ANSA Reference Manual

Release 01.00

Part III: System Modelling

Part III

Chapter 1

Introduction

1.1 About Part III

This part forms a detailed glossary of generic architectural concepts. It is intended to be used as a dictionary, to clarify concepts as they are encountered in the rest of the manual. As with any dictionary, the terms and their definitions are subject to change in response to experience of its use in modelling and design. These changes will be motivated by the discovery of new concepts, through the consolidation of terms, and by the replacement of awkward or unfamiliar names of concepts.

The description of generic concepts is object oriented and is based on a carefully selected graphical and mathematical notation. Arrangements of objects are shown as graphs and objects are described in set theoretic terms. A mathematical approach helps to achieve consistency between definitions. The notation has been deliberately kept simple and is introduced within Part III. The concepts have been organized in a sequence so that early definitions are developed to provide the later entries. Part III can be read from beginning to end as a tutorial introduction to architectural concepts.

1.2 Generic concepts

Some concepts that are employed in describing distributed systems appear to be generic. Some concepts are immediately applicable to a wide range of situations, and some concepts can be presented as simple refinements of a generic idea. Identification of the generic aspects of descriptions of distributed systems reveals the potential for eliminating duplication of effort in design and standardization. A common conceptual basis for the various steps in a design also simplifies the integration of tools that help designers of systems.

Use of a common vocabulary and notation enable the different members of a design team to communicate and to reason in a well-defined fashion. This helps to assure team members that they are dealing with the same problem and are contributing to the same overall design. Similarly, the experience of one design team can be of benefit to another because communication between teams becomes feasible through the common vocabulary and notation [PARNAS 85].

Objects are the pieces out of which a structural model of a system is built and are chosen so as to represent pieces of a system that are of some significance to an observer. Within a model, an object is elementary in that it is described without reference to any internal structure or mechanism.

Most of the concepts used in describing distributed systems can easily be associated with an object and generic concepts are associated with an object type.

Sometimes a concept is better described with reference to an arrangement of objects or even possible relationships between arrangements of objects, but the emphasis in Part III is on treating concepts as though they identified types of object.

There are four principal types of object. Firstly there are objects that seem useful to the application designer. Examples of such objects are configurations, hierarchies and federations, interpreters and models, associations and bindings, names, resource managers, interface adapters and traders, groups, transparencies, and so forth. Most of these immediately useful objects are described in Chapters 7 to 13, 15, 16, and 17.

Secondly there are objects that when used as part of an arrangement have characteristics that allow the arrangement to be transformed and thus provide the ways in which the description of a system can be transformed to allow simplification, or comparison. Because the focus of this Part is on objects, the transformations that can be carried out on system descriptions are scattered through the chapters. Chapter 6 on Relationships does include a large number of "mathematical objects" and includes the corresponding transformations. Chapter 18 has been included to serve as a repository for the most commonly used transformations of object diagrams.

The third kind of object is introduced to simplify the description of the relationships between other objects. These objects are described in the early chapters (3 to 6 and 14), as are the last kind of objects: the basic concepts, described in chapter 2.

1.3 Criteria of choice

There is a choice in which concepts form the foundation of the architecture. There are some broad criteria that can be stated:

- The number of concepts necessary for the description of systems should be minimized[‡].
- Concepts should be chosen so that they are applicable to many stages in a design process.

- Concepts that are applicable at different times and in different places allow the postponement of decisions about temporal and spatial distribution. The ability to abstract from space and time in the early stages of design allows the development of generic designs and proofs which help to identify commonality between what one might previously have considered to be quite distinct.
- Some aspects of design are independent of the type of objects that are involved. It is their arrangement that is important. Generic descriptions that focus on structure can help to detect common characteristics of a range of systems.

1.4 Reading Part III

When reading the manual for the first time, it is recommended that you read the chapters in the order in which they appear. In that way the material can be used as a tutorial in ANSA architectural concepts. It is anticipated that readers will frequently refer to specifications in Part III when studying or working with with the specializations of the generic concepts elsewhere. The cross reference tables in chapter 20 may be an aid to finding particular specifications. To understand a particular concept it is often necessary to understand several related concepts as well. The grouping of related generic concepts in one chapter should reduce the effort involved in cross referencing.

1.5 Topics to be covered

This part aims to give a complete description of the generic concepts. There are a number of aspects to a description that together ensure the completeness of descriptions.

Once a generic concept has been identified, it must be named, so that it can be referred to. The names adopted in this part are based on common usage. However, because a generic concept may be used in different projections, the way in which names have been used is rarely consistent. Sometimes one name is used for two or more different generic concepts. At other times the same generic concept has several names. In this Part a generic concept is given only one name.

An object will be used in conjunction with other objects. There are sometimes restrictions on the types of objects that can be used together or restrictions on the ways in which they can be arranged. These arrangements

---

[†] This criterion seems to be contradicted by the size of Part III. However, similar concepts are given different names by different specialists, which has led to a multiplicity of entries, and the current version represents a snapshot of the work on architectural concepts and the entries have not been fully consolidated.

can be generalized to give a canonical form that guides the identification and use of a generic concept.

Emphasis is given to explaining constraints on the necessary structure and behaviour of a distributed system, while permitting the designer free choice of how the components are to be individually defined.

In this part, systems are represented by a graph of **objects** with **connections** between them. Objects are named to identify the **constraints** that they impose upon other objects. The constraints are expressed algebraically. The choice of an algebraic form for constraints helps designers to transform arrangements of interconnected objects into alternative combinations of objects and connections. Such transformations are necessary when exploring design options and when explaining the dynamic behaviour of a system.

Formalism is achieved by showing arrangements of components as graphs and by describing the components themselves in set theoretic terms. The link between set and graph theories is very well documented in standard works on mathematics and physics, see for instance [CARRE 79, NANZETTA 71].

While the study of information systems is primarily an intellectual exercise, such systems are ultimately built from physical components. Information systems are therefore subject to the constraints and uncertainties of physical devices. A model of a distributed computer system must take into account the constraints of the physical world. In other words, the designer must at some stage demonstrate that the design is realizable.

Some of the concepts are related to one another and the text notes such links.

Examples often help to explain a concept and its use. Each chapter of Part III ncludes a section for examples of the concepts introduced in that chapter.

1.6 Layout of descriptions

Each of the later chapters in Part III specifies a set of related generic concepts. Each chapter is briefly introduced and a set of examples of use of the concepts usually appears at the end of the chapter. A reference section is placed between introduction and examples. The reference section contains he description of the related generic concepts. It has been organized like a dictionary with one entry for each generic concept. Each entry is set out in sections as follows:

NAME
the name of the generic concept or arrangement of concepts

PURPOSE
a short statement of the purpose of the generic concept.

SYNOPSIS
a short informal description of the generic concept

CANONICAL FORM
an illustration, often in terms of an object diagram, that shows how an object that represents the concept would be arranged in relation to other objects.

SPECIFICATION
a mathematical statement together with its interpretation in English that describes the constraints that the object imposes on its interactions with other objects.

REALIZABILITY ISSUES
special considerations and constraints that must be met if the generic concept or arrangement of concepts is to be physically realized.

SEE ALSO
references to other related generic concepts or arrangements of concepts which might help the reader's understanding of the description

FUTURE DIRECTIONS
known problems in the description or use of the generic concept or arrangement of concepts and possible directions for the solution of these problems

1.7 Notations

The notation adopted for the models of distributed computer systems is closely aligned with the mathematical foundation upon which the modelling is based: graph theory and set theory. The graphical and sentential notation is defined in the entries in the rest of Part III as it is required. The tables below are for reference and contain some of the sentential notation with their common mathematical "meaning". The source for most of the mathematical notation that was adopted is [HAYES 87] and [LIPSHUTS 64]. The notation for logic is summarized in table 1.1; in the table: A, B and C are predicates. Set notation is shown in table 1.2; in the table: X and Y are sets; x and x_k are terms; B is a predicate.

The algebraic formalism enables comprehensive and rigorous descriptions to be given. There is also a graphical syntax for showing how objects are related. A graphical syntax has been adopted to emphasize the structure of arrangements of objects and because at times the detail of a full algebraic description can be a distraction to the arguments being presented. The diagrammatic notation enables sketches of systems to be described, as well as complete designs, without stepping outside the formal framework.

Table 1.1 Logic

Notation	**Name**	**Reads as**
true, false	Logical constants	true, false
$\neg A$	Negation	not A
$A \wedge B$	Conjunction	A and B
$A \vee B$	Disjunction	A or B
$A \Rightarrow B$	Implication	if A then B
$\forall a . B$	Universal quantification	for all a, B is true
$\exists a . B$	Existential quantification	there exists an a such that B is true
$t_1 = t_2$	Equality between terms	t_1 equals t_2

Table 1.2 Sets

Notation	**Name**	**Reads as**
$\{x_1, x_2, x_3, \ldots, x_n\}$	Set definition	the set containing $x_1, x_2, x_3, \ldots x_n$
$\{x \mid Bx\}$		the set containing exactly those x for which Bx holds
$\{\}$		the empty set
$x \in X$	Set membership	x is an element of X
$X \subseteq Y$	Set inclusion	X is a subset of Y; X and Y may contain exactly the same members
$X \subset Y$	Strict set inclusion	X is a subset of Y; X always contains fewer members than Y
$X \cap Y$	Set intersection	the intersection of X and Y
$X \cup Y$	Set union	the union of X and Y
$X - Y$	Set difference	difference between X and Y
$\#X$	Cardinality	the cardinality of X

.8 References

[CARRE 79] Carré, B., *Graphs and Networks*, Clarendon Press, Oxford, 1979

[HAYES 87] Hayes, I. (Ed), *Specification Case Studies*, Prentice-Hall International (UK) Ltd, 1987

[LIPSHUTS 64] Lipshuts, *Set Theory and Related Topics*, Schaum Outline Series, Schaum Publishing, New York, 1964

[NANZETTA 71] Nanzetta, P. and Strecher, G.E., *Set Theory and Topology*, Boyden and Quigley Inc., New York, 1971

[PARNAS 85] Parnas, D.L., *Software Aspects of Strategic Defense Systems*, *American Scientist* (November 1985)