ANSA Reference Manual

Release 0 1.00

Part II: Foundations

Part II
Chapter 1

Distributed processing

Distributed processing is an emerging discipline for organizing complex computer systems. The development of the discipline is essential for the exploitation of both the current technologies of computer networks (especially local area networks), workstations, personal computers and the emerging technologies of wideband communications and advanced processor architectures. Distributed processing has many application domains, including many kinds of information processing systems, control systems and multi-media communication systems.

A simple networking system provides communications between otherwise autonomous computer systems to enable remote access to independent information and processing resources. Distributed processing combines such resources into an integrated structure to support the needs of a particular application, such as the automation of information processing in an office or factory.

Networking is an important part of distributed processing, but communications alone are not sufficient to ensure the required degree of compatibility between the components of a distributed system. Constraints on the structure, behaviour and interfacing of the networked components themselves are also necessary if they are to be able to support distributed processing.

1.1 The need for distributed processing

Human organizations are, by their nature, distributed: people work in different places and information is acquired and stored at different locations. People increasingly rely on computers in their work and require good response and availability from the computers they use. It is often convenient, and sometimes necessary, for reasons of performance, accessibility or security, to locate computers close to the information they process. Sometimes privacy is important. These requirements can be met by provision of personal computing power. In turn this places a requirement upon computer systems which come to reflect the distributed nature of the human organizations they serve.

As organizations add increasingly complex functions to their systems, suppliers will find it more difficult to manufacture and support a full range of system components and will be unable to offer a complete solution to all needs. However, because the information technology market is growing, suppliers are able to focus their product range, and maintain sales by concentrating on selected application domains. In addition specialized and start-up suppliers are selling innovative system components.into niche markets. With distributed processing, purchasers can take advantage of this market diversity and combine their own mix of components into a distributed system that serves precisely their needs.

Human organizations are subject to change. Change must be accommodated by the computers in an organization. Technology is providing opportunities to develop computers with new functions which justify new applications. As technology changes, organizations will wish to purchase new equipment to exploit the new developments. It is unlikely that they will be able to justify or manage the replacement of old equipment. Organizations will expect gradual and continual evolution of their systems. As a consequence, systems will be thought of as a collection of components rather than a single resource.

Large organizations engage in many diverse activities, each of which can be supported by a computer system. For example, a manufacturing company may have computer systems to support design, production, warehousing and accounting. To make the most efficient use of these individual systems it is necessary to provide links between them and provide an integrated information and control system for the company. Integrating multiple computers is relatively easy when they are all of the same type. They can be linked in such a way that, to the application designer, they appear to be one very powerful machine. This approach does not always lead to a satisfactory solution, however, because different applications have different operational requirements and will emphasize different quality attributes such as response time, throughput, security and reliability. For example, factory automation applications will require real-time response and a high degree of reliability whereas applications in the accounts department will place fewer constraints on real-time response but will probably require security and backup. This diversity of application requirements demands a diversity of hardware and software support. Distributed processing is the way of achieving this integration, but it is desirable that the standardization required for integration does not restrict the freedom to choose the most appropriate technology for each activity.

1.2 Viewpoints on distributed processing

To derive the Advanced Networked Systems Architecture, the ANSA team studied current practice and research in distributed computing and system design techniques.

The study revealed that different distributed processing experts have different viewpoints about what are the crucial concerns that make up

'distributed processing'. Further examination revealed that five viewpoints were dominant and that each viewpoint in some way or other acknowledged the concerns addressed in other viewpoints but with a lesser priority. As a consequence, ANSA has been structured as a set of projections of the architecture onto these five viewpoints. The projections that make up ANSA are termed the enterprise projection, the information projection, the computational projection, the engineering projection and the technology projection. A distributed system can be described in any one of these projections of ANSA, and the resulting descriptions, or **models,** reveal different facets of the system. Each model is self-contained and complete. The difference between models is not how much of the system they describe, but rather to what aspects of the system they emphasize.

Designers used to working with one projection often have difficulty assessing the relevance of the concepts used in any of the other viewpoints. It is important to realize however that a system has ultimately to be described from each viewpoint; all of the viewpoints are equally valid and it is a mistake to argue which is the more fundamental.

1.2.1 Enterprise projection

The purpose of the enterprise projection of ANSA is to explain and justify the role of a computer system within an organization. An enterprise model describes the overall objectives of a system in terms of roles (for people), actions, goals and policies. It specifies the activities that take place within the organization using the system, the roles that people play in the organization, and the interactions between the organization, the system and the environment in which system and organization are placed.

Enterprise models provide managers of the enterprise with a description showing how and where the system is placed within the enterprise. The interactions between the organization and the designers of the system can be included in this projection so that the process of procurement, installation, maintenance and evolution is included as part of the system design.

Design decisions made using the enterprise projection of ANSA concern *what* a system is to do and *who* it is doing it for. The design concepts defined in the enterprise projection allow the designer to develop a closed (i.e. bounded) model which represents all the real world requirements which the designer is prepared to incorporate into the structure of the system.

1.2.2 Information projection

The purpose of the information projection of ANSA is the identification and location of information, and the description of information processing activities. An information model describes the structure, flow, interpretation, value, timeliness and consistency of information held within a system. An important feature of distributed systems is that information can be collected, processed and presented in different places and at different

times. Consequently, information models consider the system as a collection of information resources and information processes. The asset value attached to information in modern business makes these models importani to managers as well as designers.

Designers use information models to explore the nature and role of information in the system. They take design decisions that are epistemological in nature: "Who knows what?", and 'Where can information flow?". The people in the organization may also figure in the information model so that the impact of a system on the organization that is to use it can be properly analyzed.

1.2.3 Computational projection

The purpose of the computational projection of ANSA is to show the organization of a distributed system as a set of linked applications programs A computational model provides programmers with a description of a system that explains how distributed application programs may be written for it This description is in terms of data representations, programming languages, system services and program specifications. Computational models show how programmers have structured systems for modularity and parallelism, for linking separate applications into integrated packages and for making programs independent of the computers and networks on which they run.

The computational projection of ANSA necessarily takes account of systems in which computations can proceed in parallel, can fail independently and which are configured dynamically.

1.2.4 Engineering projection

The purpose of the engineering projection of ANSA is to describe distributed systems in such a way that designers can reason about the performance of the systems built to their designs. The engineering projection describes processing, memory and communications functions that can be used to support the requirements of programs structured according to the computational projection.

A designer uses an engineering model to make decisions that concern trade. offs between quality attributes such as performance, dependability and scaling. Designers, system management and maintenance staff all use these sorts of models as a basis upon which to predict the consequences of possible system reconfigurations.

There is no discretion in the engineering projection over the sorts of information processing components the system is to contain: design is restricted to the sorts of coupling between components and the management of computer and network resources to achieve desired quality attributes.

1.2.5 Technology projection

The purpose of the technology projection of ANSA is to describe the physical components, in terms of hardware and software, that make up the distributed system so that they may be related to the architectural concepts defined in the other projections. Technology models act as blueprints of systems during their construction and maintenance, showing how architectural concepts have been implemented in practice.

1.3 Scope of ANSA

To understand the scope of ANSA, the ANSA team studied the various stages that are involved in the establishment of an architecture for distributed processing and proposed the progression outlined below:

- **definition** of concepts to develop the terminology and the basic understanding of the issues involved
- formalization of the concepts and development of models, to permit rigorous reasoning about systems
- specification of standard design templates and rules that describe how these templates may be combined
- verification of the adequacy and suitability of the templates by prototyping
- provision of tools to automate the implementation of templates for particular systems
- maintenance of the architecture to support its continued validity and to allow for evolution
- exploitation of distributed systems built using the architecture in new products
- investment in education to disseminate the architecture to distributed systems designers and implementors.

The set of ANSA projections on the one hand, and the stages described above on the other, provide a 'map' of the field of distributed processing (see Figure **1.1).**

The areas of distributed processing currently included within ANSA are shown in the figure by '•'.

To date, the ANSA team have concentrated on the computational and engineering projections, since they have a large degree of independence from both application domains and technology trends. These projections provide an environment for the creation of stable specifications for the interface between distributed applications and the hardware and software that support them.

Figure 1.1: Scope of distributed processing

	Definition of concepts	Formalization of concepts	Specification of templates	Verification	Provision of tools
Enterprise	•				
Information	•				
Computational	•	•	•		
Engineering	•	•	•	•	•
Technological	•				

Part II
Chapter 2

Architectures

First the taking in of scattered particulars under one Idea, so that everyone understands what is being talked about . . . Second, the separation of the Idea into parts, by dividing it at the joints, as nature directs, not breaking any limb in half as a bad carver might.

Plato, *Phaedrus,* 265D

The purpose of an architecture is to systematize the practical experience of designing complex systems into a secure design discipline. Some architectures may deal with social and political systems, others may deal with technical systems such as buildings, or physical systems such as mechanical devices or computers.

The creators of an architecture need to analyze possible, proposed and existing solutions to design problems. This analysis needs to identify and describe the generic features found in different solutions and to find common ways of structuring these features. Once identified and described, the features and structures can be named in such a way as to assist dialogue amongst designers and reduce the burden of design work. The distributed systems architect's job is to analyze, summarize and systematize knowledge and experience about the features of distributed systems and their organization.

The analysis is not straightforward. It requires insight and meticulous corroboration. The architect needs to set boundaries between the variety of available solutions in order to identify common features. This partitioning is a judgment and there are no unique answers. There can be many architectures for a particular kind of problem. The choice of a particular architecture involves consideration of aesthetics and approach: in choosing a style for the architecture simplicity of expression, the practicability of solutions and current issues, such as performance, all act as guiding principles. The adoption of a style imposes constraints on how systems may be designed and built. It is important that rules about style do not hamper the designer or prevent desirable features from appearing in designs.

2.1 ANSA and the system design process

In designing an architecture care should be taken to choose an approach that can represent both problem and solution in the same terms. Thus the elaboration of the problem in terms of the architecture becomes an expression of the solution [ALEXANDER 64]. The designer's apprenticeship is made less burdensome when the problem description, the systems model and the implementation blueprint all have a similar style, The goal for the architect is to find a manageable collection of constructs that can, in combination, be used to explain clearly and simply how systems are organized to meet their functional requirements,

System development is often rationalized by treating it as the repeated application of a single canonical step, where each step has the same general form. Each step takes the currently latest description of the system and produces the next description by means of a transformation. The word transformation is used here in the widest sense: it may be entirely automated, as when performed by a compiler, semi-automated, or entirely manual. Each step is constrained to be sufficiently small that it will succeed with some confidence. Each description is expressed in some symbolic form - that is a linguistic system with notation and defined concepts - which is appropriate to its position in the sequence of steps.

Transformations, particularly those performed manually, are often subject to checks. Two kinds of incremental check on the target can be distinguished, namely verification and validation.

Verification is the process of showing that the result of a transformation is internally consistent and displays all the properties required by the description from which it was derived.

The fact that a target has been verified does not guarantee that it provides a good basis for further development, At any step, a large number of descriptions could have been produced, all equally correct, but showing different trade-offs and design judgements. Some of these descriptions will be entirely satisfactory; others will have brought undesirable properties into the design. Thus, in addition to verification, there is a need to check that the target is fit for its purpose. Such a check is termed validation. It requires reference to the requirements of the system owners and the enterprise into which the system is to be placed.

Bringing these ideas together, it becomes clear that an architecture has to provide two different things:

(1) a set of languages for describing systems, and

(2) sets of concepts, design rules, design principles and guidelines (i.e. a design schema) for verifiable transformations of descriptions in one language to descriptions in another.

Concepts specify the basic elements from which a model can be constructed.

Rules state how the concepts may be combined: they constrain the freedom a designer has to make design choices. A system which violates an architectural rule may operate as required but could be more difficult to enhance or modify than one which did not, or it may not work at all.

Design principles suggest useful compositions of concepts to meet particular needs. Principles aid the design of structures that achieve particular effects. They are compiled so as to provide effective ways to do things while remaining within the rules. It is not mandatory to follow the principles but the use of principles reduces design effort. It is likely that components adhering to design principles will become the most widely available, since following the principles will be a quick way to design a system.

Guidelines explain the problems that can be encountered in design and how they can be avoided; guidelines give general techniques for constructing models of systems rather than the more explicit instructions that are contained in the recipes.

2.2 Design techniques

Techniques for the design of many kinds of social and physical systems have common features. These features have been noted by many distinguished authors [LE CORBUSIER 54, SIMON 62, ALEXANDER 64, CHECKLAND 81, LISKOV 861. The intention of all design techniques has been to assist the system designers and builders by making designs and descriptions manageable. The essence of the various approaches is to reduce complexity and to identify and classify common patterns.

Designers of all kinds choose to ignore some of the details in order to cope with complexity and this deliberate omission of detail is called **abstraction.** The detail that is considered establishes a point of view or projection of the system and several different projections may be appropriate at different design stages. The projections that comprise ANSA were introduced in Part II, Chapter 1.

The following sections explore some ways in which designers use abstraction.

2.2.1 Classification and types

Classification involves looking at systems as compositions of parts and ignoring detail so that several components can be characterized by a single description. A set of components characterized by a single description is called a class. Many classification schemes are hierarchical by nature with components grouped into classes which are themselves grouped in classes and so on. Some schemes permit a component to be assigned to more than one class. Some schemes insist there be a single universal class to which all components belong. These are all matters of organizational convenience for the classifier. The term **type** is used for classes whose descriptions are

constrained to avoid circular definitions. A particular component which is an element of a type is known as an instance of that type and the description that is common to all the potential instances of a type is known as the type specification.

A component can be described by reference to its type specification, together with an outline of the additional features that make that component unique, thereby reducing the size of the specification for each component. The use of classification stimulates sharing of specifications which can reduce effort in design and manufacture. The use of classification also helps to organize the description of components; types are frequently used to check the consistency between designs and their implementation.

In computer science, classification has become an important tool in areas such as formal specification, database organization, programming languages and operating systems, under the umbrella label of 'object-orientation'. The application of the same label to different spacializations can cause problems when people compare what is meant by terms such as 'object'. In ANSA these differences are made clear by relating objects to the projection in which they are defined. The notations used in ANSA for describing concepts are neutral to the various interpretations of 'object'. This enables the relationship between objects in different projections to be explained.

2.2.2 Decomposition and epochs

Even though in abstraction much of the detail is ignored, the remaining description of a problem or solution may still be overwhelming. A rational way of tackling complex problems or expressing complex solutions is to think of the problem or solution as a composition of several parts. The nature of each of the parts can be discussed individually, repeating the decomposition process if necessary, and then other, separate, discussions can focus on how the components fit together.

This raises an important point for these designer of a system because some of the structure in a design may be necessary to satisfy the requirements upon the system, whereas other structure may be merely an artifact of the way in which the designer has chosen to think about the problem. Examples of necessary structure may be representations of geographical location - "the system is to have three computers, one each in London, Paris and Munich" - or of security - "there is no communication path between trusted and untrusted processes". The latter example also indicates that in some cases requirements are phrased in terms of what is, or should not be, possible.

Some components may appear in several projections of a system and be decomposed quite differently in each projection to explore different aspects of the components. The designer then has to relate these different decompositions to explain all the features of that component.

In addition to the "spatial" decomposition of a component into parts, the lifetime of a system can be divided (or decomposed) into time periods called

epochs. The operation of a system can **then** be studied in one epoch at a time. In any one epoch, some components are inactive and can be ignored. Each epoch is then an abstraction, in which a simplified system description is valid. Epochs play an important role in the discussion of distributed computer systems, since there is the potential for different things happening in different places at the same time and some activities may depend upon a long chain of previous events. For example, the transition of a program from source code to executing binary may be broken into several stages - compilation, linking, loading and execution - and there is a need for communication and consistency between all of the stages.

The design of distributed systems often involve trade-offs between "space" and "time". For example data can be transmitted serially or in parallel. All other things being equal, serial transmission takes longer, but only requires one communications link; whereas parallel transmission is faster, but needs more links. Sometimes the designer wants to be explicit about these matters, but at other times only the existence of communication need be discussed. Thus the designer needs to be able to treat space and time as a continuum.

2.3 Physical realizability

Designs are descriptions of proposed systems. Eventually a builder will be given the task of building the system from physical components. A design created through the imagination of the designer is not subject to the constraints of the physical world. Every aspect of the design has to be built into some physical form and will be subject to the basic constraints described in physics. Therefore, designers must check that their designs are physically realizable.

Only two physical constraints are of widespread importance. These are that the propagation of physical effects takes time and that something cannot be created from nothing. Designers will often assume the instantaneous operation of equipment, even though this is not physically feasible. This assumption leads designers to neglect taking into consideration the speed of operation and performance of systems; it leads designers to assume that consequences of actions occur simultaneously in different places and leads them to ignore possible transient inconsistencies in the system; it leads designers to neglect the kind of causal loops that result in deadlock or livelock; and it causes designers to ignore the consequences of design changes, such as increasing the number of connected systems or the physical extent of a system, that modify the rate of propagation of effects and hence the performance of their system, often to the worse.

When compiling a system design it is convenient to disregard the need for the construction of system parts. For example, files are conveniently thought of as being infinitely extensible and stacks as being infinitely long. No system has unlimited resources and attention needs to be paid to the

deployment of resources and the actions to be taken when resources are fully consumed.

A feature of physical components, often due to a lack of knowledge, human error, or poor manufacture, is that they fail and the designer must be aware of which failures are likely to occur and which of these are significant and the kind of effects that can arise.

2.4 References

[ALEXANDER 641 Alexander, C., *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, Massachusetts, USA (1964).

[CHECKLAND 81 I Checkland, P.B., *Systems Thinking, Systems Practice,* Wiley & Sons (1981).

[LE CORBUSIER 541 Le Corbusier, *The Modulor*, Faber & Faber, First English Edition (1954).

[LISKOV 861 Liskov, B. & Guttag, J., *Abstraction and Specification in Program Development,* MIT Press, Cambridge Massachusetts, USA (1986).

[SIMON 62] Simon, H.A., "The Architecture of Complexity", *Proceedings of the American Philosophical Society, 106* (6), 467-482 (December 1962).