Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: Cambridge (01223) 515010
INTERNATIONAL: +44 1223 515010
FAX: +44 1223 359779
E-MAIL: apm@ansa.co.uk

ANSA Phase III

DIMMA BNR presentation

Guangxing Li

Abstract

This document provides an overview of the ANSA Distributed Interactive Multimedia Architecture for BNR ANSA workshop.

The slides describe the ANSA work-in-progress on an open distributed multimedia systems architecture. It examines the problem space and technology bases. An integrated system architecture is suggested and the benefits of the architecture are presented. It then goes on to outline the important progress of the ANSA phase 3 project in extending the ANSA architecture for real-time and multimedia processing.

APM.1399.01.02 **Draft** 7th April 1995

Briefing Note

Distribution:
Supersedes:
Superseded by:

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

ANSA Distributed Interactive Multimedia Architecture

(or how to extend CORBA for real-time and multimedia processing)

Guangxing Li

gxl@ansa.co.uk

In this session

- Drives
- Challenges
- Scope
- Approach
- Technologies
- Design progress
- Engineering - ANSAware/RT
- Work strategy
- Engineering - current state
- Benefits

Market drives

- public access to interactive information and services
- increasing use of multi-media technology in daily life
- increasing bandwidth available with the deployment of B-ISDN networks
- open systems technology in embedded systems (e.g. telecomms)
- PCs with broadband network interfaces

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
  - applications can coordinate different media flows across the network
  - applications are guaranteed enough processing resources to meet user's QoS requirements

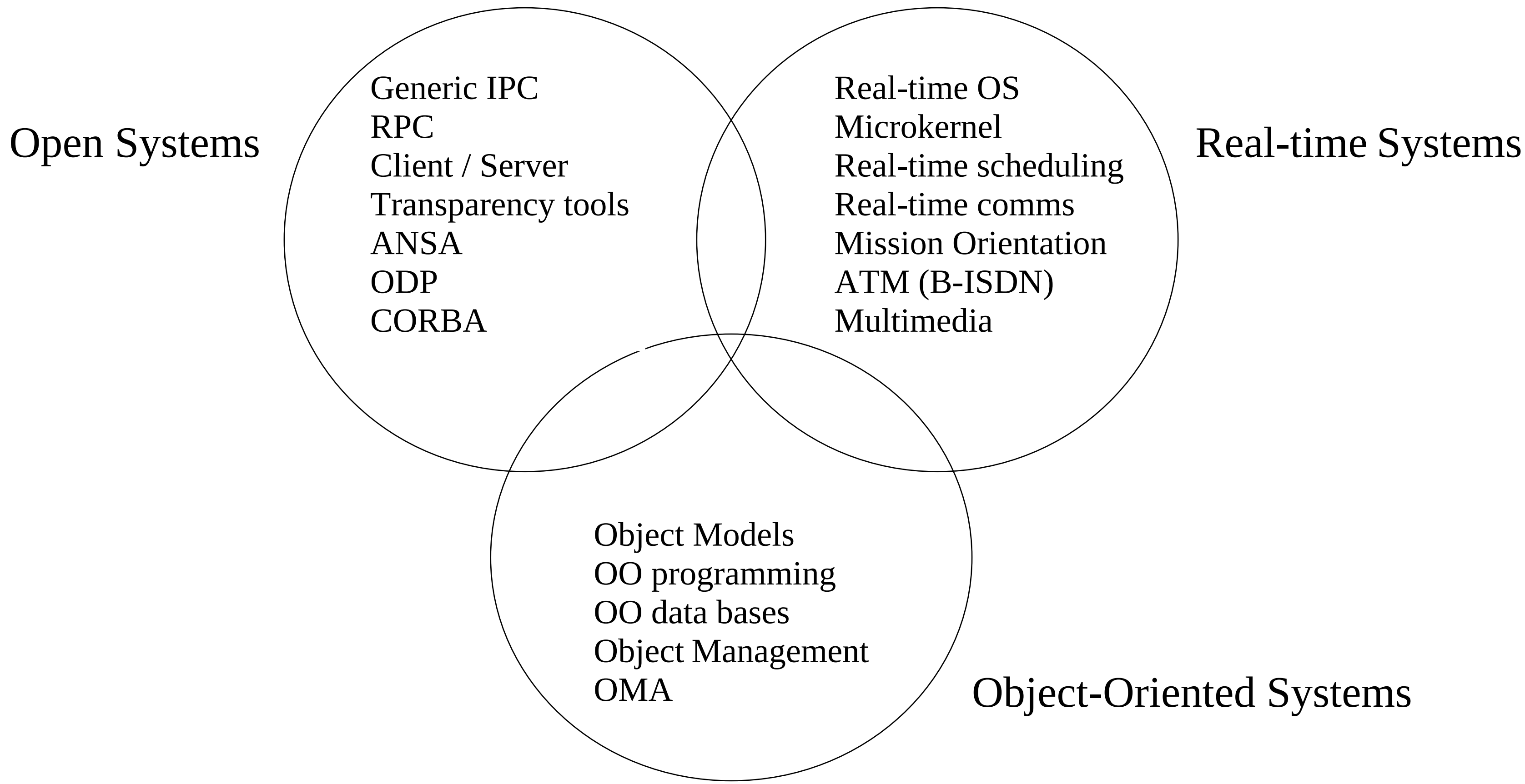
Scope

- what are the multi-media and real-time objects in application programming
- what are the additional abstractions for services management
- what are the mechanisms to be added in the infrastructure for multi-media objects interworking
- what are the abstractions for fine-grained control and monitoring of resources to give QoS

System integration

- real-time (multi-media) objects are *first class citizens*
- a common platform for both real-time and non-real-time objects
- a *TOTAL* system design architecture
- importance of system integration
  - general purpose distributed computing environments are evolving towards real-time systems
  - real-time applications are evolving towards large distributed systems
- real-time applications increasingly need to cross organisational boundaries

Contributory technologies



Design for multimedia and real-time processing

- interaction model: client/server (many-to-one) + end-to-end (stream)
- invocation model: (RPC) call/reply style + (message-oriented) signal passing
- control model: asynchronous + synchronous
- binding model: implicit + explicit
- QoS model: addressing non-functional requirements
- real-time programming model: introducing priority & deadline to distributed processing (or what is a real-time object)
- engineering model

Engineering - ANSAware/RT

- ANSAware/RT 1.0 over OSF/1, HP/RT, LynxOS
- compatible with ANSAware 4.1
- running over the real-time POSIX thread standard
- full p-thread scheduling and threading capabilities
- selective communication multiplex
  - QoS specification and explicit binding operations
- application controlled resource allocation
- supporting the ANSA real-time programming model
- interworking between different real-time platforms

ANSAware/RT (1)

- scheduling points
- real-time scheduling: preemptive priority-based scheduling
- multiple scheduling policies
- real-time tasks and threads
- thread scheduling: policy/mechanism separation

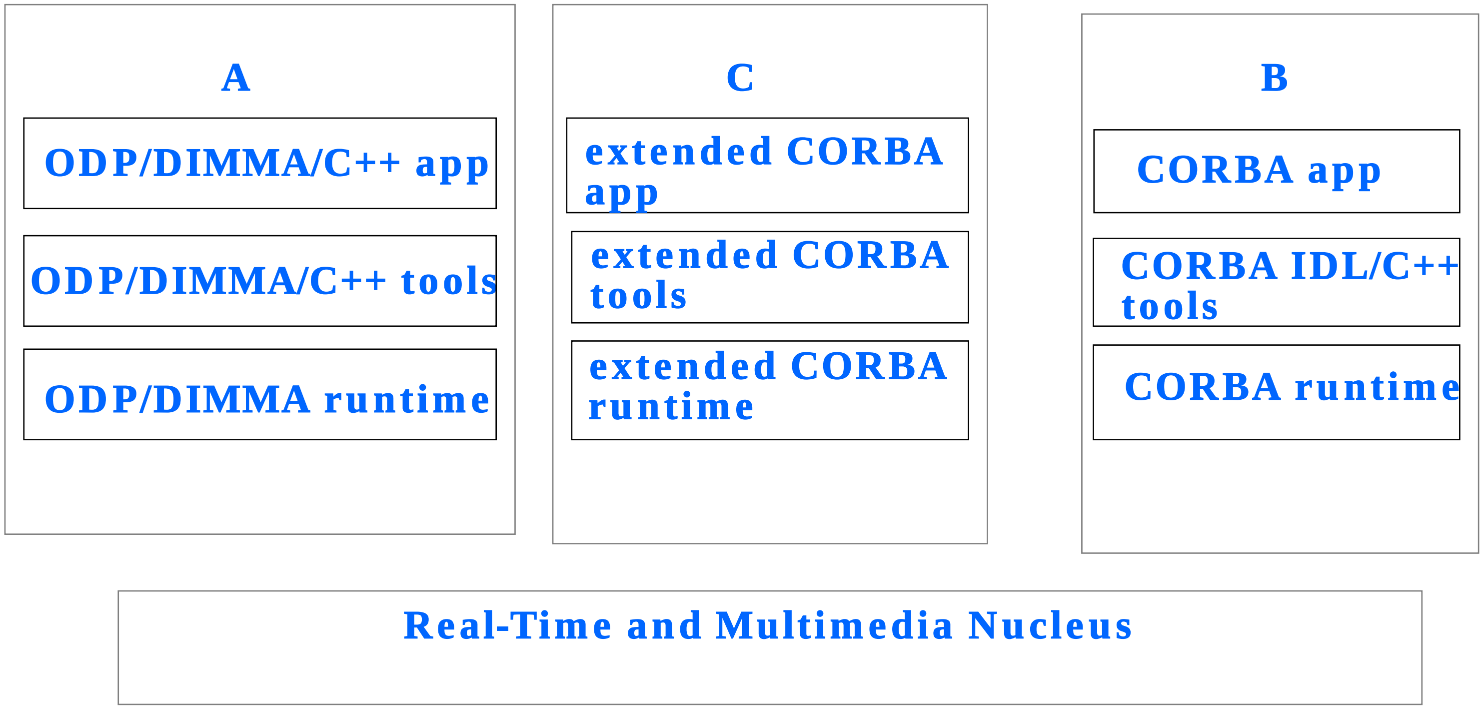
ANSAware/RT (2)

- generic communication stack: multiple execution (RPC) protocols
- real-time RPC protocol
  - understand priority and deadline
  - support distributed scheduling
  - enforce consistent semantics of deadlines
- flexible communication multiplexing based on interfaces
- QoS driven explicit binding

Performance: Distributed Hartstone Benchmark

- synthetic benchmark: measure the overall real-time performance from application points of view
- evolved from Hartstone (testing single machine Ada run-time systems)
- to evaluate better schedulability rather than individual low-level operations
- model based on typical client/server interaction
- four sets of experiments
  - DSHcl: communication latency
  - DSHpq: priority queuing of messages
  - DSNpp: protocol preemptivity
  - DSHcb: communication bandwidth

The next steps



Programming model goals

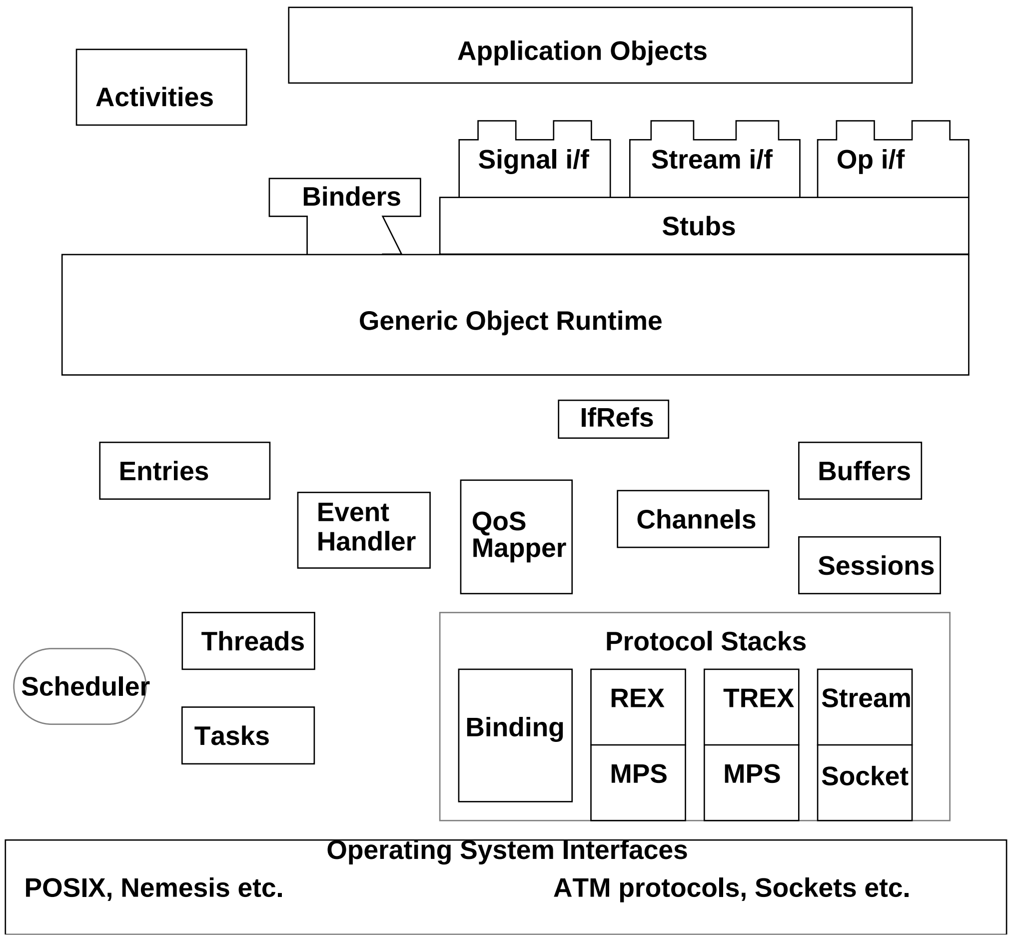
- a complete set of constructs for the extended computational model
  - explicit and implicit binding
  - synchronous and asynchronous expressions
  - operational interfaces and stream/signal interfaces
- strong type checking
- access and location transparency
  - enable local optimization
- selective resource transparency
  - controlled scheduling
  - controlled communication multiplexing
- imperative specification of QoS directives

Engineering model goals (1)

- generic framework (interface) enabling different implementations and implementation tradeoffs are possible
  - modular - easy to replace alternative components
  - extensible - easy to add extended functionality
  - scale up (for large applications of many clients and servers) and scale down (for embedded applications which have limited resources)
- scalable and resource-efficient implicit binding
- QoS driven explicit binding
- generic communication architecture for multiple protocol stacks, addressing schemes and communication models
  - x-kernel (UBC and UA) and ADAPTIVE Communication Environment (WU)
- different execution protocols for different style of object interactions

Engineering model goals (2)

- generic QoS specification, conformance check, negotiation and monitoring
- resource separation and independent scheduling
- both real-time and time-sharing scheduling
- allow to map onto any suitable real-time and multimedia technology
- end-to-end communications management
- object-oriented concurrency (fit C++)
  - u-C++ (U.Waterloo), SyncC++ (EPFL, switzerland), ESTEREL
- open to alternative APIs (e.g. specialized languages, preprocessor, library, CORBA etc.)


Application Objects
Activities
Signal i/f
Stream i/f
Op i/f
Binders
Stubs
Generic Object Runtime
IfRefs
Entries
Buffers
Event Handler
QoS Mapper
Channels
Sessions
Threads
Scheduler
Tasks
Protocol Stacks
Binding
REX
TREX
Stream
MPS
MPS
Socket
Operating System Interfaces
POSIX, Nemesis etc.
ATM protocols, Sockets etc.

Current state

- C++ coding templates for the programming model
  - objects, interfaces, operations, terminations etc.
- a multiple platform development tool set
  - an integration of the GNU development tool set and ANSAware development tool set
  - added functionality and reduced complexity
- in progress
  - nucleus design and implementation
  - other distribution tools

Benefits

- allow ANSA/CORBA technology to cover
  - 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