



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (0223) 323010
+44 223 323010
+44 223 359779
apm@ansa.co.uk**

ANSA Phase III

TINA Workshop 1993 Keynote Talk: Distributed Objects and Intelligent Network Services

Andrew Herbert

Abstract

A 30 minute keynote presentation for the 1993 TINA Workshop, L Aquila, Italy, September 1993.

The presentation discusses issues arising in the development of intelligent network services and the way in which distributed object technology can support such developments.

THE ISO Basic Reference Model for Distributed Processing is used as a framework for analysing and example service.

The OMG CORBA and object services are identified as good foundations for service implementation.

The ANSA work on dependability and performance is identified as technology that fits into the OMG architecture, and takes a step towards meeting the needs of the telecommunications sector.

The ANSA work on tools and federation is used to indicate area of service modelling, service design and service deployment where object orientation has a role, but where technology is lacking.

APM.1063.00.01

Draft
External Paper

24 September 1993

Distribution:

Supersedes:

Superseded by:



Object-based Distributed Processing

Andrew Herbert

ANSA Chief Architect

**Technical Director
Architecture Projects Management Limited**



Object-based Distributed Processing

- Previous TINA workshops have addressed **TELECOMMUNICATIONS** and **NETWORKING**
- The goal of TINA is to enable **INFORMATION SERVICES**
- What is an **INFORMATION SERVICE**?
- How are information services **ENGINEERED**?
- What role do **DISTRIBUTED OBJECTS** have in service engineering?
- What **DISTRIBUTED OBJECT TECHNOLOGY** is available and usable?



The Business of Information Services

- **Networked information SERVICES will succeed ONLY IF:**
 - **they can be developed rapidly, to meet market windows**
 - **existing services can be interwork into new services**
 - **services are easy to deploy**
 - **services are easy to manage**



Distributed Object Technology

- **This business model is driving the COMPUTER industry**
 - application integration
 - right-sizing
- **DISTRIBUTED OBJECTS are the foundation**
- **The STANDARDS for Distributed Objects are in alignment**
- **The TECHNOLOGY for Distributed Objects is coming on stream**
- **The TELECOMMUNICATIONS industry can BUILD UPON and ENRICH the technology**
- **The SERVICE concept is the key to ALIGNMENT of the two industries**



Basic Principles of Distributed Objects

- **Service SPECIFICATIONS must be visible in the network**
 - enables integration and evolution
 - basis for **CONTRACTS** between service providers, service users and service bearers
- **Service IMPLEMENTATIONS must be available from the network**
 - enables re-use and reduction of complexity
 - requires **STANDARDS** for **PORTABILITY, INTERCONNECTION** and **MIGRATION**

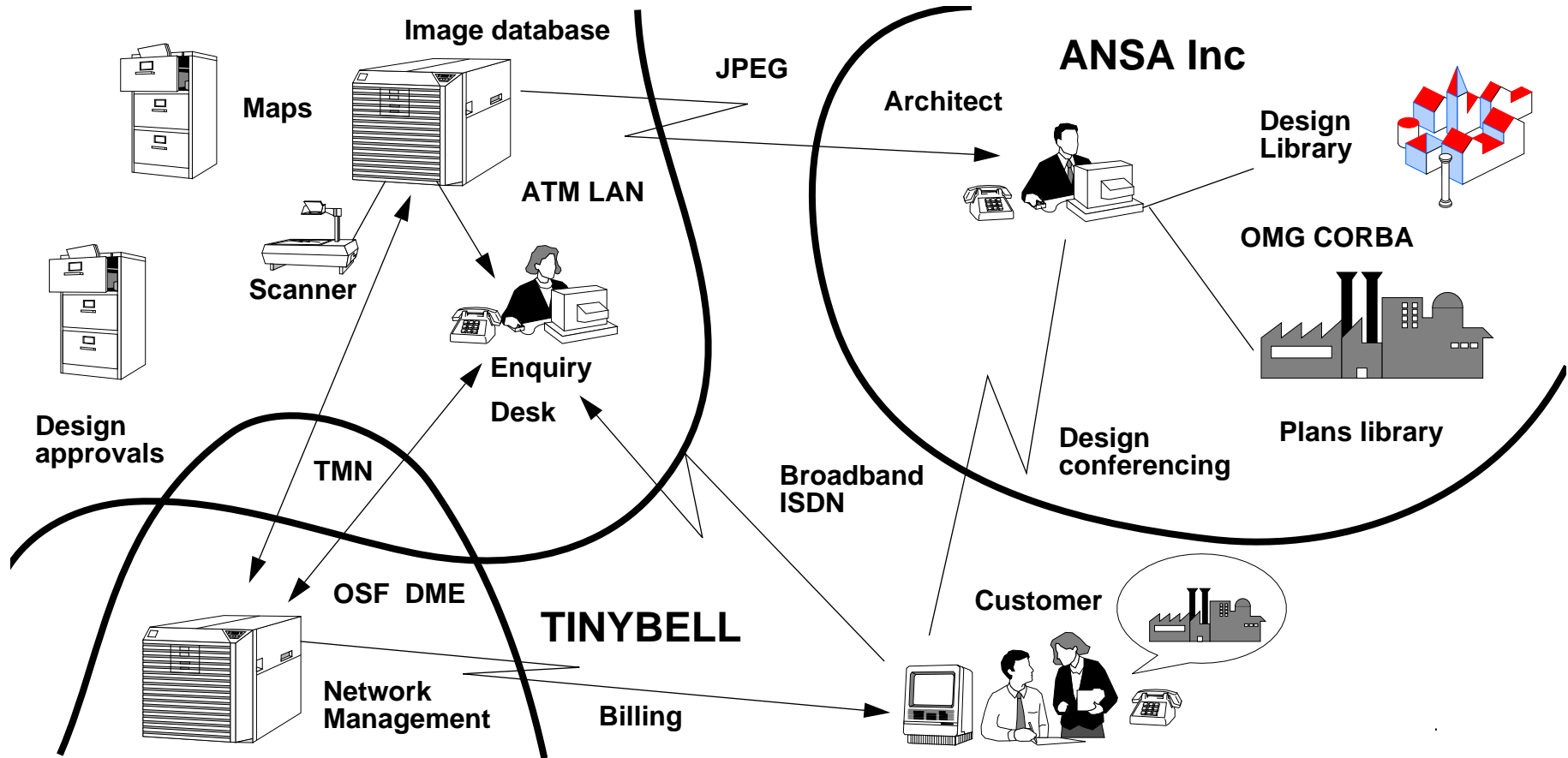


A Scenario - to explore SERVICE ENGINEERING

- **The local authority for Tinaville wants to encourage new housing and high-tech industry to move in**
- **The authority decides to provide electronic access to its Planning and Land Registry Functions**
- **The leading architects in the town (ANSA Inc) develop interactive design conferencing services with their clients**
- **the local authority out sources management of its telecommunications services to the local teleco (TINYBELL)**



TINAVILLE





Service Design

- **How can we analyze the scenario to IDENTIFY potential services?**
- **How can we structure the service DESIGN to be sure that**
 - **all design options are considered?**
 - **all constraints are satisfied?**
 - **designs can EVOLVE to accomodate NEW REQUIREMENTS and adapt to NEW TECHNOLOGY**
- **Use ARCHITECTURAL PRINCIPLES to separate concerns**



ANSA Projections

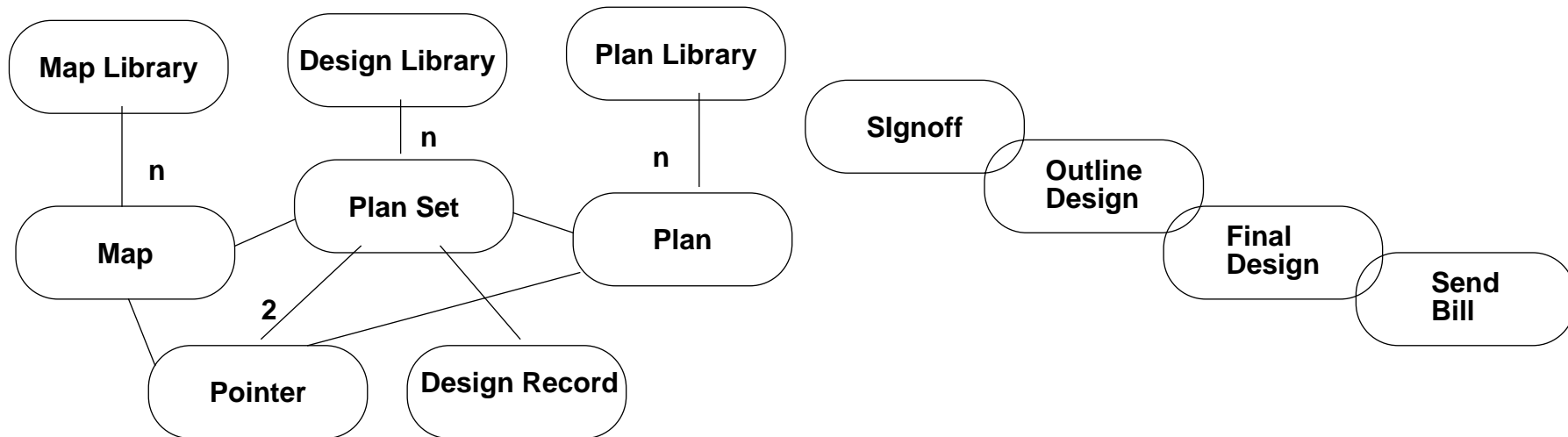
- **called VIEWPOINTS in ISO / ITU-TS Basic Reference Model for Open Distributed Processing (DIS 1994?)**
- **take consistent alternative views focussing on different concerns**
 - **enterprise: policy issues**
 - **information: the domain(s) of discourse**
 - **computational: functional view**
 - **engineering: software and communication structures**
 - **technology: packaging, standards conformance**
- **Let's walk through the scenario, using viewpoints to see where objects can help**



Policy Issues

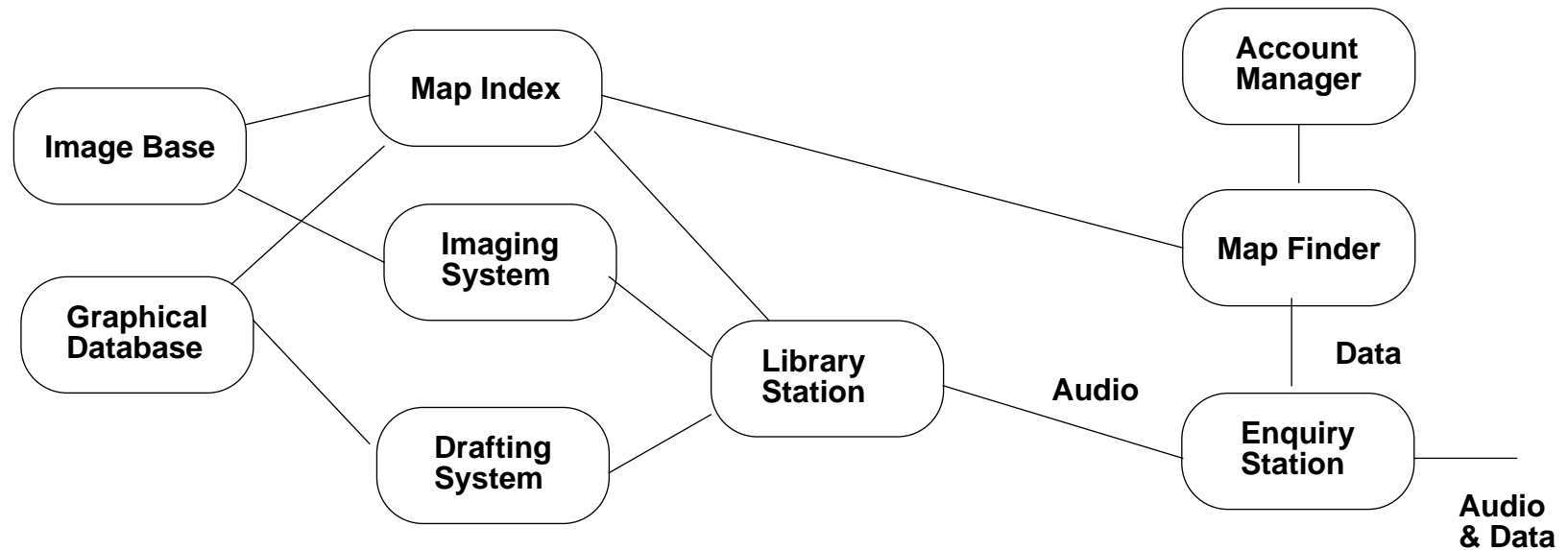
- **WHO uses WHICH information and for WHAT purpose?**
- **What OBLIGATIONS and LIABILITIES have to be respected?**
 - e.g. sensitivity of information
 - e.g. charging model
 - e.g. ease of installation, access, user training, maintenance
 - e.g. regulatory requirements
- **IN systems will have to NEGOTIATE end-to-end CONTRACTS from sets of SERVICE SPECIFICATIONS**
- **Concepts of VALUE, POLICY, NEGOTIATION and SANCTION required to model BUSINESS PROCESSES behind services**
- **OBJECTS help in this kind of analysis because of strong concept of BOUNDARY**

Information Issues



- **Schema for information used by the services as OBJECTS and RELATIONS**
- **View the service as an ACTIVITY changing CLASS MEMBERSHIP of INFORMATION OBJECTS**
- **Information model must be tied to enterprise model so that services assist rather than inhibit the business process**

Functional Distribution



- **OBJECTS** as units of management and replacement
- **INTERFACES** as points of provision and use of **SERVICE**
- **INTERFACES** are **TYPED**, **BINDING** is dynamic and **TYPE CHECKED**
- **OBJECTS** can have several **INTERFACES**

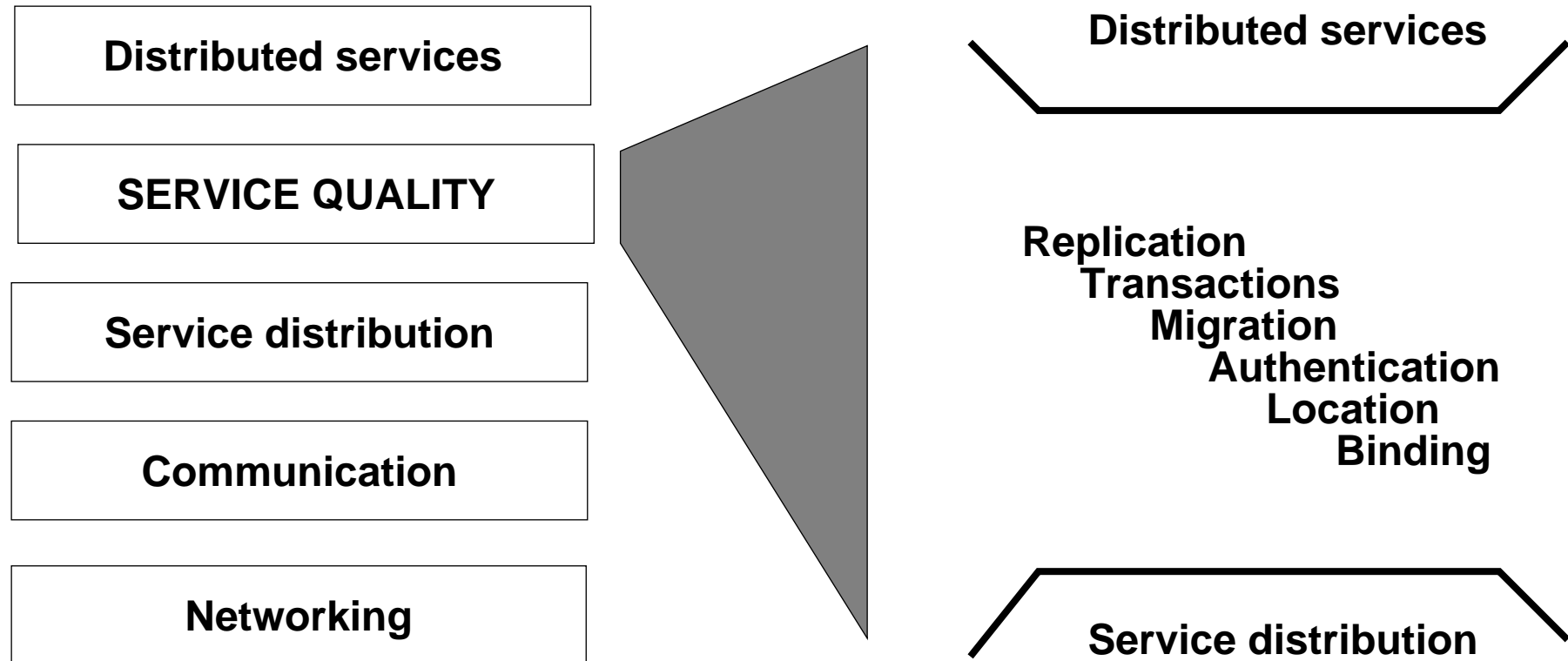


Engineering issues

- **Engineering is about making trade-offs**
- **ABSTRACTION versus SPECIALIZATION**
 - The more you hide, the less control you have
- **CONSISTENCY versus AVAILABILITY**
 - Availability means copies, increases risk of inconsistency
- **AUTONOMY versus UNIFORMITY**
 - Autonomy gives more freedom, but leads to differences which increases complexity
- **SECURITY versus CONVENIENCE**
- **There is no one answer, and therefore there cannot be one ubiquitous DISTRIBUTED PROCESSING ENVIRONMENT**
- **This concept, SELECTIVE TRANSPARENCY, is beginning to spread in the computer industry**



SELECTIVE TRANSPARENCY





Choosing an OO Analysis and Design Tool

- **Range of issues to analyze is large - there isn't a magic methodology**
- **Computer aided design housekeeping is of more benefit than methodology and formal methods**
 - **select tools where object meta model is configurable**
 - **select tools where you can manipulate object specifications**
 - **implies tools based on a OO database**
 - **select OO database where you can program consistency rules**
 - **If the tool generates code templates, make sure you can add your own heuristics**



Federated Design

- **Service designers are separated in location and time**
- **select tools where repository is open and can cross link to other repositories**
- **select tools where TYPE [WHAT] is distinct from CLASS [HOW]**
- **Some (but only) a few tools meet these goals - start using them now**



Distributed Objects - A Snapshot

- **OMG, ODP, ANSA - all converging on a common view**
- **GENERIC service CONFIGURATION, MANAGEMENT and INVOCATION**
 - if it works for one object it works for all
 - Object Request Broker [OMG] as portability target - everything else is a service
 - Trader [ODP/ANSA] - a place to discover available services - a type-safe directory
 - Lifecycle services [OMG/ODP/ANSA] - resource management for services
 - Interface Repository [OMG/ODP] - a place to discover & negotiate contracts
 - Implementation Repository [OMG] - a place to discover service templates
 - Event Management [OMG]
- **MEDIA INDEPENDENT CONFIGURATION and MANAGEMENT functions**
 - interface can include data and telecommunications flows
 - interface binding = connection management
- **Single-address space features of C++ are a stumbling block**
 - dialects of C++ versus distributed object languages?
 - wrapping / proxy techniques for “remoteable” objects



CONCLUSIONS

- You need a framework within which to address **SEPARATE** concerns **CONSISTENTLY**
- The computer industry is laying the foundations of usable distributed object technology
 - but more work is needed on **SELECTIVE TRANSPARENCY**
- **SERVICE ENGINEERING** is highly skilled
 - Service engineers will use **DISTRIBUTED OBJECTS** to increase **PRODUCTIVITY** and reduce **COMPLEXITY**
 - Object Orientation is a **TOOL** to enhance skill not a **MACHINE** to replace it.
- Get a handle on the **BUSINESS** aspects of **SERVICE DESIGN**
 - What are the sensible charging structures?
 - How are the services policed?
 - Who resolves complaints?



Do Objects Really Work?

- **Yes - look at effect of OO techniques on productivity in GUI, CAD, DTP**
- **Yes - look at ANSA applications**
 - **NASA Astrophysics Data System - a distributed multi-database**
 - **GESI Distributed Healthcare Patient Management System**
 - **CTI Distributed Newspaper “Hypertext” Composition System**
 - **AEG OSI resilient network product**
 - **audio workstation conferencing product**
- **Yes - look at success of HP Distributed Smalltalk as tool for building short-lived high-value applications**



The SERVICE MANAGEMENT ENGINE

