Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: Cambridge (0223) 323010
INTERNATIONAL: +44 223 323010
FAX: +44 223 359779
E-MAIL: apm@ansa.co.uk

ANSA Phase III

Operational Plan: 1/3/93 - 28/2/95

Abstract

This is the operational plan for the period from 1/3/93 to 28/2/95 of Phase III.

It shows:
- what are the deliverables
- when they are derived
- when they are reviewed and delivered
- the process by which they are derived and delivered
- the external dependencies.

The audience for this document is the ANSA Phase III Management and Technical Committees. It assumes familiarity with the objectives set out in APM1030.

APM.1031.00.01 **Draft** 29 June 1993
Marketing and Contracts

Distribution:
Supersedes:
Superseded by:

Copyright © 1993 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.

Operational Plan: 1/3/93 - 28/2/95

Marketing and Contracts

Operational Plan: 1/3/93 - 28/2/95

APM.1031.00.01

29 June 1993

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	(0223) 323010
INTERNATIONAL	+44 223 323010
FAX	+44 223 359779
E-MAIL	apm@ansa.co.uk

Copyright © 1993 Architecture Projects Management Limited

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 Introduction

This is the operational plan for the period from 1/3/93 to 28/2/95 of Phase III of the ANSA work programme. The target audience for this document is the ANSA Phase III Management and Technical Committees.

This document provides the details of the manner in which the ANSA Team plans to achieve the objectives, agreed with the sponsors for Phase III of the ANSA programme, and set out in [APM1030].

This plan shows:

- what are the results
- when they are derived
- when they are reviewed and delivered
- the process by which they are derived and delivered
- what external dependencies exist.

The Management Committee has delegated the control of the plan at the macro level to the ANSA Phase III Technical Committee. Detailed day to day control is the responsibility of APM.

The plan assumes the current 10 initial sponsors, which, at a funding level agreed by the Management Committee, provides a team[1] of 16, including the Chief Architect and the Project Director.

It is anticipated that this plan will be updated as work progresses. Updates require, and will only be made with, the prior agreement of the sponsors through the mechanisms of the ANSA Phase III Management and Technical Committees.

The plan has a top-down structure:

- **Main Deliverables** integrate the results achieved in the different foci of the programme. A main deliverable is achieved when a set of work packages is completed, and the results of these work packages are consistent.
- **Work Packages** each aim to answer one or several key questions. Questions are agreed with the Technical Committee to be representative of a technical focus.

1.1 Structure of this document

Chapter 2 sets out the organisation of the ANSA Team, tailored to fit the technical foci. It also describes the available resources.

1. This team is located in Cambridge, UK, and is referred to as “the ANSA Team” in this document.

Chapter 3 sets out the main deliverables and dates and identifies the work packages which contribute to each deliverable.

Chapter 4 describes the mechanisms and targets for the delivery of the results of the work programme, to sponsors and standards organisations.

For each of the technical foci, Chapters 5, 6, 7 and 8 set out:

1. **Approach**: a statement of the planned approach and of the technical assumptions which have led to this approach
2. **Baseline**: identifying the Phase II (ISA) results which provide foundation for the work[1]
3. **Key questions**: which scope the work and are agreed in advance with the Technical Committee
4. **Work packages**: definition of work packages for the next 24 months, identifying effort required and dates for completion.

Chapters 9 and 10 describe the work packages associated with standards and project management.

Chapter 11 contains a glossary.

Chapter 12 contains the planning charts, dependency graphs and other relevant output from AutoPlan[2].

1. The references remain in old format for the time being.

2. This material needs to be printed separately as it cannot be integrated with FrameMaker source files. If it is missing, please apply to Phase III Project Management.

2 Method of work

2.1 Team organisation

The ANSA Team will be managed by the Project Director and the Chief Architect, as defined in the Third Schedule of the Programme Management and Execution Agreement. Certain tasks are delegated to the Research Unit Manager.

The ANSA Team will be organised in Reporting Groups. Each group looks after a Reporting Area (an area of work). The following Reporting Areas have been identified for the period covered by this plan:

1. Dependability
2. Performance
3. Automated transparency
4. Federation
5. Standards
6. Project Management.

The first four are technical foci aimed at satisfying the objectives set out in [APM1030]. They each include the activity of transferring technology to sponsors. Standards and project management are reported separately.

Each of the four technical groups interacts with the sponsors to:

- obtain sponsors' requirements
- effect technology transfer: placing results at the sponsors' disposal
- ensure smooth operation of Review Teams set up by the Technical Committee

Each technical group will maintain

- a set of scenarios setting out sponsor's application requirements (some scenarios may be shared amongst groups)
- a list of questions generated from the scenarios and the work itself to provide a test for review of progress. [The lists to be made initially from the questions generated by the team discussions.]
- a standard presentation (developed jointly with the corresponding Review Team) summarizing the group's objectives and accomplishments
- a bibliography covering relevant research literature
- a summary of current documents and documents which are either being updated or written. This will include design notes and documentation for software prototypes
- summaries of significant technical discussions, particular those held via internal email which have not otherwise been captured in current documentation.

Each group will have

- an editor with responsibility for consistent documentation,
- a librarian with responsibility for software integration, and
- a liaison officer with responsibility for sponsor interaction.

To ensure consistency and integration across the project, the Chief Architect and his deputy exercise overall design and editorial authority. Groups are also expected to work closely together.

The plan has been derived so as to ensure that

- where possible, work packages are staffed by 2 to 3 people
- team members are responsible for results of one work package at the time, but have every opportunity to relate their work with work in other work packages and other groups.

2.2 Resources

Size of reporting groups for the period 1/3/93 to 28/2/94 (as agreed with the task force) is given by the table below:

Table 2.1: Size of Reporting Groups

Reporting Group	group size
Dependability	4
Performance	4
Automated transparency	3
Federation	2
Standards	1
Project management	2
	16

2.2.1 Staff availability

Effort figures assume a normal working year (allowing for holidays, sickness and training) of 1650 hours per person per year.

2.3 Technology baseline

For work requiring a distributed object oriented environment, ANSAware 4.1 will be the baseline until CORBA is available in a form suitable for use by the ANSA Team and sponsors. To ensure a smooth transition, efforts will be made to increase alignment of ANSAware and CORBA.

Several real time kernels will be investigated during the early stages of the work on performance. Technology selection for prototyping will be agreed with the Technical Committee in advance.

At the outset, C will continue to be used as the primary implementation language. The result anticipated in Automated Transparencies suggests a transition to greater use of C++.

3 Schedule of results

The main deliverables, their delivery dates and the work packages which contribute to the production of the main deliverable are described in the table below. Contributing work packages deliver intermediate results, at the dates given in Chapter 12.

Table 3.1: Delivery schedule of Main Deliverables

Code	Main deliverable	Due date	Contributing work packages	
MD 1	Models for performance, dependability and federation (MODELS)	12/93	P1	Modelling requirements for performance
			D1	Modelling requirements for dependability
			F1	Architecture for federation
MD 2	Programming abstractions and management models (PROMGM)	1/94	P2	Programming abstractions and management model for performance
			D2	Programming abstractions and management model for dependability
			F2	Programming abstractions and management model for federation
			A1	Approaches to automated transparencies
			A2	Structuring applications for distribution
			A3	Architecture for automated transparencies
MD 3	Programming tools (TOOLS)	5/94	A7	Automated transparencies for C++
			A8	Automated transparencies for dependability
			A9	Automated transparencies for performance and real time
MD 4	Nucleus Interface and Engineering Model (NUIFEN)	5/94	P3	Engineering model for performance
			D3	Engineering model for dependability
			F3	Engineering model of management of interface references in federation
			A4	Nucleus interface and engineering model
MD 5	Nucleus Interface implemented for CORBA (NENCOR)	2/95	P4	Performance functions implemented for CORBA
			D4	Dependability functions implemented for CORBA
			F4	Prototype of a network of interceptors
			A5	Nucleus interface implemented for CORBA
			A6	Nucleus interface for performance and real time
MD 6	Optimised implementations for dependability and performance (OPIMP)	2/95	P5	Implementation of performance functions in an optimised environment
			D5	Implementation of dependability functions in an optimised environment

4 Delivery of results

Note: The current text provides a satisfactory basis upon which to deliver Phase III results, but several issues of detail remain outstanding. This section will thus be updated as and when the details of technology transfer policies and mechanisms are agreed by the Technical Committee.

4.1 Mechanisms

The results of the programme will be delivered to the sponsors by

1. **secondees**: Secondees are the primary contact point for sponsors, especially regarding technical matters. The Secondee Agreement states that a secondee is “expected to spend approximately 5% of his or her time transferring technology into the sponsor”.
2. **work in progress reports**: The Chief Architect will report on the technical progress of the ANSA Team to the Technical Committee when it meets. It is anticipated that this will be quarterly. The report will cover:
   — key results, potential patents
   — overview of documents produced
   — software development in progress
   — capabilities and limitations
   — how the individual items fit together in a coherent architecture
   — relationships to other initiatives in industry and standards
   — issues for resolution by the Technical Committee.

   The work in progress report will be captured in hard copy of overhead slides. It is intended to experiment with recording the presentations on video tape.
3. **workshops**: The ANSA Team will organise workshops when the work programme requires a major input or produces a Main Deliverable. Each workshop will result in a set of proceedings (presented papers) and a report, analysing the result of the workshop. The intent is to have two or three workshops per year. Workshops associated with Main Deliverables will be structured as tutorials followed by interactive discussion.Where appropriate they will be targetted at specific technical groups so that architectural issues are put in the context of specific application requirements
4. **consultancy**: Consultancy can be provided to sponsors as required. Consultancy covers the scheduled use of team members’ expertise on a problem of a sponsor’s choice. Scheduling this effort is by negotiation with the Project Director. The Sponsorship Agreement (clause 8.2(a)ii) allows 5 person days per sponsor per year. This has been factored into the plan.

Further consultancy (using APM staff, at a fee, can be arranged by negotiation with APM).

5. **review team**s: Review teams will have responsibility for helping the team produce documents and prototypes that can be readily transferred to the sponsors (see 5.2 below)
6. **documents and prototypes**: Documents and prototypes will be used to back up the forms of technology transfer described above. They are not to be used as the sole means of technology transfer. Five types of document exist: RC (request for comment), TR (Technical Report), AR (Architecture Report), BR (Briefing), and CO (Contractual). Software may be a stand alone prototype, or part of an integrated system of prototypes.

4.2 Review teams

Sponsors are required to join in and support the technical progress of the project. For each focus a Review Team will be formed. Review Teams consist of between three and five members, appointed by the Technical Committee. Membership of Review Teams will be reassessed upon completion of each workpackage.

Review Team members are drawn from sponsor companies and/or associated universities.

Review Teams interact at least monthly (meetings, e-mail, etc.) with ANSA team members:

- to agree requirements by proposing scenarios for application of ANSA
- endorse the technical directions taken to meet those requirements
- to develop and maintain jointly with the team a standard “presentation and briefing paper” on the work done it the area being reviewed. The briefing to be sufficient to enable a Review Team or Technical Committee member to present the work in their own organization
- to monitor the quality and progress of the work by active participation in document and software reviews at all stages from planning to completion
- to participate in technical discussion (via email)
- to recommend adjustments of the objectives and scope of the work, where necessary, to the Technical Committee.

The lead reviewer in the Review Team for each workpackage will produce a one page summary for presentation to the TC at each of their meetings. This report should cover (at least) major progress made, issues that have arisen, and expected results for the next quarter.

They agree with the Chief Architect when presentations should be made to the Technical Committee. A Review Team is only disbanded after the results of a work package have been accepted by the Technical Committee.

The minimum commitment of a Review Team member is 2 or 3 days a month. The maximum use will be made of email to minimize the need for meetings and travel.

4.3 Targets

The results of the programme will be specifically targeted as shown in the table below. The general principle is that Architecture and Models are directed towards ODP, whereas components and interfaces are directed towards industry standards. Some results are targeted at standards initiatives which have not yet been formed. In such cases the Technical and Management Committees are invited to formulate strategies by which suitable standards initiatives can be encouraged.

Table 4.1: Targets for the results of ANSA Phase III

WP	Result type	Destination
D1	AR	ODP dependability framework (new work item)
D2	AR, prototype	Field trials > AR > OMG Object Services
D3	TR, prototype	Field trials > TR > RM-ODP, OMG Object Services
D4	TR, prototype	Field trials > TR > RM-ODP, OMG proof of concept
D5	TR, prototype	Field trials > TR
P1	AR	ODP Real Time Framework (new work item)
P2	AR, prototype	Field trials > AR > OSF, ISO OSE
P3	TR, prototype	Field trials > AR > RM-ODP, OMG Object Services
P4	TR, prototype	Field trials > TR > OMG proof of concept
P5	TR, prototype	Field trials > TR
F1	AR	ODP, NMF, OMG, OSF/DME
F2	AR, prototype	Field trials > AR > RM-ODP, OMG, OSF/DME
F3	TR, prototype	Field trials > TR > RM-ODP, OMG, OSF/DME
F4	TR, prototype	Field trials > TR > RM-ODP, OMG, OSF/DME
A1-3	AR	New work items OMG
A4-6	TR, prototype	input to D3-5 and P3-5
A7-9	TR, prototype	Field trials (D and P areas) > New work items OMG

5 Dependability

5.1 Approach

The focus here is the use of replication and atomicity in ANSA, building upon and generalizing from the current work on groups and atomicity.

Dependability is achieved through redundancy using replication techniques and consistency using atomicity techniques[1]. These two techniques can support a wide range of potential applications. They are closely interlinked: tackling both together will result in synergies between the two, and an understanding of the possible conflicts in the design of systems requiring both.

Dependability represents a range of options and solutions. For example, from the point of view of a client, a service may be highly available in the sense that it is always available for connection, but that errors in individual transactions must be recovered from by explicit client action; or the transaction may abort and consistency be restored transparently; or when a server fails another takes over to continue the service, perhaps transparently, or perhaps in cooperation with current clients. The purpose of the ANSA work in this area is to investigate the parameters of this range and architect the components needed to support as wide a range of parameters as possible.

To enable dependability techniques to be widely usable the work will attempt to minimise the extra application design and programming effort required.

5.2 Baseline

1. AR.002 A Model for Interface Groups
2. AR.004 A Model for Atomic Activities
3. AR.006 The ANSA Storage Model (available Feb 1993)
4. RC.248.03 Using Path Expressions as Concurrency Guards
5. RC.436 Principles and Techniques for Checkpoint-Based Recovery in ANSA (available Feb 1993)
6. ANSAware 4.1 support for transparent active replication (GEX) (available Feb 1993)
7. ANSAware 4.1 support for transactions (Arjuna) (available Feb 1993)

1. The term “atomicity technique” covers the notion of “transaction”.

5.3 Key questions

5.3.1 Modelling dependability

- QD1.1: what are the parameters by which dependability requirements can be expressed (e.g. fault models, consistency constraints, correctness constraints)?
- QD1.2: what are the meaningful combinations of parameters in terms of realistic application scenarios?
- QD1.3: what safeguards need to be put in place in order to disallow the construction of systems with incompatible dependability requirements?
- QD1.4: how do dependability requirements relate to interface types and quality of service requirements?
- QD1.5: how should long running interactions and intermittently connected systems be made dependable?

5.3.2 Structured kit of parts

- QD2.1: what are the basic functional components of group and atomic infrastructures?
- QD2.2: what is the engineering framework into which these functional units fit - i.e. the rules and recipes for building infrastructures to achieve different dependability requirements?
- QD2.3: what are the issues in federating systems that have different dependability infrastructures (e.g. ISO/TP, XOpen XA)?

5.3.3 Programming model

- QD3.1: what programming abstractions are required to enable selective automation of dependability requirements?
- QD3.2: what are the trade-offs between application complexity and transparency in programming dependable applications?

5.3.4 Management of dependable systems

- QD4.1: where are the management control points in dependable infrastructures and what are their interfaces?
- QD4.2: what are the configuration, system and network management requirements of dependable infrastructures?

5.4 Work packages

- D1 Report on Modelling Dependability Requirements

  A description of models for parameterising and reasoning about dependability requirements.

  Effort 9 pm

  — research survey 4 pm

  — synthesis 5 pm
- D2 Report on Programming and Management Model for Dependability

A description of a programming model and interfaces for the selective use of distribution transparency in support of dependability; guidelines for use of the model to be given in terms of the dependability model from task D1.

Effort 15 pm

— research analysis 5 pm

— experimental prototype 5 pm

— synthesis 5 pm

- D3 Report on Engineering Model for Dependability

  A complete design for a structured kit of parts to support the programming and management model for dependability from task D2; design alternatives and trade-offs to be related to the dependability model from task D1.

  Effort 21 pm

  — research analysis 4 pm

  — experimental prototype 7 pm

  — synthesis 10 pm

- D4 Implementation of Dependability for CORBA

  A prototype implementation of key components and tools to aligned with the programming and engineering models of dependability from tasks D2 and D3; the implementation to be done so that the prototypes can be used as a template for adding dependability functions layered above distributed object environments based on the OMG CORBA specification.

  Effort 28 pm

  — familiarization with CORBA 2 pm

  — replication functions 10 pm

  — atomicity functions 10 pm

  — automated transparency support 6 pm

- D5 Optimised implementation of Dependability

  A prototype implementation of optimised low-level support for the key components from task D4 to enable their use in a real-time environment.

  Effort 19 pm

  — familiarization with Real Time Operating System 2 pm

  — replication functions 8 pm

  — atomicity functions 8 pm

  — automated transparency support 1 pm

6 Performance and timeliness

6.1 Approach

The focus here is how to bring time and resource utilization issues to the application programming domain, and the engineering consequences of doing so.

Real-time systems are those in which performance is explicitly managed to meet specified timing constraints. Performance can be managed in a variety of ways: by optimising systems structure for least overhead, trading off flexibility and scaling properties against efficiency; by setting explicit performance goals for tasks within an application and using this information to schedule and cancel tasks to best effect in overload situations; by enabling applications to negotiate for resources and tailor their behaviour according to the resources obtained. The purpose of the ANSA work in this area is to investigate each of these aspects by architecting the trade-offs between flexibility and efficiency, the means for applications to set performance goals and negotiate resources and the means for systems to compensate for overload.

Distributed computing environments are taking on more and more systems functions; at the same time operating systems kernels are themselves using and providing distributed computing functions. There is considerable duplication of function and the boundary between the two is becoming blurred. ANSA work will therefore explore the potential for the migration of functionality between distributed computing environments and operating systems and its impact on the performance and structure of future systems software.

6.2 Baseline

- TR.028 Support for Multi-media Operations
- RC.425 Minutes of the First Meeting of the ANSA Work Group on Stream Interfaces

6.3 Key questions

6.3.1 Modelling performance and timeliness

- QP1.1: what parameters are required to express timing constraints, performance guarantees, and resource utilization constraints?
- QP1.2: how can performance characteristics of a “timely” system be derived from performance characteristics of components within the system?
- QP1.3: what are the meaningful combinations of parameters in terms of realistic application scenarios?

- QP1.4: how do performance requirements relate to interface types and dependability requirements?

6.3.2 Structured kit of parts for “timely” systems

- QP2.1: what operating system and network functions are required to enable applications to be able to reserve and/or monitor resources?
- QP2.2: how should communication functions be structured to enable maximum flexibility in quality of service management?
- QP2.3: how should communication functions be structured to enable maximum use of low-level protocol support for dependability and timeliness?
- QP2.4: what are the issues of federating systems using different timeliness and resource management structures?

6.3.3 Programming model for timeliness

- QP3.1: what programming abstractions are required to react to time-constrained events?
- QP3.2: what programming abstractions are required to manage and monitor resources in time-constrained systems?
- QP3.3: how can performance and timeliness be traded off against application consistency requirements and how does this impact upon dependability?

6.3.4 Managing “timely” systems

- QP4.1: how can timeliness guarantees be negotiated and managed across networked systems from end-to-end?
- QP4.2: how can network and system management information be used in time-constrained systems?

6.4 Work packages

1. P1 Report on Modelling Performance and Timeliness Requirements

   A description of models for parameterising and reasoning about performance and timeliness requirements. To include means to express, for example, deadlines, communications quality of service, resource requirements. To be related to models for dependability.

   Effort 9 pm

   — research survey 4 pm

   — synthesis 5 pm

2. P2 Report on Programming and Management Model for Performance and Timeliness

   A description of a programming model and interfaces for applications with performance and timeliness constraints; guidelines for use of the model to be given in terms of the performance and timeliness model from task P1.

   Effort 15 pm

   — research analysis 5 pm

— experimental prototype 5 pm

— synthesis 5 pm

3. P3 Report on Engineering Model for Performance and Timeliness

   A complete design for a structured kit of parts to support the programming model for performance and timeliness from task P2; design alternatives and trade-offs to be related to the dependability model from task D1.

   Effort 24 pm

   — research analysis 5 pm

   — experimental prototype 8 pm

   — synthesis 11 pm

4. P4 Performance and Timeliness for CORBA

   A prototype implementation of key components and tools to aligned with the programming and engineering models of performance from tasks P2 and P3; the implementation to be done so that the prototypes can be used as a template for adding performance and timeliness functions to distributed object environments based on the OMG CORBA specification.

   Effort 22 pm

   — familiarization with CORBA2 pm

   — timeliness functions 6 pm

   — resource management functions 6 pm

   — automated transparency support 8 pm

5. P5 Optimised Implementation of Performance and Timeliness

   A customisation of optimised kernel-level support (protocols, schedulers, memory management etc.) for the performance and timeliness components in demanding real-time environment.

   Effort 19 pm

   — familiarization with Real Time Operating System 2 pm

   — timeliness functions 7 pm

   — resource management functions 7 pm

   — automated transparency support 3 pm

7 Automated transparency

7.1 Approach

The focus here is how to mask the engineering detail of distributed processing from the application programmer and to maximize the ability to re-use distributed application software across a wide range of engineering infrastructures.

To be successful, automated programming must be based on a small, clear set of principles for what the automation process is going to achieve and how it works: if the outcome is not predictable, or if understanding the automation process is harder than ordinary programming, then users (application programmers and their organisations) will perceive no benefits and ignore the tools.

Automation relies upon abstraction of engineering into simple concepts that can be readily combined to express a wide range of requirements. Automation requires that the supporting technology be modular so that functions supporting abstract concepts can be combined. The purpose of the ANSA work in this area is to identify abstractions for distributed computing that capture both the basic concepts and those arising from dependability, performance and federation.

Since an important use of distributed computing technology is for integrating existing applications, automation is specifically required for the provision of the distribution components: the architecture must be open-ended so that distribution components can be readily combined with other components of the application (perhaps themselves generated automatically - e.g. user interfaces and database access). Previous ANSA work has explored the use of preprocessing techniques and distribution-oriented programming languages. The Phase III work will look at intermediate points between these two with the goal of enabling an incremental adoption of automated transparency and an understanding of the trade-offs between alternative approaches.

In addition to examining the spectrum of possible approaches, the work will identify and show examples of techniques that enable automated transparency tools to be converted easily between different programming environments and engineering infrastructures.

For automated transparency to be open-ended, the means of configuring engineering components supporting transparency needs to be modular. The foundation of this modularity is the ANSA nucleus. Engineering components can be combined with the nucleus to build a higher level infrastructure with added capabilities, for example dependability, but substantially the same runtime applications interfaces. The Phase III work will revise the ANSA nucleus interface so that it provides the necessary support for a modular approach to the provision of engineering components coming from the work on dependability and performance.

7.2 Baseline

- TR.031 DPL Programmer's Manual
- TR.032 DPL Reference Manual
- AR.001 The ANSA Computational Model
- PrepC and its documentation as delivered with ANSAware 4.1
- DPL documentation and software, developed in Phase I and II.

7.3 Key questions

7.3.1 Exploring the design space (architecture)

- QA1.1: what are the means of packaging programming abstractions for distributed applications and their respective strengths and weaknesses (e.g. inheritance, preprocessors, program generators)?
- QA1.2: which engineering aspects of distribution can be automated, and to what degree?
- QA1.3: what are the trade-offs in the design of the interface between applications programs and engineering infrastructures (e.g. class libraries, interpreters, stubs, generic infrastructures)?

7.3.2 Automated transparency tools

- QA2.1: how can automated transparency tools be decoupled from applications programming languages?
- QA2.2: what links between applications programming tools and runtime services must exist to deploy automated transparencies and ensure end-to-end consistency of mechanisms and protocols (e.g. what is the relationship between automated tools and object management)?

7.4 Work packages

1. A1 Report on Approaches to the Automation of Transparency

   A description of the scope for and means of automating the provision of selective distribution transparency.

   Effort 6 pm

   — research survey & experimental prototype 4 pm

   — synthesis 2 pm

2. A2 Report on Structuring Applications for Distribution

   An explanation of what kinds of programming structures are readily distributed and those which are not. Guidelines for writing and applying automated transparency to distributed programs in C and C++.

   Effort 6 pm

3. A3 Report on Architecture for Automated Transparency

   A description of the semantic information needed about applications to enable provision of automated transparency and the rules for use of that information by automated transparency tools. The links between compile / link-time tools and run-time installation and configuration management.

Effort 10 pm

— research analysis 2 pm

— experimental prototype 5 pm

— synthesis 3 pm

4. A4 Report on Nucleus Interface and Engineering Architecture

   A description of the nucleus interface and engineering structures which define the target for automatically generated code. Guidelines for instantiating the architecture for different granularity of object.

   Effort 8 pm

5. A5 Instantiation of Nucleus Interface and Engineering Architecture for CORBA

   An instantiation of the Nucleus Interface and Engineering Architecture of A4 layered above distributed object environments based on the OMG CORBA specification.

   Effort 12 pm

   — familiarization with CORBA 4 pm

   — prototype 8 pm

6. A6 Instantiation of Nucleus Interface and Engineering Architecture for a real-time environment.

   An instantiation of the Nucleus Interface and Engineering Architecture of A4 layered above distributed object environments based on a real time environment.

   Effort 12 pm

   — familiarization with Real Time Operating System 4 pm

   — prototype 8 pm

7. A7 Report on Instantiation of Architecture for Automated Transparency for C++ Applications

   A tools framework based on the architecture of task A3, to support automation of selective transparency for C++ applications written according to the guidelines of task A2.

   Effort 10 pm

8. A8 Report of Automated Transparency for Dependability

   Description of the use of the results of A7 to provide tools automating transparencies in support of dependability.

   Effort 8 pm

9. A9 Report of Automated Transparency for Performance and Timeliness

   Description of the use of the results of A7 to provide tools automating transparencies in support of performance and timeliness.

   Effort 8 pm

8 Federation

8.1 Approach

This focus is concerned with how ANSA can be used to build distributed applications that span organizational and technological boundaries; how such federations are configured and managed, and the interface between ANSA and other technologies. Many of these questions are implicit in other milestones, they are drawn out here to ensure they are covered coherently. The aim is to ensure architectural integrity.The broadest possible range of issues will thus be identified. Examination of the issues will however be constrained by the limit placed on the effort for this focus.

The use of distributed systems technology to couple existing applications is necessarily a piecemeal process. Islands of distributed computing are growing, each under autonomous control and possibly using different technologies (or different evolutions and parameterisations of the same technology). In due course these islands will become interconnected as the applications they support are integrated, but this federation must not compromise the autonomy of its components. The purpose of ANSA work in this area will be to investigate means of negotiating and managing a federated system and architecting the systems required to support federation.

Many different kinds of boundaries can exist between federated domains. These will be identified and classified and the issues that must or can usefully be addressed at these boundaries (such as naming, scoping, typing, accounting, auditing, security[1], transformation, etc.) will be explored. Particular attention will be paid to the issues raised by the increasing prevalence of mobile computing devices which wander between federated domains, including both those whose movements can be tracked and those which disconnect for extended periods and reconnect at unpredictable locations.

The application which assists in the management of existing systems (such as telecommunications networks and process control systems) is itself a distributed application. If such a management application is to be developed within the architecture, it must be possible to represent and manipulate the resources and components of the existing system in the distributed environment. The purpose of ANSA work in this area will be to assimilate into the architecture the management interfaces and management components by which distributed applications can monitor, control and configure external resources. The ODP work on streams is an initial attempt to explore this area.

1. Boundaries are particularly important for security: without boundaries, there is no separation and no security. Federation can weaken boundaries and special attention will be paid to the security implications.

8.2 Baseline

- AR.005 The ANSA Model of Trading and Federation (available Feb 1993)
- TR.023 A Framework for Federating Secure Systems
- TR.039 Management in Object-based Federated Distributed Systems
- TR.040 Monitoring Distributed Systems
- TR.eee Management in Object Based Federated Distributed Systems (available Feb 1993)

8.3 Key questions

8.3.1 Modelling federation

- QF1.1: what kinds of boundaries can divide domains in a federation, where can such boundaries occur, how are they scoped and managed and what are the consequences of crossing and moving boundaries?
- QF1.2: what are the issues at each kind of federation boundary and how are they dealt with?

8.3.2 Programming model for federation

- QF2.1: what programming abstractions are required to represent and configure (both architected and non-architected) resources and components in a federated system?
- QF2.2: what are the generic design principles for engineering and managing boundaries in a federated system?

8.3.3 Structured kit of parts for federation

- QF3.1: what engineering and management functions are required to enable interworking across federation boundaries?
- QF3.2: how much of the support for interworking can be automated and what information is needed to enable such automation?
- QF3.3: what are the constraints on reconfiguring boundaries in a federated system and how is such configuration engineered and managed?

8.4 Work packages

1. F1 Technical report on the architecture of federated systems

   Identify all the types of boundaries that may exist between federated systems. What are their functions, problems, constraints and scope rules. What are the problems of enabling or preventing the crossing of boundaries. What are the additional problems of mobile objects which leave and re-enter the federated system at different points.

   Identify all the issues that must be or can usefully be addressed at each type of boundary. What are the engineering options available for dealing with these issues. What are the trade-offs that can be made. How are the management problems and opportunities posed by boundaries. How can these be resolved and what are the consequences.

Particular attention will be paid to the effects of boundaries on names and interface references.

Effort:12 pm

— research survey 3 pm

— synthesis 9 pm

2. F2 Technical report on programming abstractions for federated systems

What programming abstractions are required to represent and manage boundaries, resources and components (both those which are fully architectural and those which are managed architecturally but have specialised non architectural implementations) in a federated system. How can these abstractions be engineered. How are they presented to the applications programmer. What are the management mechanisms and policies. What are the design rules for building federations.

Effort: 9pm

— research analysis 2 pm

— synthesis 7 pm

3. F3 Technical report on the engineering and management of interface references in a federated system.

A complete design for interface references and their associated generators, binders, relocators, traders, interceptors, translators, marshallers, and monitors. The design will take place within a particular scenario, selected by the sponsors.

Effort: 12 pm

— scenario selection 1 pm

— experimental prototype 8 pm

— synthesis 3 pm

4. F4 A prototype implementation of a network of interceptors

This should include both data and telecommunications services over wide and local area networks. The prototype should be targeted at a particular field trial.

Effort: 15 pm

— familiarization with LAN/WAN equipment and software 3 pm

— experimental prototype 9 pm

— synthesis 3 pm

9 Standards

9.1 Approach

The objectives are:

1. to maintain alignment of the ISO ODP Reference Model with ANSA
2. to bring architectural integrity to the work of industry standards organisations, such as OMG, OSF, UI.

The objectives will be achieved by:

- alerting sponsor companies of potential conflict between de-facto and de-jure standards work, before standards are frozen
- planning future input to standards groups to initiate convergence of their work including:
  - — contribution of the architecture to ISO/IEC JTC1 SC21 WG7
  - — editorship of ISO ODP Reference Model
  - — contribution of the architecture to industry standards, as directed by the Management Committee
  - — providing “backcloth” presentations at the time of calls for contributions or submissions (e.g. OMG).

9.2 Work packages

The following are work packages for which ANSA Team members have specific management roles:

1. S1 ISO/ODP standardisation

   Effort: 6 pm/yr
2. S2 OMG Common Object Request Broker

   Effort: 6 pm/yr

Preparation of standards contributions and reviews of standards proposals has been factored into each of the other work packages.

10 Project Management

10.1 Approach

The plan includes 2 py/yr effort for the project management function provided by the Project Director and the Chief Architect, as defined in the Third Schedule of the Programme Management and Execution Agreement.

11 Glossary

AR: Architecture Report

BR: Briefing Notes

CORBA: Common Object Request Broker Architecture

ISO: International Standards Organisation

ODP: Open Distributed Processing

OMG: Object Management Group

OSE: ISO Open Systems Environment

OSF: Open Systems Foundation

RC: Request for Comments document

RM-ODP: Basic Reference Model for ODP (ISO/IEC 10746 or CCITT X.901 - 904)

TR: Technical Report

UI: Unix International

12 Detailed planning material

Attached are:

- Work package dependency (Pert) chart
- Gantt chart
- Delivery schedule (work package start/finish dates). The time schedule beyond 28/2/94 is provisional.
- List of activity dependencies
- Resource allocations for work packages, for the period to 28/2/94 (to be provided)

Note: These are provided separately; they are not an integral part of this document as they are outputs from AutoPlan, which cannot be imported into FrameMaker. If the material is missing, please apply to the management of the Phase III work programme.