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

Distributing Real-Time ANSA Objects

Guangxing Li

Abstract

A concise paper for an invited presentation for the IEE colloquim on “Distributed systems support for digital audio and video”.

Object distribution has been regarded as one of the most important paradigms for distributed processing since the early eighties. The increasing popularity of digital audio and video applications presents new challenges to the traditional object distribution models. This paper outlines some of important progress of the ANSA Phase 3 project to extend ANSA object model for real-time and multimedia processing. The paper focuses on *why* the extensions are needed rather than *what* are the achieved technical results.

APM.1265.00.01	**Draft**	17 June 1994
	External Paper	

Distribution:
Supersedes:
Superseded by:

Distributing Real-Time ANSA Objects

Distributing Real-Time ANSA Objects

Guangxing Li

APM.1265.00.01

17 June 1994

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

Contents

1	**1**	**Distributing Real-Time ANSA Objects**
1	1.1	ANSA objects
2	1.2	Stream
2	1.3	Signal
2	1.4	Synchronous computing
3	1.5	Explicit binding
3	1.6	QoS framework
4	1.7	Real-time programming model
4	1.8	Real-time ANSAware
4	1.9	Related work

1 Distributing Real-Time ANSA Objects

Object distribution has been regarded as one of the most important paradigms for distributed processing since the early eighties. The increasing popularity of digital audio and video applications presents new challenges to the traditional object distribution models. This paper outlines some of important progress of the ANSA Phase 3 project to extend ANSA object model for real-time and multimedia processing. The paper focuses on *why* the extensions are needed rather than *what* are the achieved technical results. The extensions include:

- a symmetric object interaction model to remedy the traditional asymmetric client/server interaction model, so that applications may access communication endpoints for multimedia streams (the provision of peer-to-peer communication)
- a symmetric object invocation model to remedy the traditional asymmetric RPC style invocation model, so that multimedia frames and real-time signals can be treated as normal object invocations
- a synchronous computation model to provide predictable computing
- a QoS driven explicit binding model for performance guarantee and resource management
- a generic QoS framework for handling the complexity of different QoS domains
- a real-time programming model for extending the traditional real-time computing concepts to distributed processing
- a real-time ANSAware for supporting the above computational extensions.

1.1 ANSA objects

ANSA is an Architecture for Open Distributed Processing (ODP), which provides new ways of thinking about the design and construction of object oriented client/server distributed systems. The overall ANSA framework has been captured in the joint ISO/IEC and ITU-T Basic Reference Model of ODP (RM-ODP) [ISO93].

The core of the ANSA computational model [APM.0001] is the use of *objects* as units of distribution for management and replacement. An object has one or more *interfaces* that are the points of provision and use of services. Interfaces are first class entities in their own right and references to them may be freely passed around the system.

An interface contains a set of named *operations* (i.e. procedures or methods) which defines its type. Interfaces have the usual remote procedure call style of interaction: operations are invoked with a set of arguments and a response is returned. Arguments and results to invocations consist of references to other interfaces. The effect of an interaction is that the client and server share access to the argument and result interface. This model makes each interface an abstract data type.

1.2 Stream

Traditional ANSA object interaction model is asymmetric: any client holding an interface reference can invoke a server object. The server does not have the protection to which client it allows the access.

In terms of communication paradigm, the asymmetric interaction model is a many-to-one model. The server can only reply to a request, and cannot initiate any communication. This is not sufficient for multimedia and real-time processing where end-to-end (one-to-one) communication is a must.

The support of endpoints in real-time ANSA objects is provided by ***stream interfaces*** [APM.1108]. Streams consist of bidirectional data flows. Flows consist of application dependent frames. The source(s) of a data flow must be bounded to its sink(s) explicitly before data can pass along it. Streams can also be bidirectional: a stream endpoint can be both a source and a sink. Computational constructs are provided to read and send frames at stream endpoints. This models a symmetric interaction paradigm in which the traditional client and server distinction does not exist, and the communication happens between two peers.

1.3 Signal

Traditional ANSA object invocations are typical RPC style, which enforces the normal call/reply ordering. For real-time and multimedia applications, the notification of events and signals cannot normally be structured as call/reply style of messages. A more free ordered message passing scheme is necessary.

The support of free ordered messages is provided by ***signal interfaces*** [APM.1108] in real-time ANSA objects. A signal models a call or a response message for a client, and a request or a reply message for a server (In other words, signals are the elements necessary to construct invocations). Each signal has a direction, and has a source and a sink. Multiple signals can be combined together as a signal interface.

Signals can be transmitted, received, monitored and controlled by synchronous expressions as discussed in the next section.

1.4 Synchronous computing

Traditional ANSA object model is asynchronous which is needed to programming large, federated, concurrent, and non-deterministic distributed systems. Real-time systems require predictable access to shared resources and deterministic behaved programs.

In real-time ANSA objects, the reactive model of synchronous systems used by such language as ESTEREL [BOUSSINOT91] is adopted [APM.1108] because it produces deterministic parallel programs whose behaviours are reproducible and whose execution times are predictable.

In a synchronous system, time is divided into a series of logical instants. At each instant, a number of expressions are executed in response to the input signals received since the previous instant. These logical instant are ordered but have no duration. A synchronous expression waits for a certain (combination of signal(s) and is executed in the instant that the signal(s) are received. Typical synchronous expressions include sending a signal, waiting a signal, testing the presences of signals, watchdogs, parallel execution, sequential execution et al.

The synchronous expressions allow the construction of synchronous islands within an asynchronous world.

1.5 Explicit binding

Binding is the process by which an activity in one object establishes the ability to invoke operations at an interface to some other object. A binding establishes and controls the communication sessions involving multiple objects so that their interactions are possible.

Traditional ANSA object model is supported by an implicit binding model which was designed to have good scaling characteristics and to optimise the usage of resources. It uses maximum multiplexing to efficient resource management and provides only one single Quality of Service: interoperation.

Real-time objects (including the service provided by stream and signal interfaces) require predictable (and different) ways of resource allocation, resource scheduling and a wide variety of Quality of Services. Explicit binding operations [APM.1239] are introduced

- to associate QoS with bindings
- to control the time of binding
- to manage/control a binding.

1.6 QoS framework

QoS is a generic mechanism which can express performance requirements for a user, the performance a server provides and the performance constraint of the infrastructure between them. It can also be used to conduct the negotiation of the required performance and the provided performance.

QoS requirements are categorised: for a particular class of application area, a particular QoS domain is required. A universal QoS domain for many applications is unlikely practical.

As QoS are categorised and there are many different QoS domains, it is unrealistic to use the same mechanism for all of the domains. On the other hand, since the ANSA project is interested in a common architecture that can afford many applications, it is logical to work out a framework so that QoS domains can co-exist and relate to each other. A QoS framework provides a common conceptual model for the definition, organization, co-relation, management and engineering of different domains of QoS.

The real-time ANSA object model uses a language-based approach to define QoS domains, QoS expressions, QoS domain supported explicit binding operations [APM.1151]. It allows the association of QoS statements to interface definitions, binding operations and object invocations.

1.7 Real-time programming model

Real-time processing places unique requirements on distributed systems including predictability, programmer control, timeliness, mission orientation and performance. Real-time processing concerns not only how the computational activities are carried out, but also how shared resources are used (i.e. the manner in which contention for system resources is resolved taking into account timing constraints of real-time activities). Traditional real-time systems provide concepts such as priority and deadline to achieve the execution predictability of computational activities.

Real-time ANSA objects defines a real-time programming model [APM.1222] for the same purpose, and extends the traditional real-time computing concepts to distributed processing. It defines various priority-based and/or deadline-based object execution models. The programming model incorporate tasks and communication channels (the two most important resources in real-time distributed computing) as its basic programming components. It synthesises aspects of resource requirements, resource allocation and resource scheduling into an object-based programming paradigm.

1.8 Real-time ANSAware

ANSAware is an implementation of the ANSA computational model and an example of the ANSA engineering model. Traditional ANSAware was designed to scale, to cover diversity, to support federation and to be efficient in resource usage. Real-time and multimedia processing introduces new requirements such as predictable resource access, separation of resource allocation, and making use of existing real-time technologies etc.

A real-time ANSAware is under development, the first version (named RAW 1.0) [APM.1207] is already in operation. RAW 1.0 has achieved the following results:

- compatible with ANSAware 4.1
- running over a de-facto industry standard: real-time POSIX threads
- full p-thread real-time scheduling and threading capabilities
- selective communication multiplex by QoS specification and explicit binding operations
- application controlled resource allocation
- supporting the real-time programming model given in [APM.1222]
- comparable performance to other distributed real-time system environments.

1.9 Related work

Current research at CNET [CNET93], Lancaster University [COULSON92] and University of Kent are all converging a common architecture for distributed multimedia processing relevant to the real-time ANSA object model.

References

[APM.0001]

O Rees, The ANSA Computational Model, Architecture Projects Management Ltd., Cambridge U.K., February 1993.

[APM.1108]

D Otway, Streams and Signals, Architecture Projects Management Ltd., Cambridge U.K., May 1994.

[APM.1151]

G Li, A Model of Real-time QoS, Architecture Projects Management Ltd., Cambridge U.K., March 1994.

[APM.1207]

G Li, Real-time ANSAware (RAW) Version 1.0: Programming and System Overview, Architecture Projects Management Ltd., Cambridge U.K., May 1994.

[APM.1222]

G Li, Some Engineering Aspects of Real-Time, Architecture Projects Management Ltd., Cambridge U.K., May 1994.

[APM.1239]

D Otway, Explicit Binding, Architecture Projects Management Ltd., Cambridge U.K., June 1994.

[BOUSSINOT91]

F Boussinot and R Simmone, The ESTEREL Language, Proc. of the IEEE, Vol. 79, No. 9, September 1991.

[CNET93]

L Harzard et al., Towards the Integration of Real Time and QoS Handling in ANSA Architecture, ANSA Phase 3 Project Report CNET/RC.ARCADE.01, June 1993.

[COULSON92]

G Coulson eta al., Extensions to ANSA for Multimedia Computing, Computer Networks and ISDN Systems, 25, pp 305 - 323, 1992.

[ISO93]

ISO/IEC 10746-3, ITU-TS Recommendation X.903: Basic Reference Model of Open Distributed Processing: Prescriptive Model, (2nd CD draft) June 1993.