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 ANSA Concepts

Abstract:

This document contains the Author's working definitions of the Key Concepts in the ANSA Meta-Architecture. These are distributed as a contribution to the establishment of definitions of these concepts on a project wide basis.

Number: AA.KC.00

Date: 27th March 1987 10:32 am

Area: AO (Architectural Overview)

Status: P (Discussion Paper)

Classification: U (Unrestricted)

Distribution: 4 (Part 4 contributors)

Author: HJW (John Winterbotham)

1 INTRODUCTION

This document contains the author's working definitions etc of the Key Concepts within the ANSA meta-architecture. These definitions are given in a 'top down' order as far as possible.

1.1 Conventions

The words used as symbols for referring to the various concepts are **emboldened** in use, and those that are considered as atomic (IE are defined externally to this document) are also ***italicised***.

1.2 Specification

A specification is a statement, expressed in some appropriate language, of the required characteristics, behaviour, structure etc of the thing specified (the target). As commonly used, a specification forms the basis of testing of the targets specified for conformance to the expressed requirements. Since there are some aspects of the targets that will be omitted from the specification, due to incompleteness or irrelevance, additional refinement and design are required before the target can be manufactured.

1.3 Design

Ths system designers will therefore take the specification, and will add details, make choices, and in otherways reduce the uncertainty of a specification. At some point the uncertainty will be reduced to a sufficient extent that the system can be manufactured. At this point a design exists.

However a carefull examination shows that some uncertainty always exists in designs, and that a design is actually a design specification and will also be used for testing conformance.

The division of this process into specification and design is therefore seen as somewhat arbitrary and problem dependent, and more related to the organisation of the work than to a fundamental change in the characteristics of the work itself.

1.4 Instantiation

Once a design has been created, instances of equipment must be manufactured that conform to that design and that actually exhibit the required behaviour. In the ANSA world, there does appear to be a fundamental difference between specification/design and instances of the design. Specifications anf designs say what should be built and what behaviour the instances should exhibit, without them selves exhibiting that behaviour.

1.5 The Role of an Architecture

Architectures are applicable to the structuring of both specifications and designs and to the specification of the componenbts and the relationships between them. The architecture will therefore have an effect on the various instantiations of systems conforming to that architecture. However it may well be that not all aspects of the architecture will actually be instantiated, resulting in the architecture of instantiation being a subset of the architecture of the specification or design.

2 ARCHITECTURE

2.1 Definition

An **architecture** consists of -

- a **framework** that provides a basis for reasoning about the structure of a system
- a ***set*** of **templates** from which to construct **components**
- a ***set*** of **components** or system building blocks that can be composed and extended to provide complete system designs
- a ***set*** of **relationship** definitions
- a ***set*** of **relationships**
- 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 **components** and **recipes** can be applied to meet real needs, and why the **rules** are reasonable constraints on designs.

2.2 Discussion

ANSA is an architecture project, and it is therefore vital to define, and ultimately to specify, what we mean by 'architectures', and how we represent them.

The purpose of architectures is to provide structure for the design of systems, and commonality between the various parts of both designs and implementations.

The common structure has been found to assist designers to separate important concerns from each other, and different parts of the system from each other. Both separations reduce the complexity of the designer's task, and allow several designers to work on different parts of the whole at the same time, while constraining their interdependancies.

The structure is provided by dividing the system into **components**, by relating the **components** to a common **framework**, and then by combining the **components** into **models** by establishing **relationships** between them.

Commonality of design or implementation reduces the effort required to produce a system, by reusing the same part in several places in one design or implementation, or indeed in several designs or implementations. This is a

very common technique in engineering which provides considerable benefits but at the expense of reducing the freedom available to designers.

Commonality is provided and encouraged within the architecture by providing **templates** from which **components** may be derived, and for sharing the definitions of common parts of these **templates** by the mechanism of **inheritance**. Further commonality is provided by requiring conformance to **rules** that limit the freedom of the designers to construct arbitrary designs from the **components** , and by advising the designers via **guidelines** as to common or advantageous techniques or parts of designs.

2.3 Specification

To be added

3 FRAMEWORK

3.1 Definition

The **framework** of an **architecture** consists of a ***set*** of **dimensions**, each of which has a ***set*** of **coordinates** defined along it. Each **coordinate** corresponds to one possible value of a **property** of a **component** of the system. Each **component** may be placed relative to the **framework** at the position at which each **coordinate** corresponds to the value of the **component**'s corresponding **property**.

3.2 Discussion

The **framework** within an **architecture** is the main way of providing and describing the structure of systems conforming to the **architecture**. Since the main reason for providing such structure is to make the systems easier to comprehend and to design, the structure needs to be based upon the main concerns that designers, customers, users, etc actually have. The **framework** is therefore based on the variation of key characteristics of the **components** comprising the system, where each characteristic is determined from the value of an appropriate **property** of each **component**. The purpose of the structure is the separation of **components** with particular characteristics from other **components** with different characteristics. Such separation implies a representation that is capable of showing the different groups of **components**, and their commonality or lack of it.

Given the wide range of people concerned with comprehending the **architecture**, a 'spatial paradigm' is used to represent the structure. This involves the definition of a set of **dimensions**, each concerned with a particular aspect of system **components** that is of interest, and each having **coordinates** along it that correspond to all possible values of the **property** that is represented along the **dimension**.

For the spatial paradigm to be valid, the **dimensions** must be orthogonal, that is variation in the **property** associated with one **dimension** must not *a-priori* cause variation in another **property**. This must not be cofused with the normal correlation between values of the **properties** of different **components** which is a consequence of design requirements rather than links between **properties**.

3.3 Specification

To be added

4 COMPONENT

4.1 Definition

See **Object**

4.2 Discussion

Component is used as a general term for an identifiable part of a system described by the architecture. In ANSA we use an 'object paradigm' for describing the separate parts of the architecture, and so for the purposes of this document, a **component** is synonymous with an ANSA **object**.

4.3 Specification

See **Object**

5 TEMPLATE

5.1 Definition

To be added

5.2 Discussion

Template is a general term that is used to refer to that part of a design specification that contains or provides all of the information needed for instantiating a single **component**.

In many cases **template** may be considered to be synonymous with **type**, although at this point we do not have an adequately complete definition of **type**.

5.3 Specification

To be added

X RULE

x.1 Definition

To be added

x.2 Discussion

To be added

x.3 Specification

To be added

OBJECT

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.

2.1 Definition

To be added

2.2 Discussion

To be added

2.3 Specification

To be added

X CHAPTER HEADING

x.1 Definition

To be added

x.2 Discussion

To be added

x.3 Specification

To be added

1.1.1 B (sub-section) heading

architecture
component
object
boundary
operation
interface
relationship
property
model
frame
framework
dimension
coordinate
layer
level
type
rule
recipe
inheritance
specification
definition

language
name
set
list
map
mapping
tuple
predicate
event
message
atomic
guideline
abstract machine
infrastructure
interpreter
behaviour
binding
system
contact
medium