systems Architectures - Viewpoints - Assumptions and their consequences - Transparency policies and mechanisms - Trading, naming, and binding - Commercial product overview	Introductory	2 days
BUILD-INTRO	Building Distributed Applications - Computational Viewpoint in depth - Engineering Viewpoint in depth - Remote Execution and RPC protocols - ODS Tools and their use - Building a simple client/server application - Commercial products compared and contrasted	Introductory	3 days
SPEC-ADV	Specifying and Managing Distributed Systems - Types and type systems - Trading, its specification and use - Naming and federation - Error handling policies and approaches - Testing and debugging - Trade-offs and compromises - Specifying and building a managed client/server application	Advanced	2 days
CONTROL-ADV	Resource Control and Concurrency in Distributed Systems - What is resource control? - Memory management - Concurrency rationale - Processes, tasks, and threads - Scheduling and synchronization - Concurrency policies - Concurrency mechanisms - Systems support for concurrency - Building a concurrent server - Testing and debugging concurrency	Advanced	3 days

3.2.1 Prerequisites

Each module builds on the expertise gained in the previous one; it is not possible to omit any of them before progressing to the next module.

The Introductory course has the following prerequisites:

- a degree qualification in computer science or a related discipline, or equivalent industrial experience
- basic fluency in programming in C++
- basic fluency in using simple Unix user commands

The Advanced course has the same prerequisites, but is technically more challenging, and will be faster-paced; participants must anticipate this.

3.2.2 Understanding Distributed Systems Architecture (2 days)

This module will cover the principles and concepts of distributed systems as shown in the ANSA architecture and in ODS. On completing this module, participants will be able to:

- describe the characteristic properties of distributed systems
- relate distributed systems to proposed solutions being marketed commercially
- give examples of the types of trade-offs made possible by distributed systems
- assess objectively the business and technical benefits offered by a distributed system in solving a particular problem
- describe a distributed systems architecture
- analyse a software component to determine if it is consistent with a distributed systems architecture
- outline the different viewpoints of a distributed system
- decide when and how to apply these techniques

There are no prerequisites for this module.

The topics are:

- the business and technical need for distributed systems
- challenges in legacy systems and their connectivity
- the scope of distributed systems
- how networking relates to distributed systems
- how client-server and object technology relate to distributed systems
- distributed systems and software reuse
- declarative and procedural approaches
- separation of concerns: the viewpoints and their contents
- layering and its limitations
- systems characteristics (for example, dependability)
- transparency policies and mechanisms
- services and trading
- naming and binding
- existing architectures: standards, strengths, weaknesses, and work-in-progress

The self-study material covers all the above topics.

3.2.3 Building Distributed Applications (3 days)

This module will describe the Computational and Engineering viewpoints in depth, and the tools that support their use. It will cover the practical techniques for developing distributed systems.

On completing this module, participants will be able to:

- describe the key features of the Computational and Engineering viewpoints and the difference between them

- compare and contrast ODS, ANSA, DCE, and CORBA
- use ODS tools to build a simple client/server application

The module “Understanding Distributed Systems Architecture” is a prerequisite for this module.

This module includes hands-on practical sessions.

The topics covered in this module are:

- the Computational viewpoint in depth
- the Engineering viewpoint in depth
- detailed comparisons between ODS, ANSA, DCE, and CORBA, and corresponding products
- remote execution and RPC protocols
- common pitfalls in building distributed applications
- building and enhancing a simple client/server application

The self-study material for this module will cover all the above topics.

3.2.4 Specifying and Managing Distributed Systems (2 days)

This module will cover the techniques needed to specify and manage distributed systems.

The module “Building Distributed Applications” is a prerequisite for this module.

This module includes hands-on practical sessions.

On completing this module, participants will be able to:

- determine how and where to make effective use of trading techniques
- specify interfaces using declarative techniques and tools
- determine when multiple interfaces are needed
- determine when and where transparencies are needed
- determine when trade-offs and compromises are appropriate, and when they are not
- define error handling policies

The topics covered in this module are:

- types, trading, federation, and object management
- naming and binding
- management tools
- transaction processing techniques

The self-study material for this module will cover all the above topics.

3.2.5 Resource Control and Concurrency in Distributed Systems (3 days)

This module will cover practical techniques for designing, implementing, testing and debugging concurrency in distributed systems. It also covers other resource control issues.

This module is an intensive module, and will be technically challenging for the participants.

The module “Specifying and Managing Distributed Systems” is a prerequisite for this module.

On completing this module, participants will be able to:

- determine the resource requirements of an application
- decide when to use concurrency in a design
- design and implement a server with the appropriate concurrency characteristics
- describe and select concurrency mechanisms and policies appropriately
- plan and implement a test and debugging strategy for a concurrent application
- build robust applications by eliminating memory leaks

The topics covered in this module are:

- transparencies and resources
- the need for concurrency
- concurrency and real-time
- concurrency and databases
- scheduling and synchronization
- concurrent service design and non-determinacy
- lightweight and heavyweight concurrency
- concurrency policies
- concurrency mechanisms
- systems support for concurrency (e.g. Unix and Microsoft Windows NT)
- memory allocation and management
- enhancing a server application to add concurrency
- testing and debugging

The self-study material for this module will focus on the detailed understanding of concurrency.

3.3 Follow-on Course

The training programme described here does not cover the complete range of topics in distributed systems. A follow-on course would cover advanced issues in real-time systems:

- quality-of-service and bindings
- streams
- deadlines
- synchronous programming

It would also cover dependability techniques for:

- transactions
- replication

APM will submit a proposal for such a follow-on course on request.

3.4 Project Control

The development and delivery of this training programme will be governed by a Project Plan agreed with BNR. Subsequent changes to the Project Plan will be recorded and negotiated by agreement between APM and BNR.

APM will provide a Project Manager for this project. BNR will nominate a person as a main point of a contact with the APM Project Manager. Whilst it is expected that APM will liaise with various contacts within BNR, this person must be responsible for agreeing project decisions.

3.5 Project Plan Structure and Deliverables

APM propose that the project plan have the following phases:

- **Programme Definition Phase**

During this phase the detailed programme objectives will be refined by discussion with BNR, and a detailed programme summary produced.

A Main Example based on BNR's activities will be identified and matched against the detailed programme objectives.

The format of course materials will be finalized.
- **Programme Development Phase**

During this phase APM will write and review the course materials. The Main Example will be refined and specific exercises derived from it. APM will write self-study material in collaboration with BNR. APM will write and test the source code examples for use in hands-on sessions.

The Programme Development Phase has two parts, for the Introductory and Advanced courses.
- **Course Presentation Phase**

APM will present each course module once, and record and collate feedback. APM will provide support for staff self-study via electronic mail during this time.
- **Programme Handover Phase**

APM will revise the programme in the light of feedback from the course presentations, and update the course materials accordingly.

APM will then package and hand over the complete course materials and supporting documentation to BNR.

APM will deliver to BNR:

- a detailed programme outline, for review by BNR management
- the course materials in a form suitable for BNR to present the courses themselves (including speaker's notes) - in both electronic and printed form
- complete source code for practical examples. No separate documentation is necessary, as this is covered in the course materials

- one presentation of each course module to a suitable number of BNR staff on BNR premises
- self-study material in a form to be agreed with BNR.
- support for staff self-study via electronic mail using the Internet

This forms a complete self-contained training programme that, with guidance from BNR, can be maintained and upgraded. APM will submit a proposal for maintaining and upgrading this material at BNR's request.

APM does not include in this proposal an activity for training any BNR staff to present the course themselves. APM will submit a separate proposal for this activity on request.

3.6 Assumptions and Dependencies

APM will assume and rely on BNR's involvement in the following activities:

- providing a person as a main point of contact with APM
- promoting the course within BNR
- reviewing the detailed programme outline
- defining a main example for practical sessions
- providing technical information and support for BNR's Open Distributed Systems (ODS) Platform
- providing room space, presentation facilities, workstations and networking facilities for the course presentations at BNR's Harlow site

3.7 Progress Reporting

APM's Project Manager will provide regular monthly progress reports to the nominated BNR main point of contact.

3.8 Timescales and Costs

APM propose that the work be carried out on the basis of time plus materials with an overall ceiling which will not be exceeded without BNR's authority and negotiation. For budgetary purposes, the materials costs should be allowed as 10% additional to the time cost.

The costs are summarized as follows:

Table 3.2: Programme Time Costs

Element	Cost
Programme Definition and Introductory Course Development	£28,800
Advanced Course Development	£30,400
Self-Study Development and Support	£9,600
Course Presentations	£5,600
Programme Handover	£2,400
<i>Total</i>	<i>£76,800</i>

APM propose payment terms as against a monthly invoice for work done and accepted according to the progress report described above.

The timescales are driven by the start date determined by BNR, therefore all dates given are with respect to that. There are no other external timescale constraints. APM's analysis shows that the programme development cannot be speeded up by applying additional effort; the programme outline must be completed before course development starts, and it is not feasible to develop course modules in parallel.

Refer to the Appendix for an outline plan. One month has been allowed for pre-project negotiation, so the work in the plan starts at Month 2.

Table 3.3: Programme Timescales

Deliverable	Date
Proposal delivered	Month 1
Work start	Month 2
Programme definition complete	Month 3 + 2 weeks
Introductory Course complete	Month 5 + 2 weeks
Advanced Course complete	Month 8 + 2 weeks
Handover complete	Month 8 + 4 weeks

3.9 Terms and Conditions

This proposal is subject to APM's standard terms and conditions.

3.10 Confidentiality, Rights, and Liabilities

APM will retain copyright and IPRs to that which it develops and will grant BNR a licence in perpetuity to use it.

In particular, APM reserves the right to use non-BNR-proprietary material in other courses which APM may develop and deliver to the general market from time to time, and APM's quotation takes this into account.

BNR will provide APM with the necessary software licenses, documentation, and support for APM's use of the BNR ODS platform for the purposes of course development only. APM will not use the ODS-specific software in other courses that it may develop without prior consultation and agreement with BNR.

APM agrees to keep confidential all BNR proprietary information, in accordance with the principles and details established in the ANSA Sponsorship Agreement Revision 7.

This proposal is provided for the sole purpose of enabling BNR to evaluate the proposed project and workplan. It is provided on the understanding that all ownership rights and copyright in it remain with APM and that it shall not be disclosed to any third party without the express consent of APM.

This proposal does not constitute a contract or any part of a contract and nothing contained in it shall be binding unless and until it is expressly incorporated in a formal contractual agreement.

This proposal is valid for sixty (60) days from 15 March 1994.

4 Experience and Qualifications

4.1 General

APM has been researching into distributed systems for over 9 years. Having originated many of the key concepts in distributed systems and successfully introduced them into international and industrial standards, APM is well placed to train organizations in their practical application.

The APM Business Unit has presented training courses in distributed systems, ranging from one-day executive overview courses, to one-week in-depth courses on ANSAware. The experience derived from developing and presenting these courses has been incorporated into this proposal.

APM understands that this project is of key importance to BNR, and APM will therefore allocate high quality resources and management attention in order to ensure its success. The staff which APM will allocate to this task will include:

- Chris Mayers, who will be the Project Manager, liaise with BNR, and lead the course development
- Ian Macmillan, who will develop the source code for the practical examples

The technical content and overall quality of the course will be overseen and approved by Dr Andrew Herbert, APM's Technical Director.

The APM Business Unit can draw on the expertise of the Research Unit for specific discussions.

5 Appendix - Proposed Timescales and Cost Breakdown

Please refer to the attached sheets.

