Advanced Networked Systems Architecture

24 Hills Road
CAMBRIDGE
United Kingdom CB2 1JP

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

# On the Dimensionality of Architectures

| | | |
|---|---|---|
| **Number:** | AO.14.01 | |
| **Date:** | 2nd September 1986 12:24 pm | |
| **Area:** | AO | (Architectural Overview) |
| **Status:** | P | (Discussion Paper) |
| **Classification:** | C | **(COMMERCIAL-IN-CONFIDENCE)** |
| **Distribution:** | T | (ANSA Team) |
| **Author:** | HJW | (John Winterbotham) |

## 1 Background

Many of the systems with which we are concerned within ANSA fall into the general categories of 'distributed', 'decentralised', 'integrated', 'multi-media', and similar terms. The designers of these systems are faced with a bewildering array of concepts, design techniques and enormous range of required knowledge, a wide range of customer requirements, and rapidly reducing timescales. These problems are driving a search for methods and techniques for controlling the design process, scoping the problem domain, and clarifying the major technical choices. These techniques are broadly grouped and thought of as being concerned with the design and analysis of 'Architectures'.

Systems designed according to these architectures must be presented to a wide range of concerned groups in ways that are meaningful to those groups, so that they may make soundly based decisions about them. Techniques are therefore required for deriving these different descriptions from a single description of the system, and of showing their equivalence and the relationships between the representations.

In addition to the need for the system architects to have techniques for doing their job, they must also have techniques and representations for transferring their architectures and the designs of conforming system to the implementation teams, who will require clear, unambiguous, and verifiable specifications from which to work.

## 2 What is an Architecture?

To understand the problem further, we must establish a reasonable definition of what we mean by the term 'architecture'. Within the ANSA activities we take an architecture to be a framework within which system designs may be placed and whose structure they adopt, a set of generic components that may (or must) be used in the systems, a set of generic relationships that may (or must) be used between these components, and some rules about all of these that guide and constrain their use, as well as the addition of new parts of the framework or new components.

Before the framework can be established, the components described, or the rules formulated, a coordinate system must be put in place, and this in turn requires a basic knowledge about the scope and properties of the problem domain.

## 3 The need for dimensions?

There is no a-priori reason for the coordinate system that is used to describe architectures to be based on mutually independent properties, and indeed it is probably impossible to do so given the present state of knowledge in this subject. However there are considerable benefits to be obtained from reducing the interdependencies between the properties associated with the coordinates, from equating each coordinate with variation in a single significant property of the problem domain, and from identifying a sequence of values for each coordinate that correspond to more or less significant changes in the values of these properties. Since the same location and manipulation effects can be obtained from a variety of coordinate systems (cf cartesian and polar co-ordinates in geometry), we need to choose a particular set of co-ordinates, and once chosen we refer to them as the **Dimensions** of the **Problem Space.**

## 4 Selecting Dimensions

The selection of properties and their associated coordinates as Dimensions is difficult and the subject of a separate paper [1], but some objectives for the selection can be given here.

The total number of Dimensions should be minimised, to as to reduce the complexity of the architectural description. Each Dimension should be mapped onto a major property of the problem space that has significant variation over the systems of concern to the architecture. The property that varies should have significance to the world outside the system and its designers (EG to the Users, Buyers, Salesforce, PTTs, etc) in order to ensure easier expression of the design in their terms. In addition to all of these, the choice of Dimensions should help the separation of major problems from each other as far as this can be judged. (For example in a distributed system the choice of communications layering - OSI possibly - as one Dimension reduces the complexity of distributed system design by ensuring that the topology of Service provision, another possible Dimension, need not address the complexities of the communications protocol layering).

## 5 Can we Calibrate these Dimensions?

The establishment of Dimensions is of little use if fixed points cannot be established along each of them. This is an essential prerequisite for the location of components within the architectural framework, and thus to the design of architectures and systems.

The four dimensions that we meet in everyday life are arbitrarily divided into equal increments (and regular sub-increments) but such regularity is not required for architectural dimensions, and is not always possible as there will often be discontinuities or boundaries within the range of the property value mapped onto the Dimensions. What is required is a series of fixed reference points whose interrelationships are known and understood, and such interrelationships do not need to be continuous.

Where no continuous variation in property exists (EG where categories are used), the mapping of the Dimensions onto physical representation may, or may not, choose to employ a linear equal increment representation (cf Euclidian). This is a representational convenience rather than a fundamental characteristic of the dimensions.

## 6 Can we derive an 'algebra' for manipulating the Architecture?

Once we have established a coordinate system and a set of Dimensions, the next requirement is for an **algebra**. In this context, an algebra is a collection of rules, theorems, assumptions, representations, and prior results describing the objects within the problem space, and bounding and guiding their manipulation.

The Object paradigm appears to be very suitable for the descriptions of architectures (with possible modifications to the strict software object paradigm), and if we follow Lorin's work [2] on extending the object paradigm to include attributes and relations (which are the type definitions for the properties of and interactions between object instances), then we have the means for manipulating the components of our architectures, assigning and constraining property values (including location within the problem space), and describing the interactions between the components.

We can go further than this, and can adopt the DASE work on interfaces by defining an interface description as being a type of relation, which is required to be between objects having particular combinations of attributes and relations with other objects.

Consideration of the possible discontinuous nature of the Dimensions, of the need to express logical combinations of attribute, relation and object type, leads to the conclusion that the appropriate algebra for describing architectures is **not** based on the algebra of real or integer numbers, but on the same foundation as Formal Specification techniques - discrete mathematics and the Predicate Calculus.

Similar consideration of interactions and the requirement for chains of interactions and how they terminate indicates the applicability of Path Algebras.

## 7 Models

The construction of **models** is a fundamental part of the work of ANSA and also of DASE. Models consist of object instances and the interactions between them (including interfaces), and they provide the mechanism for representing systems to people in such a way that they can reason about their behaviour, and use the conclusions drawn to modify their actions (If I do this here then that will happen there, so I will do the other instead as something else will happen...). Models are usually simplified representations of the total complexity of the system, and are intended for use by particular groups of people with particular sorts of task. For example a physically oriented model of a system as being composed of computers in boxes connected by cables is suitable for those installing or maintaining it, but is incidental to those using it for wordprocessing etc. It should be noted that such models often hide internal details of the objects represented (Services, Software, management .....) in order to simplify and present only what is of relevance to those using the model.

From this is can be seen that models are 'projections' of the actual system design onto some 'slice' through the problem space, and also that the projection of an object onto that slice hides the fact that it may be composed of other objects, linked by interactions, that are distinguished from each other by properties that have been ignored by the act of projection. For example a wordprocessor may be modelled as a single object with appropriate behaviour, thus hiding the differences between hardware, software, operating system or application because they are not of relevance to the simplified problem space within which the typist works.

From this it can be seen that some objects may occupy regions in the full problem space and so may include other objects (a wordprocessor will include both a processor object and a software package object), but which may not intersect other object boundaries (the same software instance cannot exist in two processors at once, although copies can).

## 8 Slices

As identified above, slices through the problem space can be a valuable way of simplifying the presentation of the architecture (cf 'cuts' in OSI 7 layer model [3] ). In particular slices provide the basis for formulating a description of the architecture for particular audiences. They must therefore be chosen carefully so as to be meaningful to the target audience, who will be mainly concerned with a particular aspect.

The choice is made much easier if one or more of the dimensions of the problem space are constant for the slice, and so by working backwards from the aspects of the system that are meaningful to particular audiences, and establishing the slices, the main Dimensions of the problem space may be determined.

## 9 Pictures

Slices will ultimately by presented to the target audiences in the form of diagrams or pictures. These will, given currently practical technologies, be 2 dimensional representations on paper or VDU screen. While it is possible to produce good isometric representations of 3 dimensions, the added complexity is not desirable.

Pictures or representations of the architecture should therefore employ no more than 2 dimensions, which in turn implies that slices should have no more than 2 dimensions which vary, while the rest remain constant. When this is the case, they may be termed 'planes' through the problem space.

## 10 ANSA Views

The current ANSA work is based upon the Multi-dimensional view of architectures, and the representation of the architecture to a wide variety of people who have different perspectives on the system. The more basic question of the required number of dimensions and their associated coordinate system is discussed in [1].

## 11 References

[1] *The ANSA Climbing Frame* AO.16

[2] "An expanded Approach to Object" Harold Lorin. - Operating System Review, Jan 1986

[3] "Structuring Principles of the Communication Architecture of open Systems - A Systematic Approach". H J Burkhardt and Sschindler Computer Networks 5 (1981) p157 - 166