Advanced Networked Systems Architecture

ANSA

24 Hills Road
CAMBRIDGE
United Kingdom CB2 1JP

TELEPHONE: Cambridge (0223) 323010
INTERNATIONAL: +44 223 323010
TELEX: 817343 BLUCAM G

Key Architectural Concepts

Abstract:

This document discusses the key architectural concepts of ANSA.

It arrives at these concepts by attempting to integrate existing work on architectural frameworks, object models and formal methods.

This circulation attempts to integrate comments received so far.

As the ANSA Reference Manual progresses, a more detailed description will appear in III,2 and IV.

Number: AO.82.02

Date: 6th March 1987 11:00 am

Area: AO (Architectural Overview)

Status: P (Discussion Paper)

Classification: U (Unrestricted)

Distribution: G (General Public)

Author: AJH (Andrew Herbert)

1 General

This discussion paper leads to a set of dimensions and coordinates for ANSA, in terms of the architectural framework concepts developed in *On The Dimensionality of Architectures* (AO.14) by HJW.

This text uses the terminology developed in AO.14 and in *ARM III, 2 (Architectural Concepts).*

Settling the dimensions and coordinates of the architectures is a matter of urgency to avoid clashes of meaning arising during the drafting of the *ANSA Reference Manual.*

Very little attempt is made to explain the concepts listed in this document; much of the background can be distilled from the references already given. *ARM IV* will be the definitive description. This document is a contribution towards the thinking process.

1.1 Design representations and architecture

Architecture and design are concerned with the description and specification of complex systems. Descriptions have to serve several purposes; for example, to guide the implementor and to convince the customer the system is what he wanted. It is possible to separate out parts of the description that serve different purposes and to use a specialized description technique for each part. The purpose of an architecture is to provide a corpus of knowledge about design and to impose a consistent style, so that different descriptions of systems and parts of systems derived from the architecture are complementary and not contradictory.

The following five level schema shows a schema of tools for describing systems:

- a language for writing down and analysising requirements (viz., the design process and its analysis forms)
- a series of useful design paradigms and a language for describing the structure of designs (viz., Parts III and IV of the *ARM*)
- for each design paradigm, a specification of its design components and the relationships that can hold between these components
- for each design paradigm, an interpretation of its relationships in real world terms
- for each design paradigm, an interpretation of its components in real world terms

The design models correspond to Minsky's notion of a 'frame' - a self-contained body of knowledge with a well-founded method of reasoning for using it.

Since ANSA is an architecture for a wide range of fileds of application, it necessarily contain several frames, corresponding to the various technical skills (such as communications, software and human-computer interfacing), required to build systems. The architecture must knit together these separate frames into a coherent structure, so that it is possible to reason about systems as a whole.

The ANSA view of an **architecture,** is that it should be a basis for organizing the schema discussed above, and can be made up from the following parts:

- a **structural framework** to provide a basis for reasoning about the structure of a system - i.e., the possible **relationships** between the components of a system
- a set of **components** or system building blocks that can be composed and extended to provide complete system designs
- a set of **rules** that constrain how the components may be structured and how structures may be extended
- a set of **recipes** that suggest useful compositions of components to meet particular needs
- a set of **guidelines** that explain how the architectural components and recipes can be applied to meet real needs, and why the rules are reasonable constraints on designs.

The rules explicitly set criteria for conformance.

Further explanation of these parts of an architecture, and their use in ANSA, is discussed in the following sections.

1.2 Components

Component, as used above, is a general term to cover system elements without prejudice to a particular style of design.

ANSA is oriented towards an object-oriented design method and therefore the term **object** is used as a synonym for ANSA component. This can lead to confusion since there are considerable variations between the object model of systems used by hardware designers, software designers and specification designers.

There are however common traits to all the uses of object models in ANSA which are captured in the following definition:

> Objects are independent units of design structure. An object can be removed from a design and replaced by another object with equivalent behaviour but possibly different structure, without changing the behaviour of the system as a whole.

Objects have **properties** other than their behaviour such as their intended role, location and other physical attributes. **Behaviour** is therefore a distinguished form of property.

Structures of objects are described in terms of the **relationships** that exist between the objects in the structure. There are many kinds of relationship possible, such as 'composed of', 'colocated with', 'interacts with' and so on. Relationships may be static or dynamic. A structure of objects is called a **system.** Systems are objects. It may be possible for new objects to appear in a system and for objects to dissappear from a system.

Events are the atomic units of behaviour in a system. An interaction is the occurrence of a related sequence of external events in a system.

The simplest form of interaction is at a **contact**, where all the external events in the interaction are shared by all the objects in the interaction.

In any given system it may not be possible for an object to interact with every other object (i.e. they do not have a common contact). A **medium** is a special sort of object which provides for interactions between the objects that are in contact with it. It does so by governing the relationship between the occurrence of events at sets of contacts. These relationships are called **bindings.** Bindings are established via operations of the medium.

Examples of mediums include shared memory, a communications network, an interpreter, a Birman Bulletin Board.

An **interface** is a view of an object. Thus an **interface specification** is a view of an object specification. If there are events in the behaviour of the object that do not appear (are internal in) the interface specification then a second object that interacts with the first over a binding established with this specification will percieve non-deterministic behaviour. The hiding of events is necessary if objects are to exhibit concurrency.

An **operation** is the consequential behaviour in the rest of a system following an event shared between an object in a system and the rest of that system. In a causal description, the shared event is described as being the **invocation** of the operation. Operations terminate at some event, which may be an external event, or an internal event.

Failures can disrupt the behaviour of objects (including mediums).

It is often the case that an object model may be **transformed** from one paradigm into another.

The differences between the individual object paradigms in ANSA is a matter of the differences in:

- the kinds of event described
- the kinds of interaction described
- the way in which interfaces are specified
- the kinds of medium available
- the causes and consequences of object appearance and dissappearance
- the properties and relationships used

This gives us a simple checklist for the completeness of an object paradigm:

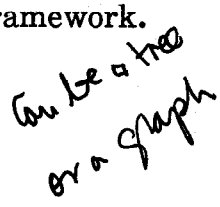
- have all the possible events been described?
- have all the possible kinds of interaction been described?
- has the method of interface description been given?
- have all the possible kinds of medium been described?
- have the causes and consequences of object appearance and dissappearance been described?
- have all the possible properties and relationships been described?
- has a composition rule for objects and mediums been defined?

The **implementation** of an object model will require an **infrastructure**, which includes an **interpreter** that mechanizes objects and interaction and a **language** for representing the structure of interaction events.

The set of object models used in ANSA is discussed in §1.6.

1.3 Framework

The commonality of object models between frames in the architecture is both helpful, since the similar reasoning processes apply in different frames, but also confusing because of the differences in description and interpretation. It is therefore essential to label designed objects with the role they have in each frame. These labels provide a taxonomy for objects made available to the designer through the architecture. The values of key object properties can be used to provide the necessary labelling. These key properties can be thoughout of as the dimensions of a multi-dimensional space, or **framework**, in which objects can be placed and related to one another. The values for each key property provide a set of coodinates for one **dimension** of the framework.



The purpose of the framework is to encourage and enforce the separation of concerns in the design of systems. The framework provides a basis for recognizing independent design choices.

Placing objects in the framework makes explicit the concerns that an object addresses.

The choice of coordinates in the framework is based on engineering decisions, rather than on the basis of fundamental principles. For example the division of the communications frame (§2.1) into seven regions corresponding to OSI layers is arbitrary, but well-accepted.

The particular dimensions of ANSA are described in §2.

1.4 Rules and recipes

The rules constrain the freedom a designer has to make design choices. If the rules are broken in a design, the resulting system is not a **conforming** system, even though the system may be fully operational and meet its design requirements.

Rules state madatory constraints on designs - for example, that every 'executable object' must be 'hosted by' a suitable 'processing object'. Some rules come from fundamental properties of technology and physics, others are merely an artifact of the architectural style.

In addition to the mandatory constraints given by the rules, an architecture may provide some guidance in the form of useful structures that achieve particular effects - for example, a replication structure for increased resilience. These structures can be though of as optional design constraints, and are called **recipes**.

If designers use recipes in their systems, the scope for interworking between systems much better than it would be if designers chose their own structures. Recipes are used in ANSA particularly for transparency mechanisms.

1.5 Guidelines

Without explanation the expression of an architecture as component specifications, dimensions, rules and recipes may be difficult to comprehend . It is therefore necessary to accomodate explanation and advice within an architecture and to stylize the presentation of this information to aid the reader.

Guidelines explain the scope of design freedom implied by particular constraints.

1.6 Analysis and synthesis

A system designer has to start from system requirements and produce a system design that matches those requirements. It is therefore necessary that the designer be able to show how real world concerns are reflected in his design, during design audits.

Therefore accompanying an architecture there have to be some models of the world outside of the mechanized information processing parts of the system, and prescribed ways of incorporating them to the technologically biassed models used in the architecture.

1.7 Object models

A number of object models presently exist within ANSA, and are not always clearly distinguished. The models are listed below with examples of the kinds of concepts found in each model.

Computational model

software modules, processors, instructions, threads, memory.

Information model

Information structures (including documents, databases and filestores).

Communications model

Interrogate, announce, modes etc.

Distribution model

Logical separation and distribution transparency.

Human-computer interaction model

Keyboards, screens, microphones, windows, icons, mailboxes.

Management and administration model

Control functions (including security, resource allocation, resource scheduling, status monitoring, status reporting).

Specification model

Abstract data types and processes.

2 The ANSA Framework

This section discusses the dimensions of the ANSA framework and the associated property whose variation used to position objects in each dimension.

2.1 The communications dimension

The communications dimension is concerned with communications functionality offered by an object to the outside (i.e. non-ANSA) world, and not with any communications inherent in the organization of the object. (Objects that have no communications functionality - such as a directory - are placed at level 7).

The OSI Reference Model provides a suitable coordinate system for positioning an object relative to its communications requirement thus:

Level 1: Physical

Real world objects that perform electrical encoding of data for transmission over a cable and regulate access to a network of physical communication objects.

Level 2: Data link

Objects that manage physical layer objects by the provision of formatting, error checking, addressing and other functions necessary to ensure data transmission across a physical network.

Level 3: Network

Objects that provide routing and relaying between cooperating objects on the same network, or on interconnected networks, thereby achieving a distance-indepent communications service. Network layer objects may also provide for communications enhancements such as flow control.

Level 4: Transport

Objects that provide transparent transfer of data between cooperating objects.

Level 5: Session

Objects that organize and synchronize data exchange between cooperating objects.

Level 6: Presentation

Objects that organize the data encodings used in the communications between cooperating objects.

Level 7: Application

Objects that access or use lower layer communications objects.

Note that an object such as a level 2 to level 2 bridge over a transport service, such as the Xerox X.25 Ethernet to Ethernet bridge consists of two gateway objects that span regions 2 to 4.

2.2 The infrastructure dimension

The infrastructure dimension is concerned with the organization of objects and infrastructures. The coordinates are as follows:

Heterogeneous layer

Components from outside the architecture used to support ANSA systems: e.g., cpus, operating systems, a bus and controllers, etc.

Homogeneous layer

Objects which overlay heterogeneous components to provide an ANSA object interpreter: e.g., abstract machine and nucleus.

Local layer

An object (e.g. a binder, or a communications object) which works on behalf of a particular interpreter.

(Local objects do not change the language of the interpreter: local objects enrich its functionality).

Global layer

System-wide objects (e.g., traders, directory, applications) which are not tied to a particular interpreter.

2.3 The topology dimension

This dimension is concerned with the layers of organizations of objects in system designs. Systems have a physical topology in terms of boxes and wires. Systems have a service topology in terms of the services available to 'applications programmers', including directories, filing services etc.

Physical layer

An object, such as a piece of hardware, in the physical topology of a system. Physical objects have an independent existance and the possibility of spontaneous decay. Such decay can be repaired by the substitution of an identical spare part. Physical objects can only enter or leave a system as a result of physical action.

Logical layer

Objects (such as binders and communications objects) which can be made to enter or leave a system as a result of internal system action (i.e., they are the manageable resources of the system). Logical objects depend upon a physical objects.

Service layer

Objects which provide services to directly support users and applications and are visible in the service topology of the system. Service objects depend upon logical objects.

2.4 The human-computer interface dimension

This dimension is concerned with the visibility of an object from the viewpoint of human users of systems. The coordinates here are based on an 'engineering' view of human-computer interfaces, in the absence of a widely accepted criteria for the separation of human interface concerns.

The coordinates are:

User layer

Objects that represent the roles played by human beings in the operation of a system.

Device layer

Objects that organize the presentation of information to human beings and the collection of stimuli from human beings. Examples of device objects include screens, keyboards, sirens and so on. Devices may be quite complex sub-systems in their own right, with facilities such as graphics refresh, ringing bells and so forth. All such behaviour which is self-contained in the object from the point of view of the user is ascribed to the device.

Presenter layer

Objects concerned with the mapping between dialogue and devices, including windows, menus, logical buttons, command line parsers and so on.

Dialogue layer

Dialogue objects mediate the flow of control and data between device objects and application interface objects - they impose a syntactic structure on the interaction.

Application interface layer

Application interface objects are the objects that correspond to the the images presented to the users of a system and include such things as

desktops, file drawers, timers and so on. Application interface objects define the top-level semantics of human-computer interaction.

Resource layer

Resource objects are system objects that do the work of application interface objects (e.g., file servers).

2.5 The evolution dimension

ANSA will change over time, in response to the identification of shortcomings and to changing requirements. This may require replacement of parts of the architecture. To keep such changes within the architectural framework an evolution dimension is introduced. The coordinates of this dimension are *ANSA Reference Manual* release numbers. An object is labelled with the numbers of the earliest and latest releases across which it is compatible.

2.6 The composition dimension

Composition recipes apply a structure to a set of composition level n objects and build a level n + 1 set of objects.

In an actual system design, the designer will have to decide how deep he wants compositions to go; this dimension is therefore relative in the architecture since the recipes go from n to n + 1. In the design for a system, absolute coordinates can be assigned to this dimension.

2.6 The temporal dimension

Temporal recipes apply a structure to a set of objects at time n and build a set of objects at time n + 1.

In an actual system design, the designer will have to decide how late into the life of a system he wants temporal compositions to be possible; this dimension is therefore relative in the architecture since the recipes go from n to n + 1. In the design for a system, absolute coordinates can be assigned to this dimension.

3 Relationship to the ANSA Reference Manual

This section attempts to set coordinates on the various chapter in the ANSA Reference Manual, Part III (note some strategic name changes):

C1 (Introduction)

Null

C2 (Architectural concepts)

The object model in general and the major architectural concepts

C3 (Distributed Processing Environment)

Explains the structure of C4-C7 as a statement of a generic processing, distribution and communications model (in C4) backed up by a description of a possible implementation using a combination of local services and infrastructures that exhibit the greatest number of implementation design choices.

C4 (Global level description)

<Level 7, Global object, HCI-any, Logical topology>.

C5 (Abstract machine description)

<Levels 2-3, Homogeneous, HCI-resource, Logical topology> (i.e. abstract machine and nucleus).

C6 (Local level description)

<Levels 4-6, Local object, Logical topology>.

C7 (Physical level description)

<Level 1, Heterogeneous, HCI-resource, Physical topology>.

C8 (Conformance)

Discusses other options for supporting the model in C4 e.g.

- all physical (i.e. a hardware implementation)
- directly by a high level interpreter (i.e. an ANSA veneer over SNA).

C9 (The human user)

A tour down the human user dimension <Level-7, Global, HCI-all, Topology-all>.

C10 (Transparency)

Recipes using <Level 7, Global, HCI-any, logical> objects to achieve various forms of transparency by indirection leading to and discussion of mechanizations via <Level $n \leq 6$, Local, HCI-any, logical> (viz., multicast set managers, concurrency control managers and so forth).

C11 (Naming)

A discussion of naming issues common to all transformation, topology and communications coordinates and descriptions of objects to support naming at various coordinates.

C12 (Security)

A discussion of security issues across all transformation, topology and communications coordinates and descriptions of objects at various coordinates to support security mechanisms.

C13 (Data)

A discussion of data access and management models appropriate to each transformation layer and objects at each layer to provide data access and management.

C14 (Management)

A discussion of system and community management models appropriate to each transformation, topology and communication layer and objects at various coordinates to provide management functions