

The ANSA Reference Manual

The ANSA Reference Manual

Release 01 .00

March 1989

Architecture Projects Management Limited

Architecture Projects Management Limited and their sponsors take no responsibility for the consequences of errors or omissions in this manual, nor for any damages resulting from the application of the ideas expressed herein

Architecture Projects Management Limited

24 Hills Road
CAMBRIDGE
CB2 1JP
United Kingdom

TELEPHONE UK	Cambridge (0223) 323010
INTERNATIONAL	+ 44 223 323010
FAX	+ 44 223 359779
UUCP	...ukc!acorn!ansa!apm
ARPA Internet	apm@ansa.co.uk

© 1989 Architecture Projects Management Limited

Permission to copy without fee all or part of this material is granted provided that notice is given that copying is by permission of Architecture Projects Management Limited. To copy otherwise or to republish requires specific permission.

Printed by S-Print

52 Clifton Road
CAMBRIDGE
CB1 4FW
United Kingdom
TELEPHONE Cambridge (0223) 3 11334



The ANSA Reference Manual

The ANSA Reference Manual consists of 11 parts, as follows:

Part I	Introduction
Part II	Foundations
Part III	System Modelling
Part IV	Enterprise Projection
Part V	Information Projection
Part VI	Computational Projection
Part VII	Engineering Projection
Part VIII	Technology Projection
Part IX	Formal Basis
Part X	Specifications
Part XI	Implementation Examples and Guidelines



The ANSA Collaboration

The Advanced Networked Systems Architecture originated from the ANSA project which was part of the the United Kingdom Alvey programme. The development of ANSA is being continued by Architecture Projects Management Limited (APM), with the continuing sponsorship of the ANSA project collaborators.

The sponsors are:

British Telecommunications plc (BT)

Digital Equipment Co Ltd (DEC)

GEC Marconi Ltd (GEC)

GEC Plessey Telecommunications Ltd (GPT)

Hewlett-Packard Ltd (HP)

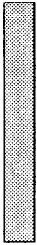
International Computers Ltd (ICL)

Information Technology plc (ITL)

Racal Group Services Ltd

**Standard Telephone and Cable Laboratories Ltd
(STL)**

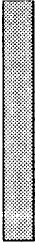
In addition to funding the project, the sponsors have given both time and expertise to the development and review of the architecture.



The ANSA Team

The ANSA Reference Manual is written and maintained jointly by the members of the ANSA team. The members of the team at the time this release was prepared were, in alphabetical order:

Janice Crofton	APM
Dr John Dobson (Consultant)	University of Newcastle
Michael Eyre	APM
Dr Andrew Herbert	APM
Dr Yigal Hoffner	APM
David Iggulden (Consultant)	
Rob van der Linden	APM
Professor John Monk (Consultant)	Open University
David Oliver	ICL
Ed Oskiewicz	British Telecom
Dave Otway	Marconi Instruments
Owen Rees	Racal
Dr Joseph Sventek	Hewlett-Packard
Bill Talbot	APM
Judith Tillotson	APM
Alastair Tocher	STL
John Warne	ICL
John Winterbotham (Consultant)	STL




Distribution

The ANSA *Reference Manual* is an evolving document and is under strict change control. Because of this, it is distributed in two formats. Registered users are supplied with a loose-leaf folder and issued with regular updates, which are subject to change control.

Paperback versions of significant releases are more generally distributed, but without a subsequent updating service.

Those wishing to register should contact the Architecture Projects Management Ltd for further details.



Change control

The ANSA **Reference Manual** is an evolving document and is under strict change control by the ANSA project.

Each release of the **ANSA Reference Manual** has a release number of the form *mm.nn*. The first number, *mm*, is incremented every time a substantial change is made to the contents of the manual; for example when a new part is added. The second number, *nn*, is incremented when more modest changes are made, such as the addition or revision of a section.

Short parts of the manual and individual chapters within longer parts are kept as separate ANSA project documents for convenience of editing and update management. The pages are therefore numbered relative to the part and the chapter within that part. For example, ARM IV, 6/7 refers to page 7 in chapter 6 of part IV.

Changes in documents between different releases are recorded in the editorial history at the beginning of each document. Minor changes between versions are identified in the text by the use of underlining to emphasize new or changed material.

Editorial comments about the contents of a document are shown in the text thus:

Editorial: this is an editorial comment for the guidance of the reader.

Editorial comments are used to indicate known inconsistencies or shortcomings in the text for later resolution.

The status of each document within the **ANSA Reference Manual** reflects the progress of a document through the change control process. The different levels are:

- ▶ **Draft**
The document is in an early stage of drafting and should be treated purely as an indicator of the intended eventual content
- ▶ **Interim**
The document is in an advanced stage of drafting and the content is mostly stable

▶ Approved

The document has been declared formally complete (subject to future correction or extension)

If defects are subsequently found in an approved document, either this document will be downgraded in the next release, and will include an error report, or the defect will be remedied and checked and a new version of the document approved.



Acknowledgements

The ANSA team gratefully acknowledges the encouragement, counsel and support it has received from the Alvey Directorate and the Information Engineering Directorate: in particular Brian Oakley, the first director of the Alvey Programme; Laurence Clarke, the successor to Brian Oakley, who chaired the ANSA Management Committee for almost three years; Keith Bartlett and Chris Cheetham, the subsequent chairmen of the ANSA Management Committee.

The ANSA team has been greatly assisted by major technical contributions resulting from liaison with the ESPRIT projects detailed below and by the encouragement received from the Commission of the European Communities:

COMANDOS	(ESPRIT project 834)
Communications Systems Architecture (CSA)	(ESPRIT project 237)
Delta-4	(ESPRIT project 818)
CIM-OSA	(ESPRIT project 688)

Great benefit to the ANSA project was also derived from the close links formed with other Alvey projects, especially Admiral, PISA and Unison.

Several universities have also helped the ANSA team in their work. Research groups at Cambridge (England), Newcastle, Cornell, Carnegie-Mellon and MIT have brought useful ideas and constructive criticism to the project.

In the field of standards, the ANSA team would like to acknowledge the encouragement and guidance that has been offered by the Department of Trade and Industry IT Standards Unit, the British Standards Institute Committee IST21/7 chaired by Professor Peter Lington and ECMA TC32-TG2 chaired by John Brenner.

The team expresses its gratitude to all the other people who have contributed towards the success of its work, in particular Dr Simon Forge and John Wratten, both of whom have now left the ANSA project, and to Richard Duckworth, who made an important contribution to the development of the computational projection, but who died after a short illness while assigned to the ANSA team.



Contents

	Page
The ANSA Reference Manual	v
The ANSA Collaboration	vii
The ANSA Team	ix
Distribution	xi
Change control	xiii
Acknowledgements	xv
Contents	xvii

PART I: Introduction

1	The Advanced Networked Systems Architecture . . .	I, 1 / 1
	1.1 Conformance	2
	1.2 Recursion	2
	1.3 ANSA and standards	3
2	The Manual	I, 2 / 1

PART II: Foundations

1	Distributed Processing	II, 1 / 1
	1.1 The need for distributed processing	1
	1.2 Viewpoints on distributed processing	2
	1.2.1 Enterprise projection	3
	1.2.2 Information projection	3
	1.2.3 Computational projection	4
	1.2.4 Engineering projection	4
	1.2.5 Technology projection	5
	1.3 Scope of ANSA	5
2	Architectures	II, 2 / 1
	2.1 ANSA and the system design process	2
	2.2 Design techniques	3
	2.2.1 Classification and types	3
	2.2.2 Decomposition and epochs	4
	2.3 Physical realizability	5
	2.4 References	6

PART III: System Modeling

	Introduction	III, 1 / 1
	1.1 About Part III	1
	1.2 Generic concepts	1
	1.3 Criteria of choice	2
	1.4 Reading Part III	3
	1.5 Topics to be covered	3
	1.6 Layout of descriptions	4
	1.7 Notations	5
	1.8 References	7
2	Basic Concepts	III, 2 / 1
	2.1 Introduction	1
	2.2 Reference section	3
	Symbol	4
	Alphabet	6
	Composite symbol	8
	Composite alphabet	10
	Extended alphabet	12
	Refined alphabet	14
	Object	15
	Pair of connected objects	18
	Object with an interface	21
	Complex object	24
	Object with a directed interface	27
	2.3 Examples	30
3	Interaction	III, 3 / 1
	3.1 Introduction	1
	3.2 Reference section	2
	Interaction	3
	Determined interaction	5
	Indetermined interaction	7
	Conflict	9
	Composite interaction	11
	Multiple interaction	13
	Indefinite interaction	15
	Composite sequenced interaction	17
	Multiple sequenced interaction	19
	Indefinite sequenced interaction	21
	3.3 Examples	22
4	Objects	III, 4 / 1
	4.1 Introduction	1
	4.2 Reference section	2
	Simple object	3
	Complex Object	5

	Determined object	7
	Undetermined object	9
	Role	12
	Fault	13
	Causal object	15
	Agent	17
4.3	Examples	19
5	Specialized Interactions	III, 5 / 1
5.1	Introduction	1
5.2	Reference section	2
	Announcement	3
	Operation	5
	Call and reply	7
	Back-call	9
	Nestedcall	11
	Non-atomic action	13
5.3	Examples	17
6	Relationships between Objects	III, 6 / 1
6.1	Introduction	1
6.2	Reference section	2
	Binary relationship	3
	Binary identity relationship	6
	Relational composition	7
	Independence relationship	10
	Dependency relationship	12
	Idempotent relationship	14
	Inverse relationship	16
	Reflexive relationship	17
	Symmetric relationship	19
	Transitive relationship	20
	Function	22
	Injective function	24
	Surjective function	26
	Bijjective function	28
	Function distribution	29
	Ternary relationship	31
	Associative relationship	33
	Commutative relationship	35
6.3	Examples	37
7	Structure	III, 7 / 1
7.1	Introduction	1
7.2	Reference section	1
	Multi-way join	2
	Join	4
	Composite join	6
	Directed join	8

	Combiner	10
	Splitter	12
	Configuration	14
	System as a structure	17
	Isolator	19
	Switch	21
	Controlled configuration	23
	Multiplexer	25
7.3	Examples	26
8	Arrangements of Objects	III, 8 / 1
8.1	Introduction	1
8.2	Reference section	1
	Federation	2
	Hierarchy	6
	Client-server pair	9
	Peer-to-peer arrangement	11
8.3	Examples	13
9	Models	III, 9 / 1
9.1	Introduction	1
9.2	Reference section	2
	Comparator	3
	Replica	5
	Representation	6
	Interpreter	7
	Denotation	9
	Analogue	11
	Transformer	12
	Filter	13
	Model	15
	Modeler	18
	Denotation under development	21
	Design	22
	Property	24
	Attribute	26
	Policy	28
	Class description	30
	Inheritance node	32
9.3	Examples	35
10	Binding	III, 10 / 1
10.1	Introduction	1
10.2	Reference section	2
	Representation of structure	3
	Structure Interpreter	4
	Structure denotation	6
	Association	8
	Collection of associations	10

	Interpreted configuration	12
	Binder	14
	Dynamic binder	16
	Binding..	18
10.3	Examples	20
11	Names	III, 11 / 1
11.1	Introduction	1
11.2	Reference section	2
	Named object	3
	Name	5
	Nameable object	6
	Sentinel	8
	Name interpreter	10
	Naming model	12
	Structured name	14
11.3	Examples	15
12	Resource Management	III, 12 / 1
12.1	Introduction	1
12.2	Reference section	1
	Resource	2
	Collection of resources	4
	Partially resourced object	6
	Shared resource	8
	Environment model	11
	Resource Manager	13
	Factory	15
	Object specification	17
12.3	Examples	21
13	Trading	III, 13 / 1
13.1	Introduction	1
13.2	Reference section	2
	Interface adapter	3
	Interface description	5
	Negotiation	7
	Trader	10
13.3	Examples	13
14	Series of Objects	III, 14 / 1
14.1	Introduction	1
14.2	Reference section	2
	Recurrence relation	3
	Bounded recurrence relations	5
	Bounded sequenced recurrence relation ..	7
	Periodic series	9
	Statemachine	13
	Clock	15

	Stable relation	16
	Synchronized objects	18
	Semaphore	20
	Synchronous state machine	22
14.3	Examples	23
15	Groups	III, 15 / 1
15.1	Introduction	1
15.2	Reference section	2
	Collection	3
	Group	5
	Domain	7
	Communicating group	9
	Addressed group	11
	Shared property group	13
	Replicated group	14
	Fault group	16
	Dynamic group management	18
	Configuration management	20
15.3	Examples	21
15.4	References	24
16	Transparencies	III, 16 / 1
16.1	Introduction	1
16.1	Reference section	2
	Transparency mechanism	3
	Simplified transparency model	6
	Epochs and transparency	9
	Static access transparency mechanism	12
	Static location transparency mechanism	14
	Static address transparency mechanism	17
	Static replication transparency mechanism	20
	Atomic action and transparency	23
	Migration transparency	27
	Dynamic replication transparency mechanism ..	28
	Fault transparency mechanism	30
16.3	Examples	32
17	Technology	III, 17 / 1
17.1	Introduction	1
17.2	Reference section	1
	Infrastructure	2
	Informationprocessor	5
	Message	7
	Memory	9
17.3	Examples	11
17.4	References	14

18	Transformations of Object Diagrams	III, 18 / 1
	18.1 Introduction	1
	18.2 Reference section	1
	Parallel connections	2
	Decomposition	3
	Refinement	6
	Composition	8
	Serial composition	10
	Pseudo independence	12
	Abstraction with epochs	14
	18.2 Examples	16
19	Miscellaneous Concepts	III, 19 / 1
	19.1 Introduction	1
	19.2 Reference section	1
	Library	2
	Directory	4
	Epoch	5
	System	6
	19.3 Examples	8
20	Concept Index	III, 20 / 1

PART IV: Enterprise Projection

1	Introduction	IV,1 / 1
	1.1 The structure of the enterprise projection chapters ...	3
	1.2 The importance of human and organizational issues .	3
	1.3 The environment of distributed systems	4
	1.4 The design process in the context of human and organizational issues	5
	1.4.1 ANSA architectural policies	5
	1.4.2 The need for a dialogue between the designer, the individual and the organization ..	6
	1.4.3 Design decisions as a commitment	7
	1.5 Futurework	8
	1.6 References	9
 2	 Human Issues	 IV,2 / 1
	2.1 Criticisms of approaches to system design	1
	2.2 A system design approach	2
	2.3 People and systems	3
	2.3.1 Users	4
	2.3.2 Non-users	5
	2.4 The human aspect	6
	2.4.1 Human concerns and constraints	6
	2.4.2 Human attributes	8
	2.4.3 Human interactions	8
	2.5 The link between human issues and the architecture	8
	2.6 Modelling people	10
	2.7 Summary	11
	2.8 References	12
 3	 Organizational Issues	 IV,3 / 1
	3.1 Organizations and new technology	1
	3.2 People within organizations and systems	2
	3.3 Organizational aspects	3
	3.4 Organizational structure	5
	3.5 Designing the organization	7
	3.6 Organizational policies	8
	3.6.1 Organizations and policies	9
	3.6.2 ANSA and organizational policies	9
	3.7 Roles	10
	3.7.1 Structuralviewofroles	11
	3.7.2 Functional view of roles	11
	3.7.3 The concept of roles and ANSA	12
	3.8 Organizations and systems	12
	3.9 References	14

4	Enterprise Modelling	IV, 4 / 1
4.1	A survey of organizational modelling techniques	1
4.2	The basic components	2
	4.2.1 Intra-entity Relations	2
	4.2.1.1 Agents	2
	4.2.1.2 Activities	3
	4.2.1.3 Data	3
	4.2.2 Inter-entity relations	3
	4.2.2.1 Agent-activity relations	4
	4.2.2.2 Agent-data relations	4
	4.2.3 The concept of interpretation	4
4.3	Other enterprise modelling techniques	5
	4.3.1 ANSA position on other enterprise modelling techniques	5
	4.3.2 ANSA requirements from other enterprise modelling techniques	6
	4.3.3 A comparison with other enterprise modelling techniques	6
4.4	An enterprise modelling example	6
4.5	References	8
5	Organizational and Architectural Issues	IV, 5 / 1
5.1	Organizational and architectural links	1
5.2	Distributed system development issues	2
	5.2.1 Centralized and distributed views of system development	2
	5.2.2 Object-oriented design	3
	5.2.3 The client-server model	3
	5.2.4 Trading issues	3
	5.2.5 Distribution transparency issues	6
5.3	Dependability issues	7
	5.3.1 Fault tolerance issues	7
	5.3.2 Availability issues	8
	5.3.3 Safety and security issues	8
5.4	Naming issues	10
5.5	Usability issues	11
5.6	Management issues	12
	5.6.1 Management policies	12
5.7	Data and information management issues	13
5.8	References	14

PART V: Information Projection

	Introduction	V, 1 / 1
	1.1 The structure of the information projection chapters ..	2
	1.2 Data and information	3
	1.3 Modelling and the information projection	5
	1.3.1 “Real” world entities and their representation ..	5
	1.3.2 Consistency and coherence	6
	1.4 Data, information and distribution	7
	1.5 Data, information, interpretation and human-computer interaction	9
	1.6 References	10
2	Data Engineering	V, 2 / 1
	2.1 The data engineering language	2
	2.1.1 Structure: Data and meta-data	2
	2.1.2 Behaviour: Abstract data types	2
	2.1.3 Constraints: Semantic integrity	3
	2.1.4 Representation: Entity, relations and attributes ..	3
	2.2 General requirements from a data management system ..	4
	2.2.1 Data items: existence requirements	4
	2.2.2 Data items: manipulation requirements	4
	2.2.3 Legality constraints	5
	2.2.4 Data items: representation requirements	5
	2.3 Identification requirements	7
	2.4 Typing	8
	2.4.1 Type definitions	8
	2.5 Dependability requirements	9
	2.5.1 Security requirements	9
	2.6 Integrity requirements	12
	2.6.1 Consistency requirements	12
	2.6.2 Coherence requirements	13
	2.7 Data independence requirements	14
	2.8 References	17
3	Information Engineering	V, 3 / 1
	3.1 Survey of information representation techniques	1
	3.1.1 Formal logic	2
	3.1.2 Semanticnets	2
	3.1.3 Frame based systems	2
	3.1.4 Conceptual dependency	2
	3.1.5 Scripts	3
	3.2 Speechacttheory	3
	3.2.1 Basicconcepts	5
	3.2.2 Speechacts	5
	3.2.3 The structure of an illocutionary act	6
	3.2.4 Conversation structure	7
	3.2.5 Policy	7

	3.2.6 Application of speech acts to security policy ...	8
	3.2.3 Summary	11
3.3	An information engineering language	11
3.4	References	13
4	Data, Information and Human-Computer Interaction V, 4/	1
4.1	Interpretation and human-computer interaction	1
	4.1.1 Interpretation and the user interface	1
	4.1.2 Interpretation and the human processor model	2
4.2	The user interface	2
4.3	The user interface and distribution	3
4.4	A user interface structural model	4
4.5	Terminology	5
	4.5.1 The User Interface (UI)	5
	4.5.2 Application semantics and pragmatics	6
	4.5.3 The scope of the proposed UI model	7
4.6	Structuring concepts of the user interface model	7
	4.6.1 The language stage concept	7
	4.6.2 The language transformation step concept	8
	4.6.3 Summary	9
4.7	Applying the new user interface model in the design of a UI for a chess playing application	10
	4.7.1 Definition of problem and resources	11
	4.7.2 The task of the UI	13
	4.7.3 Design decisions	13
	4.7.4 The single Canonical translation and interpretation step UI	13
	4.7.5 The multi Canonical translation and interpretation step UI	14
	4.7.6 Extensions and modifications	16
	4.7.7 Summary	17
4.8	A comparison of the UI model with other models ...	17
4.9	Human-computer interaction standardization issues	18
	4.9.1 The usability standardization trend	18
	4.9.2 The technical standardization trend	19
	4.9.3 Standardization of the user interface	20
4.9	References	21
5	Trading	V, 5 / 1
5.1	Information viewpoint of trading	1
	5.1.1 Definitions	1
	5.1.2 The tradingspace	2
	5.1.3 Navigation through the trading space	4
	5.1.4 Trading domains	5
	5.1.5 Federated trading	6

PART VI: Computational Projection

1	Introduction	VI, 1 / 1
	1.1 Position of the model	1
	1.2 Distribution transparency	2
	1.3 Separation	3
	1.4 Evaluation epochs	3
	1.5 Evolution	4
	1.6 Type checking	4
	1.7 Configuration	5
	1.8 Interfaces	5
	1.9 Operations	6
	1.10 Invocations	7
	1.11 Terminations	8
	1.12 Atomicity	8
	1.13 Objects	8
	1.14 References	9
2	Distributed Computation Model	VI, 2 / 1
	2.1 Basic elements	1
	2.1.1 Objects and interfaces	1
	2.1.2 Operations and invocations	3
	2.1.3 The structure of invocations	3
	2.2 Types and conformance	5
	2.2.1 Operation types	5
	2.2.2 Interface types	5
	2.2.3 Expression types	6
	2.2.4 Type conformance	6
	2.3 Internal model	7
	2.3.1 Operation bodies	8
	2.3.2 Invocation	8
	2.3.3 Object state manipulation	9
	2.3.4 Composition of expressions	10
	2.3.5 Choosing and using terminations	10
	2.3.6 Concurrency control	10
	2.4 References	11
3	General Computational Model	VI, 3 / 1
	3.1 Introduction	1
	3.2 Assumptions	2
	3.2.1 The system and its environment	2
	3.2.2 Faults and errors	3
	3.2.3 Resource limitations	4
	3.2.4 Values and references	4
	3.2.5 Naming	4
	3.3 The position of the model	5
	3.3.1 The computer, people and the environment	5
	3.3.2 A model of the system in its environment	6

3.4	Computers, programs and computational objects	7
3.5	Computational objects	10
	3.5.1 Value	10
	3.5.2 Denotation	11
	3.5.3 Node	12
	3.5.4 Network	12
	3.5.5 Resource	12
3.6	Interactions between computational objects	13
	3.6.1 Operation	13
	3.6.2 Performing an operation	13
	3.6.3 Invocation	14
	3.6.4 Evaluation	15
3.7	References	15
4	Naming and Binding	VI, 4 / 1
	4.1 Introduction	1
	4.2 Computational binding	2
	4.3 Naming	3
	4.3.1 Names and context	4
	4.3.2 Decomposing a computational node	4
	4.3.3 The naming of values	5
	4.3.4 Variables and constants	6
	4.3.5 Structured values and variables	7
	4.3.6 Pathnames	7
	4.4 Assignment	8
	4.5 The design enterprise	9
5	Invocation Model	VI, 5 / 1
	5.1 Interaction between computational objects	1
	5.2 The basic model	1
	5.3 Impact of resource allocation issues on the basic computational model	3
	5.3.1 Resource use in invoked objects	4
	5.3.2 The client server view	4
	5.3.3 Synchronization	5
	5.3.4 Mechanization of the invocation model	6
	5.4 Argument and result passing strategies	7
	5.4.1 Restrictions imposed by the general invocation model	8
	5.4.2 Constraining access to the invocation record ..	8
	5.4.3 Message passing	9
	5.4.4 Pass by value	9
	5.4.5 Pass by reference	10
	5.4.6 Pass by copy	10
	5.4.7 Pass by capability	10
	5.4.8 Changing representations	11
	5.5 References	11

6	Evaluation Model	VI, 6 / 1
6.1	General model	1
6.2	The evaluation context	2
6.2.1	The execution context	3
6.3	Evaluating denotation	3
6.3.1	The dynamics of the evaluation context	4
6.3.2	Parallel and sequential evaluation	4
6.3.3	Evaluating a reference to a denotation	5
6.3.4	Evaluating a value	5
6.3.5	Evaluating a piece of the environment	5
6.3.6	Operations defined in the execution context ...	6
6.3.7	Relation with the invocation model	6
6.3.8	Deferred evaluation	7
6.3.9	Termination	7
6.4	Fragmentation of the denotation	7
6.4.1	Single denotation fragment	8
6.4.2	Multiple independent denotation fragments ...	9
6.4.3	Partially ordered sets of denotation fragments	11
6.4.4	Multiple dependent denotation fragments	12
7	Typing	VI, 7 / 1
7.1	Typing and projections	1
7.2	Reasons for typing	2
7.3	Type checking in the computational projection	2
7.3.1	Evaluation of server denotation	3
7.3.2	Evaluation of client denotation	4
7.3.3	Conflict detection	4
8	Distribution of Computation	VI, 8 / 1
8.1	Distribution of application programs	1
8.2	Distributing computation	1
8.2.1	Configuration management	2
8.2.2	Multiple heterogeneous execution contexts ...	2
8.3	Example	2
9	Transparency	VI, 9 / 1
9.1	Computational transparency	1
9.2	Basic transparencies	2
9.2.1	Access transparency	2
9.2.2	Location transparency	3
9.2.3	Migration transparency	5
9.2.4	Identity transparency	5
9.2.5	Replication transparency	5
9.2.6	Concurrency transparency	6
9.2.7	Fault transparency	7

10	Modeling ANSA Computational Objects	VI, 10 / 1
10.1	An example of a computational object	1
	10.1.1 Providing multiple interfaces	2
10.2	ANSA conforming computational objects	3
10.3	Other ANSA conforming objects	4
11	Trading	VI, 11 / 1
11.1	General interface to a trading service	1
	11.1.1 Interface specification	1
	11.1.2 Service providers	2
	11.1.3 Service requesters	3
11.2	Administrative interfaces to a trading service	4
	11.2.1 Type lattice interface	4
	11.2.2 Context lattice interface	5
	11.2.3 Access management interface	6
	11.2.4 Federation interface	7

PART VII: Engineering Projection

1	Introduction	VII, 1 / 1
	1.1 Position of the model	2
	1.2 Structure of the model	3
	1.3 Architectural recursion	4
	1.4 Interworking	6
	1.5 Structure of Part VII	6
2	Background	VII, 2 / 1
	2.1 Technical assumptions behind ANSA	1
	2.2 Programming languages	2
	2.3 Remote procedure call	3
	2.4 Parallel execution	5
	2.4.1 Transactions	5
	2.4.2 Replication	5
	2.5 Operating systems	6
	2.5.1 Virtual memory	6
	2.5.2 Lightweight processes	6
	2.5.3 Interprocess communication	7
	2.5.4 Networked operating systems	7
	2.5.5 Distributed operating systems	7
	2.6 Protocol design	8
	2.7 Multi-media integration	9
	2.8 Heterogeneity	9
	2.9 Security	9
	2.10 Large systems	10
	2.11 References	10
3	Nucleus	VII, 3 / 1
	3.1 Capsules: creation and destruction	2
	3.2 Scheduling	3
	3.2.1 Threads	4
	3.2.2 Tasks	5
	3.2.3 Thread factory interface	5
	3.2.4 Tasks	7
	3.2.5 Evaluators	7
	3.3 The poller	8
	3.4 Scheduling and deadlines	9
	3.5 Synchronization	10
	3.6 Resource limits	11
	3.7 Communications	11
	3.7.1 Plugs and sockets	12
	3.7.2 Remote interactions	15
	3.7.3 Alarms	17
	3.7.4 Session caching and decay	17
	3.7.5 Protocol table	18
	3.7.6 Execution protocols	18

	3.7.7 Message passing services	19
	3.7.8 Buffers	20
3.8	Killing threads	21
3.9	Clock	21
3.10	Interface identifiers	22
3.11	References	23
4	Access Transparency	VII, 4 / 1
4.1	Marshalling	2
4.2	Dispatching	3
4.3	Interface representations	4
4.4	Pipelining	5
5	Protocols	VII, 5 / 1
5.1	REX	1
	5.1.1 Channels	2
	5.1.2 Threads and buffers	2
	5.1.3 Server dispatching	3
	5.1.4 Calls	3
	5.1.5 Casts	5
	5.1.6 Rate controlled fragmentation	6
	5.1.7 Packet format	7
5.2	Group execution protocol	8
	5.2.1 Group interaction	9
	5.2.2 Communications guarantees	11
	5.2.3 Destination ordering	12
	5.2.4 Source ordering without strict destination ordering	13
5.3	References	13
6	Location Transparency	VII, 6 / 1
6.1	Location transparency	1
	6.1.1 Interface references	1
	6.1.2 Locators	2
	6.1.3 Interface identifiers	3
	6.1.4 The trader and location transparency	4
6.2	Configuration management	5
6.3	Migration transparency	6
7	Atomic Object Model	VII, 7 / 1
7.1	Atomic Objects and Operations	1
	7.1.1 Properties of atomic operations	1
	7.1.2 Review of properties	2
7.2	Atomic operation trees	2
7.3	Spheres of control	3
7.4	Atomic object structure	4
7.5	Atomic operation manager	6
	7.5.1 Basic primitives and interactions	6

7.6	Special control primitives	7
	7.6.1 Composite atomic operations	7
	7.6.2 Spawned atomic operations	8
7.7	Concurrency control manager	8
	7.7.1 Notation	8
	7.7.2 Mode evaluations	9
	7.7.3 Concurrency predicate evaluations	10
	7.7.4 Comments on concurrency control model	11
7.8	Version store manager	11
	7.8.1 Version store principles	11
	7.8.2 Version store model	12
	7.8.3 Alternative access strategy for version stores	13
	7.8.4 Details of version store creation and deletion	14
	7.8.5 Comment on composite atomic operations	16
	7.8.6 Note on optimistic concurrency control and version stores	16
	7.8.7 Compensation methods	17
7.9	Atomic operation tree commit/abort protocols	17
	7.9.1 Multiphase commit/abort principles	17
	7.9.2 Failure review	20
	7.9.3 Comment on blocking and k-resiliency	22
	7.9.4 Hierarchical commit protocol	22
	7.9.5 Hierarchical Abort Protocol	23
7.10	Connecting heterogeneous atomic object systems	24
7.11	Deadlock avoidance, prevention, detection and resolution	25
7.12	References	26
8	Object Group Model	VII, 8 / 1
	8.1 Motivation	1
	8.2 Outline	1
	8.3 Object group classifications	2
	8.3.1 Functionally distributed group	2
	8.3.2 Coordinated replica group	3
	8.3.3 Parallel replica group	4
	8.3.4 Comment on group types	5
	8.4 Object group engineering model	5
	8.5 Note on functionally distributed groups	7
	8.6 Coordinated replica group behaviour	7
	8.6.1 Checkpointing and retained results	7
	8.6.2 Operation invocation	8
	8.6.3 Operation termination	8
	8.6.4 Operation invocation revisited	8
	8.6.5 Operation termination revisited	8
	8.6.6 Checkpoint frequency	9
	8.6.7 Comment on replication transparency	9
	8.6.8 Coordinator election	9
	8.6.9 Coordinator synchronization	10

8.7	Parallel replica group behaviour	11
	8.7.1 Operational principles	12
	8.7.2 Sequenced delivery	12
	8.7.3 Ordered processing	12
	8.7.4 Scheduling non-atomic operations in parallel replica groups	13
	8.7.5 Scheduling atomic operations in parallel object groups	13
	8.7.6 Collating requests and results	13
	8.7.7 N-version programmed groups	14
8.8	Interactions between different group types	15
	8.8.1 Mixed client-server group interactions	15
	8.8.2 Hybrid replica group interactions	16
8.9	Object group manager	17
	8.9.1 Object group replication domain	18
	8.9.2 Object group registration	18
	8.9.3 Object group instantiation and service trading	19
8.10	References	21

PART VIII: Technology Projection

1	Implementation Options	VIII, 1 / 1
1.1	Computational model	1
	1.1.2 Programming languages	1
1.2	Engineering model	2
	1.2.1 Nucleus	3
	1.2.2 Interpreters, compilers, libraries and stubs ...	5

PART IX: Formal Basis

	Introduction	IX, 1 / 1
	1.1 About Part IX	1
	1.1.1 The chapters	1
	1.2 Models	2
	1.3 Closed systems	2
	1.4 Requirements for modelling techniques	3
	1.5 Modelling systems in ANSA	4
	1.6 Explaining ANSA	5
2	Assumptions	IX, 2 / 1
	2.1 Objects	1
	2.2 Object types, abstraction and decomposition	3
	2.3 Links between objects	5
	2.4 Development of a protocol design	8
	2.4.1 Getting started	8
	2.4.2 Refining the design: fault detection	8
	2.4.3 Refining the design: physical representations ..	9
	2.4.4 Refining the design: choosing the dimension ..	9
	2.4.5 Refining the design: choosing the direction ..	10
	2.4.6 Summary	10
3	Notations	IX, 3 / 1
	3.1 The requirement	1
	3.2 Review	2
	3.2.1 Everyday language	2
	3.2.2 Specialized notations	3
	3.2.3 Algebras	4
	3.2.4 Finite state machines	4
	3.2.5 Grammars	5
	3.2.6 Net theories	5
	3.2.7 Axiomatic notations	5
	3.2.8 Temporal logic	5
	3.3 A notation for object diagrams	5
	3.3.1 Object models	6
	3.3.2 Objects and names	6
	3.3.3 Restrictions and their resolution	7
	3.3.4 Names for bindings	8
	3.3.5 Structured names	9
	3.3.6 Connections as classes	10
	3.3.7 Language and bindings	11
	3.3.8 Responsibilities	12
	3.4 References	13
4	Theorems	IX, 4 / 1
	4.1 Introduction	1
	4.2 Summary of interactions	2

	4.3	Summary of objects	3
	4.4	Summary of specialized interactions	4
	4.5	Constraint to a single symbol	5
	4.6	Object duplication	5
	4.7	Constant relationship	6
	4.8	Precedence of causality	7
	4.9	Object decomposition	7
	4.10	Decomposition of bindings	7
	4.11	Epochs	8
5		Satisfaction of Specifications	IX, 5 / 1
	5.1	Introduction	1
	5.2	The problem	1
	5.3	Solutions	1
		5.3.1 Simple solution	2
		5.3.2 Direct translation	2
		5.3.3 Subsetting	3
		5.3.4 Division	4
	5.4	Implementation	4

PART X: Specifications

1	Introduction	X, 1/1
1.1	Conformance	2
2	Interface and invocation specifications	X, 2/1
2.1	Software engineering environments	1
2.2	Interface specifications	2
2.3	Interface Definition Language	3
2.3.1	Basic data types	3
2.3.2	IDL grammar	7
2.4	Specifying interactions	10
2.4.1	Lexical scoping	10
2.4.2	Exception syntax	10
2.5	DPL statements	12
3	Service Interfaces	X, 3/1
	Trader	2
	TrCtxt	8
	TrType	11
4	Support Environment	X, 4/1
	Import	2
	Initiation	3
	Invocation	4
	Redeem	5
	Discard	6
	Export	7
	Withdraw	8
	Terminate	9
5	Remote Execution Protocol	X, 5/1
5.1	Summary of the REX protocol	1
5.2	Message sequence numbers	2
5.3	Implicit and explicit acknowledgements	2
5.4	REX packet format	3
5.5	REX and session states	3
5.5.1	Notes on the protocol state tables	4
6	Trader	X, 6/1
6.1	Interface matching criteria	2
6.2	Constraint language	2
6.2.1	BNF for the constraint language	3
6.2.2	Examples of constraint expressions	4
7	Computational Model	X, 7/1
7.1	Introduction	1
7.1.1	Aims of DPL	2
7.1.2	Concepts	3

PART X: Specifications

1	Introduction	X, 1 / 1
	1.1 Conformance	2
2	Interface and invocation specifications	X, 2 / 1
	2.1 Software engineering environments	1
	2.2 Interface specifications	2
	2.3 Interface Definition Language	3
	2.3.1 Basic data types	3
	2.3.2 IDL grammar	7
	2.4 Specifying interactions	11
	2.4.1 Lexical scoping	11
	2.4.2 Exception syntax	11
	2.5 DPL statements	13
3	Service Interfaces	x, 3 / 1
	Trader	2
	TrCtxt	8
	TrType	11
	TrFed	14
4	Support Environment	x, 4 / 1
	Import	2
	Initiation	3
	Invocation	4
	Redeem	5
	Discard	6
	Export	7
	Withdraw	8
	Terminate	9
5	Remote Execution Protocol	X, 5 / 1
	5.1 Summary of the REX protocol	1
	5.2 Message sequence numbers	2
	5.3 Implicit and explicit acknowledgements	2
	5.4 REX packet format	3
	5.5 REX and session states	3
	5.5.1 Notes on the protocol state tables	4
6	Trader	X, 6 / 1
	6.1 Interface matching criteria	2
	6.2 Constraint language	2
	6.2.1 BNF for the constraint language	3
	6.2.2 Examples of constraint expressions	4

7	Computational Model	x, 7/1
	7.1 Introduction	1
	7.1.1 Aims of DPL	2
	7.1.2 Concepts	3
	7.2 Syntax	7
	7.2.1 The syntactic metalanguage	8
	7.2.2 The syntax definition	8
	7.3 Range and scope of identifiers	13
	7.3.1 Range	13
	7.3.2 Scope	15
	7.4 Type semantics	15
	7.4.1 The type system	15
	7.4.2 The type semantics definition	29
	7.5 Evaluation semantics	52
	7.5.1 The evaluation semantics model	52
	7.5.2 The evaluation semantics definition	52
	7.6 Future enhancements	66
	7.6.1 Requirement and provision properties	66
	7.6.2 Behavioural constraints	71
	7.6.3 Computational independence versus retention of results	71
	7.6.4 Atomicity and undoing work already done ...	72
	7.6.5 Piped operation groups	74
	7.7 Mathematical glossary	76
	7.7.1 Logic	76
	7.7.2 Sets	77
	7.7.3 Numbers	77
	7.7.4 Relations	78
	7.7.5 Functions	78
	7.7.6 Sequences	78
	7.7.7 DPL types	79
	7.8 References	80

PART XI: Implementation

1	Introduction	XI, 1 / 1
	1.1 The ANSA testbench	1
	1.1.1 Content of the ANSA testbench	1
	1.2 UNIX testbench specifics	2
	1.3 Structure of Part XI	2
	1.3.1 Manual page format	3
	1.4 Overall structure	4
	1.4.1 Computational model	4
	1.4.2 Engineering model	4
	1.4.3 Implementation options	5
	1.4.4 Process mapping	6
	1.4.5 Implementation of engineering concepts	7
2	Building a Capsule	XI, 2 / 1
	2.1 Interface specifications and stub compilation	1
	2.2 Preprocessing the distributed processing language ..	4
	2.3 Acompleteexample	4
	2.3.1 The interface specification	5
	2.3.2 The server program	5
	2.3.3 A possible client program	7
	2.3.4 The Makefile	8
3	Testbench Tracing Facilities	XI, 3 / 1
	3.1 Design goals for diagnostic facilities	1
	3.2 C debugging and tracing facilities	1
4	Interpreter and Nucleus	XI, 4 / 1
	4.1 Module linkage	1
	4.2 Data structures	1
	4.3 Multi-threading	2
	4.4 Sessions	3
	4.5 Instructions	3
	4.6 Scheduling	3
	4.7 Communication protocols	3
	4.8 Nucleus	4
5	REX Implementation	XI, 5 / 1
	5.1 UNIX process structure	1
	5.2 Names and addresses	2
	5.3 REX interfaces	2
	5.3.1 REX / interpreter linkage	2
	5.3.2 REX/MPS linkage	3
	5.4 Timeouts and retransmissions	3
6	Message Passing Service Implementation	XI, 6 / 1
	6.1 UDP message passing service	1
	6.1.1 MPS_startup	1

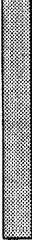
	6.1.2 MPS_sendMsg	1
	6.1.3 MPS_receiveMsg	1
	6.1.4 MPS_tick and MPS_cleanup	1
6.2	TCP message passing service	2
	6.2.1 MPS_startup	2
	6.2.2 MPS_send	3
	6.2.3 MPS_receive	3
	6.2.4 MPS_decay	3
	6.2.5 MPS_cleanup	4
6.3	IPC message passing service	5
	6.3.1 MPS_startup	5
	6.3.2 MPS_sendMsg	5
	6.3.3 MPS_receiveMsg	5
	6.3.4 MPS_tick	5
	6.3.5 MPS_cleanup	6
	6.3.6 Possible enhancements	6
7	Trader Implementation	XI, 7 / 1
	7.1 Trading	1
	7.2 Federation.....	1
	7.3 Implementation details	2
8	The Stub Compiler	XI, 8 / 1
	8.1 Purpose	1
	8.2 Datatypes	1
	8.3 Function signatures	1
	8.4 Subtypes	1
	8.5 Implementation specifics	2
9	D P L Pre-processor	XI, 9 / 1
	9.1 DPL pre-processor	1
10	Service Commands	XI, 10 / 1
	Addname	2
	Addtype	4
	Offerpurge	6
	Prepc	8
	Services	10
	Stubc	11
	Supsrv	13
	Trader	15
	Trsearch	17
	Trshut	19
11	Internal Functions	XI, 11 / 1
	Binder	2
	Buffer	5
	Channel	8
	ECS	11

	Idcache	14
	Instruct	16
	MPS	20
	MPSIPC	23
	MPSTCP	25
	MPSUDP	28
	Nucleus	30
	pin	33
	Protocol	36
	Protocol table	38
	REX	41
	Schedule	45
	Session	47
	stack	50
	System	52
	Task	55
	Thread	58
12	Installation	XI, 12 / 1
	12.1 UNIX	1
	12.1.1 The testbench distribution tape	1
	12.1.2 Preparing to read the tape	1
	12.1.2.1 The testbench logical root directory	2
	12.1.2.2 The trader host	2
	12.1.2.3 The host system	2
	12.1.2.4 The MPS modules	2
	12.1.3 Installing the testbench	3
	12.1.3.1 Non-standard port numbers	4
	12.1.4 Installation tests	4
	12.1.4.1 Starting the trader and exercising it	5
	12.1.4.2 The Echo service	6
	12.1.4.3 The Sample service	7
	12.1.4.4 The Netinfo service	7
	12.1.4.5 The test1 service	8
	12.1.5 Support and further developments	8
	12.2 MSDOS	9
	12.2.1 Requisite hardware and software	9
	12.2.2 Preparation for installation	10
	12.2.3 Installation procedure	11
	12.2.4 Installation tests	12
	12.2.4.1 REX and interpreter test	12
	12.2.4.2 Exercising the trader	12
	12.2.4.3 The Echo service	13
	12.2.4.4 The Sample service	13
	12.2.4.5 The Netinfo service	14
	12.2.4.6 The test1 service	14
	12.2.5 PC limitations	14

ANSA Reference Manual

Release 01 .00

Part I: Introduction



Part I

Chapter 2

The manual

The ANSA **Reference Manual** is divided into eleven parts, entitled as follows:

I	Introduction
II	Foundations
III	System modelling
IV	Enterprise projection
V	Information projection
VI	Computational projection
VII	Engineering projection
VIII	Technology projection
IX	Formal basis
X	Specifications
XI	Implementation examples and guidelines

Part I explains the purpose of, and concepts behind ANSA and provides a roadmap of the manual.

The newcomer to ANSA is recommended to start by reading Part II, which describes the background to the design of ANSA and reviews the concepts that underpin the architecture. It begins with an introduction to, and justification of the five design **schemas** or projections of ANSA and then explains in more detail the requirements upon ANSA.

Part III describes ANSA's modelling notation and defines modelling concepts that are common across all five projections. The description of these common concepts has resulted in a library of design templates. The templates are used to explain the function, structure and behaviour of distributed systems in subsequent parts of the manual. Part III is the keystone of the manual; it defines all the basic architectural concepts and has been organized in a dictionary format. The modelling notation described gives particular emphasis to the structuring of distributed systems.

Parts IV-VIII describe the particular distributed system concepts used in ANSA and the range of design options and trade-offs available to the system designer. Each part describes a separate projection and relates the concepts in that projection to their counterparts in the other projections. The order of

the parts is a consequence of editorial policy rather than any implicit layering in the projections themselves.

Part IX presents a formal basis for the notation and models used in ANSA. It is intended for the reader with an interest in the style of formal specification used in ANSA.

Part X gives specifications for the components of distributed systems identified in Parts IV-VIII.

Part XI describes the reference implementation of ANSA maintained by the ANSA team as an example of how to use ANSA in practical situations.

Part I

Chapter 1

The Advanced Networked Systems Architecture

The Advanced Networked Systems Architecture (**ANSA**) supports the design, implementation, operation and evolution of distributed information processing systems where the different components that make up the system, such as applications packages, operating systems, computers and networks, come from different vendors. The complexity, and consequent cost, that arises from this heterogeneity of hardware and software can be significantly reduced if information technology vendors adopt a common approach to the design and interconnection of the components they offer.

Common design principles based on a consistent model of information processing across the system ease the task of integrating diverse applications packages into a coherent, extensible system.

For system operators, the common basis allows for a consistent approach to the day-to-day management of the system, towards growth of the system to accommodate new requirements and in adapting to changes in the information technology policies of the system owners.

For system implementors, common design principles enhance productivity by enabling software re-use and facilitating the automatic generation of system-specific software from declarative statements of requirements. Application designers are freed from the need to understand how particular functions are provided in every target system. Use of common design principles also reduces the conceptual gap that designs must bridge when separate systems are to become interconnected.

The ANSA team formalized and systematized research and development experience in distributed processing into a framework of concepts, design rules and implementation recipes. ANSA links design principles to implementation techniques that allow the use of common components in different systems and enables compatible systems across a large range of application domains.

ANSA has five projections called enterprise, information, computational, engineering and technology described in Parts IV to VIII. The enterprise projection is concerned with modelling the relationships between the system owners, operators and users. The information projection is concerned with

the modelling of the information-bearing components of a system and the information processes in a system. The computational projection is concerned with the specification of algorithms for information processes and the software engineering aspects of distributed processing. The engineering projection is concerned with the provision of operating systems and networking support for distributed processing. The technology projection is concerned with the use of standard components to realize designs in the other projections. Each projection includes within its own definition an abstraction of the concepts defined in all the other projections so that design rules and principles can be given for transforming a design expressed in one projection into the same design re-expressed in another, but emphasizing different concerns.

1.1 Conformance

The *ANSA Reference Manual* defines the conformance criteria for ANSA projection by projection. It is possible for a system to conform in any number of the projections and conformance to each projection brings different benefits.

The enterprise and information projections of ANSA can be used to establish a design model of information sources, sinks and processes that meet the requirements of the enterprise that requested the system. Conformance requirements in these projections identify constraints on the conceptual schema of the system information base and on system management policies necessary to enable distributed processing.

The computational projection can be used to transform an information projection model into a network of interacting computer programs. Conformance requirements at this level identify constraints on programming language structures to enable distributed processing. Programs that conform to this projection will have the potential to interwork with other conforming programs and will be portable across all machine environments that conform to the projection.

The engineering projection can be used to transform a computational projection model into a model in terms of processing, memory and communication functions. The conformance requirements here identify constraints on systems that independently conform to the architecture necessary to enable their interconnection.

1.2 Recursion

ANSA is recursive in the sense that the components of an ANSA system can be modelled (and indeed implemented) as a network of other ANSA systems.. This enables designers to re-use components from one system in another and to provide opportunities for distributed implementation of system components.

1.3 ANSA and standards

Distributed processing is an emergent area of international standardization. ANSA is complementary to, and an extension of this standardization. Material from the *ANSA Reference Manual* has contributed to the progress of the International Standards Organization JTC1/SC21/WG7 'Open Distributed Processing' (ODP) work item via the British Standards Institute (BSI) Committee IST21/7 and the European Computer Manufacturers Association (ECMA) Committee TC32/TG2.

