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

Distributed Control (TC presentation June 94)

Dave Otway

Abstract

The two year plan for ANSA Phase III needs updating after the first year has passed. Much of the current work needs refocusing so that the business problems of our sponsors are seen to be addressed more directly.

The document APM.1204 provides a focus for the remaining activities in the original Phase III plan and sets these activities in the context of a large scale information control and provisioning scenario. It proposes a strategy for further development of the architecture through prototyping and animation.

This presentation discusses those sections of APM.1204 concerned with the development of the architecture to provide distributed control of information and information services networking.

APM.1247.00.01 **Draft** 25 April 1995

Request for Comments (confidential to ANSA consortium for 2 years)

Distribution:
Supersedes:
Superseded by:

Distributed control of information and information services networking

(APM.1204)

Dave Otway

Market Drivers

- **A market for interactive, wide area access to multi-media services is being created by:**
  - **increasing public access to networked information services**
  - **development of electronic commerce**
  - **increasing use of multi-media presentation for education and entertainment**
  - **broadband networks delivering increased bandwidth to offices and homes**
  - **development of broadband network interfaces to workstations and PCs**
  - **use of open systems technology in embedded systems**

Challenges

- **The mips, memory and bandwidth are available,**
- **but the software will hinder their exploitation until:**
  - **dynamic configuration and management replace static, pre-planned approaches**
  - **transparent interworking is possible between (most) distribution platforms**
  - **all applications have access to broadband communications**
  - **applications can coordinate different media flows across the network**
  - **applications are guaranteed enough processing resources to meet user's QoS requirements**

What are we doing already ?

- **applications interface**
  - synchronous programming extensions for streams and signals
  - explicit binding model
  - Quality of Service statements
- **infrastructure**
  - resource separation and independent scheduling
  - binding engineering
  - Quality of Service engineering
  - timed RPC
  - flexible multiplexing
- **prototype**
  - real-time ANSAware over OSF/1

What else is available ?

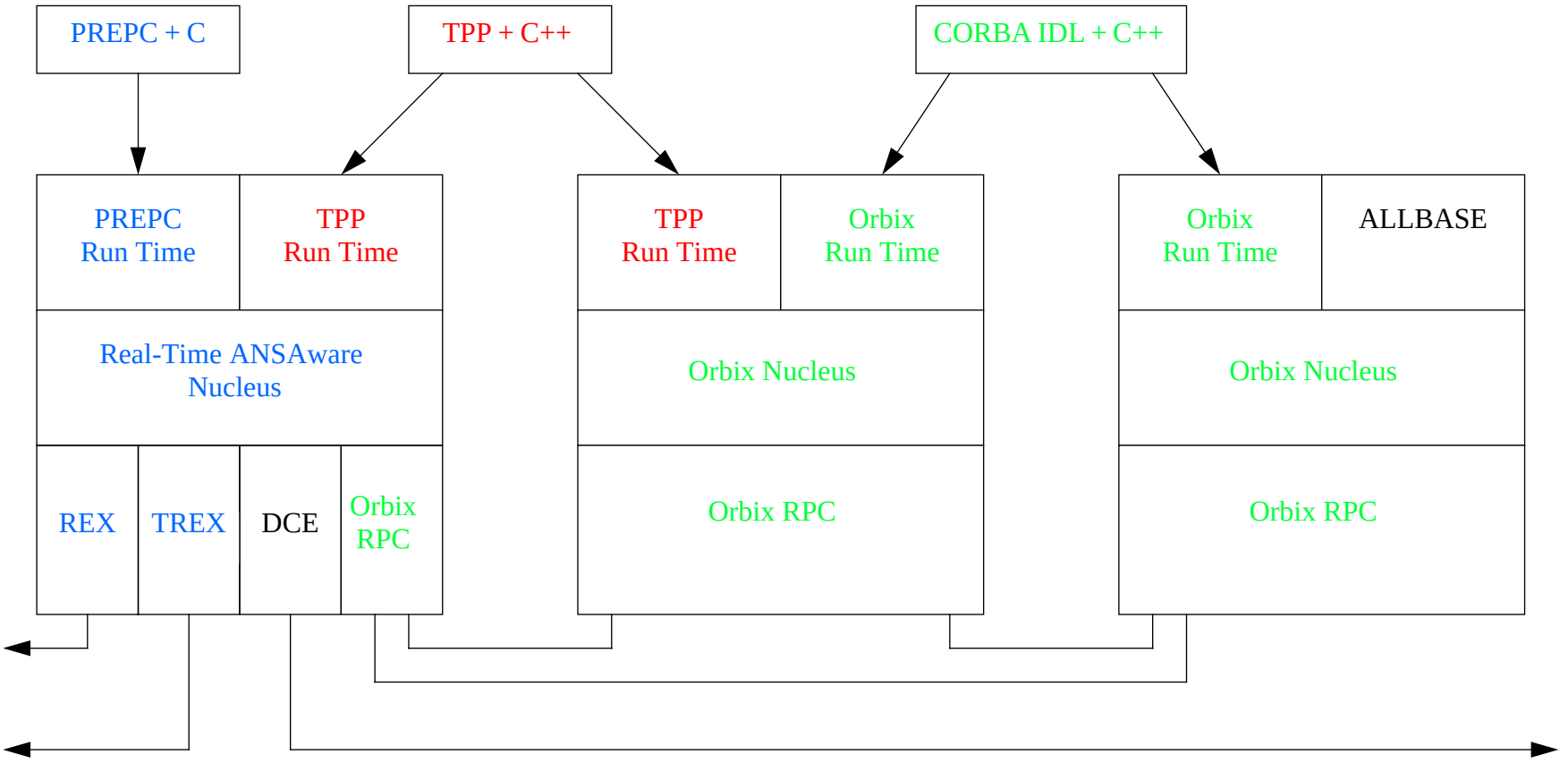
- **real-time programming**
  - Esterel compiler and finite state machine (also Reactive C)
  - CNET Quality of Service model
- **distributed programming**
  - OMG CORBAs (already using Orbix)
  - TPP (TINA Pre-Processor) and C++ run-time system
  - multi-view active Abstract Syntax Tree and type checker
  - TINA IDL C++ stub generator
  - DPL compiler
  - PREPC / STUBC
  - Tcl / Tk / [incr Tcl] / Tcl-DP prototyping environment
- **real-time technology**
  - OSF/1, HP-UX? LynxOS? Vertix? Chorus?

What else do we need ?

- focused application [dependable electronic marketplace ?]
- access to suitable technology [ATM ?]
- integration of dependability with performance
  - timeliness failure detection
  - bounded recovery mechanisms
  - high performance redundancy management
  - high performance recording mechanisms
- support services for distributed services management
- platform interoperability
- feedback from development experience

Prototyping Strategy

- provide non real-time services over a CORBA platform (Orbix ?)
- prototype own real-time platform to keep control over real-time engineering
  - re-engineer nucleus for o-o multiple protocol stacks, real-time protocols, explicit binding, QoS engineering, resource separation, independent scheduling & dependability mechanisms to work over any suitable real-time technology
- make real-time platform interwork with CORBA platforms
  - advanced trader, repository and interceptors
- program development tools must map onto both platforms
  - construct tools around multi-view active AST and type checker
  - jointly develop asynchronous C++ code generator and run-time with TINA-C
  - develop synchronous extensions to code generator and run-time based on Esterel
  - source language: TPP with synchronous extensions
  - temp expedient: graft DPL parser onto new AST & add synchronous extensions


PREPC + C
TPP + C++
CORBA IDL + C++
PREPC Run Time
TPP Run Time
TPP Run Time
Orbix Run Time
Orbix Run Time
ALLBASE
Real-Time ANSAware Nucleus
Orbix Nucleus
Orbix Nucleus
REX
TREX
DCE
Orbix RPC
Orbix RPC
Orbix RPC

Benefits

- initially targeted at TINA-C laboratory work and auxiliary projects

- **results then transferable to the development of:**

  - interactive multi-user, multi-media desktop applications

  - electronic market place

  - telecommunications service management and network management

  - open real-time control systems in command and control, process control and manufacturing automation

Risks

- **alignment with TINA-C**
  - **conflicts over IPR and/or timescales**
  - **rejection of TINA-C results by telecommunications industry**
  - **divergence of ANSA and TINA-C concepts or work programs**
- **although the architecture is being “extended”, the prototyping will involve fundamental changes to platform internals**
  - **the prototype code may not be as transferable into products as the architecture and lessons learnt**
- **dependability aspects have not been fully considered**