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

Andrew Herbert

Abstract

This slide presentation describes the vision, objectives, focus, results and benefits of the ANSA Phase III Workprogramme for 1993-4.

APM.1054.00.02 **Draft** 2 September 1993

Briefing Note

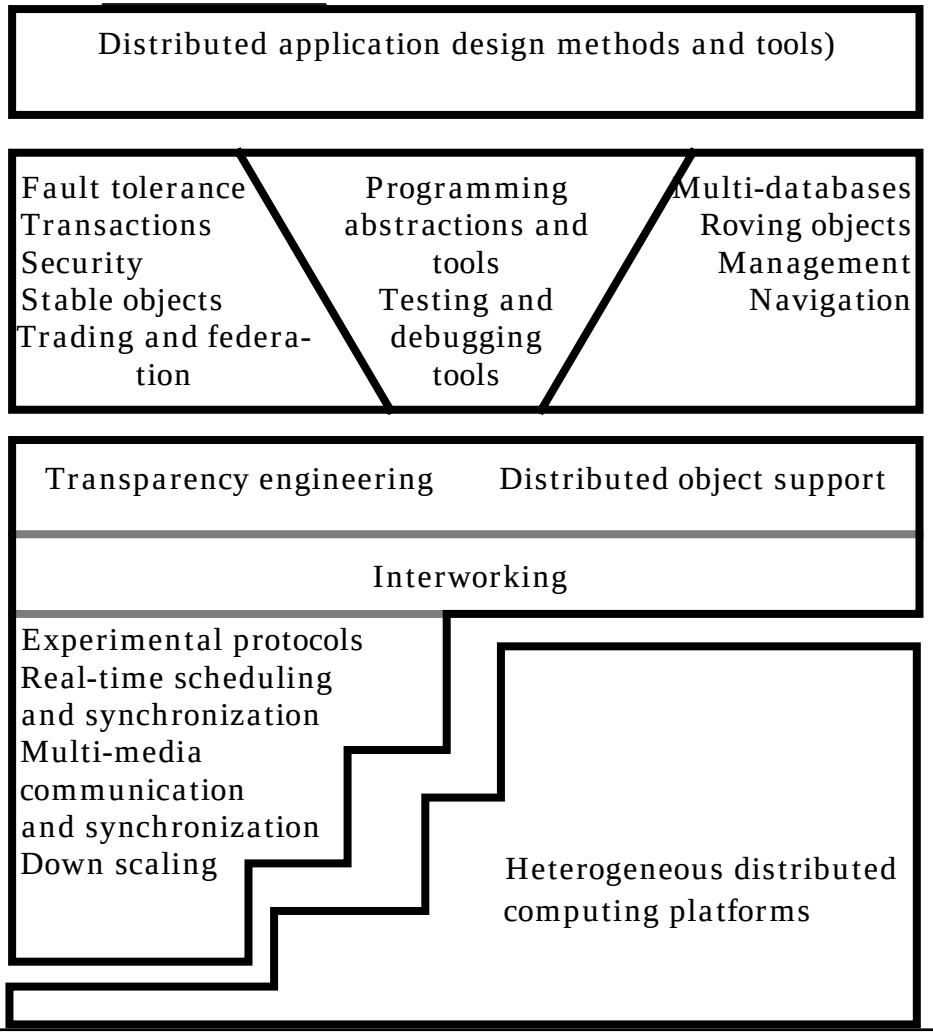
Distribution:
Supersedes:
Superseded by:

ANSA Phase III

- **ANSA has established itself as the framework for advanced development of Open Distributed Processing technology and standards**
- **Jointly agreed programme of research and development between sponsors**
- **Clear distinction between architectural design versus proof of concept in robust prototypes (e.g. ANSAware)**
- **Validation in applications done by sponsor's field trials**
- **Work with teams in sponsors on requirements, architecture and prototypes**
- **Focus on International Standards (ODP) for architectural framework, industry standards (OMG, OSF, UI, X/Open) for technology**
- **Strong input on technology from vendor sponsors**
- **Strong input on requirements from user sponsors and telecommunications sponsors**

The market

- **ONC, DCE and CORBA have put distributed computing in the mainstream**
  - **replication, transactions, checkpoint/restore, migration are missing**
  - **there are no application-building tools and no software configuration aids**
  - **there is no design process for distributed applications**
  - **there is no capability for "data integration" and "system management"**
- **Distributed systems formed by federation of diverse, autonomous systems**
  - **vendors will solve the inter-operability problems, with help from standards**
  - **standards are necessary to deal with the evolution and migration problem**
- **Downsizing, and open systems**
  - **use distributed computing abstractions to open up OS structure**
- **implement distributed computing within the OS to gain performance and scaling**
  - **limiting case is silicon implementation**
- **Real-time issues must be addressed**



Five Year Vision

Primary Themes

Enterprise-driven Computing

Selective dependability

Performance, Real-time and multi-media

Delivery

Architecture

Prototypes

Approaches

Programming abstractions

Modular engineering toolkit

Automated transparency provision

Federation

Objectives 1993-4

The overall scope is broad, so for first two years we have narrowed down on two areas of work in which to develop the architecture, and the capabilities required on the results we produce

- **Continue and complete work on *dependability* and *performance***

ANSA Phase III (ISA) included work that led to a transaction model, a replication model and the beginning of a model for real-time event handling. This initial work is ripe for taking to completion

- **Influence industry consensus and systems software suppliers**

Important standards like DCE and CORBA are light on dependability and performance functions: the vendors want to ad these capabilities and the users are demanding them. There is a window of opportunity here

- **Emphasize *automated, selective transparency***

Want to be able to wrap up applications with dependability attributes transparently; want to be able to select different guaranatees for different applications.

- **Emphasize *federation***

Want to understand how to interwork between systems that make different transparency choices, different technology choices

Key Areas of Focus

Applications Developers

SYSTEM MANAGEMENT

event notification

PROGRAMMING MODELS

REPLICATION ATOMICITY PERFORMANCE

AUTOMATED TRANSPARENCIES

groups

atomic infrastructure

message delivery guarantees

scheduling guarantees

CORBA
DCE

real-time kernels
high speed networks

Benefits

- **Elimination of the need for closed, special purpose and expensive distributed processing infrastructures**

Make dependability and performance components part of the commodity distributed processing technology pool - brings greater freedom to users in making technology choices and lowers integration costs

- **More distributed applications written using standard interfaces**

Since support for dependability and performance will be more available, and better supported than hitherto - enhances revenue stream from open distributed processing technology for vendors

- **Higher productivity for programmers**

Because of easier to access functionality, and hence reduced application costs and time to market.

Results

- **An integrated architecture including dependability and performance**

Integration is crucial - solutions to dependability exist, but not as elements of a single architecture

- **Concepts and taxonomy**

Terminology and models for expressing both requirements and solutions, so that the capabilities and limitations of each part of the architecture are clear

- **Engineering model**

A set of functional components and recipes for combining them to build practical systems

- **Programming model**

Programming abstractions that allow the use of selective transparency

- **Automated transparencies**

Tools to automatically wrap applications with dependability and performance capabilities

- **Management and federation**

Guidelines on what do do at boundaries of dependable / time critical systems

Automated transparency

- **Applications are customer specific**
- **They can be built using commodity software and then customized**
- **Trend towards application toolkits and application generators**
- **Aim is reduced development time and manageable code**
- **Extend ANSA computational model**
- **Show how to deliver abstractions - preprocessors, inheritance, extended languages, specialized languages**

Dependability and Performance

- **A range of options and solutions**
- **Who recovers from errors - client, server or both?**
- **Performance requirements may constrain choice of solution**
- **Aim is to find the parameters of the range of options and architect the solutions needed to meet them**

Federation

- **Distributed processing is seen as a means to couple application areas**
- **Coupling may span organizational (i.e. policy) boundaries**
- **Coupling may span technical boundaries**
- **AIm is to understand and parameterize the kinds of boundaries there can be**
- **Investigate tools to build the interceptors and adaptors required at these boundaries**

Application

- **Architectural input to international and industry standards to achieve industry consensus**
- **ISO/CCITT Open DiIstributed Processing and Object Management Group identified as key players**
- **Architectural input to sponsor's engineers in developing products and applications**
- **Field trials as focus**
- **Prototypes for use by sponsors to test architecture out in specific applications and generate feedback**
- **Prototypes for use by ANSA team to test ideas and gain understanding of design and implementation issues**

Sponsor Involvement

- **Management Committee: strategy, budgets, resources**
- **Technical Committee: focus, quality, technology transfer overall, standards**
- **Topic Review Groups: working with the ANSA team members in depth**
- **Field trials: joint between sponsors**
- **Secondees: in the ANSA team**
- **Workshops: both on requirements, and results**
- **Consultancy: helping individual sponsors use ANSA**
- **Reports: the audit trail and repository of knowledge**

Dependability questions

- what are the parameters of dependability requirements?
- what are the meaningful combinations of parameters?
- what are the basic functional components of group and atomic infrastructures?
- what are the issues in federating systems that have different dependability infrastructures (e.g. X/Open XA)?
- what are the trade-offs between application complexity and transparency?

Performance Questions

- what are the parameters required to express timing constraints, performance guarantees and resource utilization constraints?
- how do performance requirements relate to dependability requirements?
- what operating system and network functions are required to enable applications to reserve and/or monitor resources
- what programming abstractions are needed to react too time-constrained events?
- how can guarantees be negotiated across networked systems from end-to-end

Automated Transparency Questions

- **what means are there for packaging abstractions for distributed applications and their respective strengths and weaknesses?**
- **what engineering aspects of distribution can be automated and to what degree?**
- **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)?**
- **how can transparency tools be decoupled from applications languages?**
- **what links between applications programming tools and runtime services must exist to ensure end-to-end consistency of mechanisms and protocols?**

Federation Questions

- **what kinds of boundaries can divide domains in a federation?**
- **what programming abstractions are required to represent and configure (both architected and non-architected) resources in a federated system?**
- **what are the generic principles for engineering and managing boundaries in a federated system?**
- **how much of the support for interworking across boundaries can be automated, and what information is needed to enable such automation?**
- **what are the constraints on reconfiguring boundaries, and how is such change managed?**

Dependability workpackages

- **D1 Report on Modelling Dependability Requirements (11/93)**
- **D2 Report on Programming and Management Model for Dependability (11/93)**
- **D3 Report on Egineering Model for Dependability (5/94)**
- **D4 Implementation of Dependability for CORBA (2/95)**
- **D5 Optimised Implementation of Dependability (2/95)**

Performance and Timeliness

- **P1 Report on Modelling Performance and Timeliness Requirements (10/93)**
- **P2 Report on Programming and Management Model for Performance and Timeliness (11/93)**
- **P3 Report on Engineering Model for Performance and Timeliness (6/94)**
- **P4 Performance and Timeliness for CORBA (2/95)**
- **P5 Optimized Implementation of Performance and Timeliness (2/95)**

Automated Transparency

- **A1 Report on Approaches to Automation of Transparency (8/93)**
- **A2 Report on Structuring Applications for Distribution (12/93)**
- **A3 Report on Architecture for Automated Transparency (12/93)**
- **A4 Report on Nucleus Interface and Engineering Architecture (4/94)**
- **A5 Implementation of Nucleus and Engineering Model for CORBA (9/94)**
- **A6 Real-time implementation of Nucleus and Engineering Model (11/94)**
- **A7 Report on Automated Transparency for C++ (4/94)**
- **A8 Report on Automated Transparencies for Dependability (4/94)**
- **A9 Report on Automated Transparencies for Performance (4/94)**

Federation

- **F1 Report on the Architecture of Federated Systems (12/93)**
- **F2 Report on Programming Abstractions for Federated Systems (7/93)**
- **F3 Report on Engineering and Management of Interface References in a Federated System (6/94)**
- **F4 A Prototype Implementation of a Network of Interceptors (2/95)**

Technology Baseline

- **operating systems: POSIX for open systems, a microkernel / real-time OS for real-time and down scaling**
  - **interworking between these essential**
- **RPC technology: OSF DCE protocol and data encodings, OSF DNS name server**
- **interface definition and runtime: OMG CORBA**
- **goals:**
  - **leverage available technology as it becomes available**
  - **ANSA-based applications accessible to CORBA and DCE clients**
  - **leave options open to track alternative technologies**
  - **enable sponsors to feed back technology into standards process**