



---

**Poseidon House  
Castle Park  
Cambridge CB3 0RD  
United Kingdom**

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (01223) 515010  
+44 1223 515010  
+44 1223 359779  
apm@ansa.co.uk**

---

## **ANSA Phase III**

# **Federation Future Work Plan**

**Yigal Hoffner and Ben Crawford**

### **Abstract**

The plan will show the directions in which future work involving trading, interception, boundaries and domain concepts should proceed with a view towards management in large scale distributed systems.

---

APM.1404.00.01

**Draft**

28th February 1995

Request for Comments (confidential to ANSA consortium for 2 years)

---

**Distribution:**

**Supersedes:**

**Superseded by:**



## **Federation Future Work Plan**





## **Federation Future Work Plan**

Yigal Hoffner and Ben Crawford

APM.1404.00.01

28th February 1995

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by Architecture Projects Management Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

## Architecture Projects Management Limited

Poseidon House  
Castle Park  
CAMBRIDGE  
CB3 0RD  
United Kingdom

TELEPHONE UK  
INTERNATIONAL  
FAX  
E-MAIL

(01223) 515010  
+44 1223 515010  
+44 1223 359779  
[apm@ansa.co.uk](mailto:apm@ansa.co.uk)

**Copyright © 1995 Architecture Projects Management Limited**  
**The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.**

Architecture Projects Management Limited takes no responsibility for the consequences of errors or omissions in this Report, nor for any damages resulting from the application of the ideas expressed herein.

---

# Contents

---

<b>3</b>	<b>1</b>	<b>Introduction</b>
3	1.1	The business case
3	1.2	Overview of present work
4	1.3	Towards an Architecture of Federation
5	1.4	Federation work topics (Figure 1.1)
5	1.4.1	Work to date
5	1.4.2	Future work
5	1.4.3	Relationship to other Phase III work
6	1.5	Division of work within areas
6	1.6	Background material
<b>7</b>	<b>2</b>	<b>Trader federation</b>
7	2.1	Introduction
8	2.1.1	Background
8	2.1.2	Content
8	2.2	Linking traders
8	2.3	Minimal trading configuration
9	2.4	trading domains
9	2.5	Visibility (advertizing/search policy) issues
10	2.5.1	Specifying Advertizing/Searching policy issues
10	2.6	Trading and QoS issues
10	2.7	Trading as an interception process
11	2.8	Domain hopping (Indirect paths)
11	2.9	Extension to trading model
11	2.10	Trading and CORBA repositories
11	2.11	Design and implementation
11	2.11.1	Enhancements to Trader and Match maker
<b>12</b>	<b>3</b>	<b>Domains and domain management</b>
12	3.1	Introduction
12	3.1.1	Problems with the concept of domains
12	3.1.2	Background
12	3.1.3	Content
13	3.2	Heavy-weight model of domains
13	3.3	Light-weight domain model
14	3.4	Domain management
14	3.5	Information needed to construct domains and gateways
14	3.5.1	Design of domain and management
14	3.6	Applying domain management to different scenarios
14	3.6.1	security domain
14	3.6.2	remuneration domain
14	3.6.3	Organizational domain
14	3.6.4	Design and implementation

---

<b>15</b>	<b>4</b>	<b>Miscellaneous issues</b>
15	4.1	<i>Introduction</i>
<b>16</b>	<b>5</b>	<b>Protoype implementation</b>
16	5.1	Introduction
16	5.2	Suggested prototypes
16	5.2.1	Specific gateway implementation
16	5.2.2	Deferred resolution method
16	5.2.3	Half-Gateway management facilities
16	5.3	Trader federation
16	5.4	Domains and domain management
17	5.4.1	Applying the domain experience to construct domains of different types
17	5.5	Monitoring and visualization
17	5.6	Tools
<b>19</b>	<b>6</b>	<b>Work management</b>
19	6.1	Introduction
19	6.2	Dependency between topics
19	6.2.1	Internally to topic
19	6.2.2	Other areas of APM work
19	6.3	Effort estimate
19	6.4	List of deliverables
19	6.4.1	Planning
19	6.4.2	Presentations
19	6.4.3	Documents
20	6.4.4	Code and coding modules/principles
20	6.4.5	Demonstrations



---

# 1 Introduction

---

## 1.1 The business case

---

Agreement was recently reached by the OMG consortium concerning the UNO (Universal Networked Objects) [BROWNELL 94] which paves the way to achieving inter-operability between objects supported by ORBs from different vendors.

In addition, the OMG consortium is now beginning to address the lack of trading facilities in CORBA.

Neither initiatives, however, address the more fundamental questions which arise when different environments are being federated, namely:

- how to connect trading processes of different environments
- how to create and manage both technical and administrative domains. More specifically, the UNO proposal does not cover gateway management issues.

Without tackling those problems, any inter-operability solution will be limited in scope to small scale environments. The aim of this document is to describe the direction in which the work on Federation should proceed in order to deal with these problems.

The suggested work will form the basis for the “management engine” for distributed implementation of information services as suggested in Workpackage B2 [APM.1275 94].

The plan put forward in this document will show the directions in which future work involving trading, interception, boundaries and domain concepts should proceed with a view towards management in large scale distributed systems.

## 1.2 Overview of present work

---

**Interception** is the process which (by creating gateways) can help:

- overcome (technical) boundaries where they prohibit interactions
- create (administrative) boundaries where interactions should be managed, i.e. enabled, disabled, transformed or monitored.

If the interaction between objects in different environments requires some transformation in order to succeed, then the appropriate gateway must be inserted in the interaction path between the two objects prior to them interacting.

This means that at some stage when the interface reference between the objects is passed or when the binding between them is established, the interception process must:

- detect the attempt to establish the link

- recognize the information that has to be acted upon
- mark it for action
- and eventually act on it, to insert the necessary gateway.

The inserted gateway must be able to carry out the necessary transformations on the invocation parameters passing through it, according to the type of boundary crossed. These transformation may in turn include:

- translation of the relevant parameter values
- the interception of any passing interface reference which may cause the creation of new links. In such a case the story of interception may be applied recursively.

To create new links between objects requires **trading** (with or without an explicit trading service). This is the reason for the strong link between interception and trading.

When interface reference cross domain boundaries which require intervention, this must be marked in order to enable the eventual creation of the necessary gateways:

- immediately when interface reference crosses the domain boundary
- deferred until one of the following subsequent stages requests its resolution:
  - gateways
  - traders
  - binders
  - clients or servers

**Trading** relies on information about services and clients and their environments to enable it to detect compatibility. This information can be added by:

- application programmer
- infrastructure
- domain gateways

Used together to delineate the extant of a certain policy by controlling the visibility of services and by controlling the binding and invocation paths, trading and interception provide the basis for constructing **domains**.

---

### 1.3 Towards an Architecture of Federation

---

Note: Federation architecture mentioned in earlier documents: this is a model tying together notions of Authority, Domains, Trading and Interception

The following is a description of the process involving current and future work necessary in order to derive an architecture for federation:

- *gateways* between clients and servers residing in different platforms have been built
- the *model of interception* is being developed and used in enhancing gateways to deal with passing invocation references between different platforms

- *traders* in different platforms are being connected together using the gateways constructed according to the model of interception
- a second look at the issues involved with the *federation of traders* is now necessary
- Combining together interception and trading leads to a review of the concept and implementation of *domains*
- this in turn leads to a re-examination of *trading* within and across domain boundaries, in particular, administrative domains
- which in turn leads to integrating trading and interception and domain management into a model of *domains and their management*
- Finally, examples of how to create domains of different types will influence and be influenced by a description of *scenarios* and a working *classification of differences between systems*.

## 1.4 Federation work topics (Figure 1.1)

### 1.4.1 Work to date

The following topics have been investigated, although some work items still remain to be done (see [APM.1342 95]):

1. **Application level gateways** [APM.1406 95].
2. **Interception and its management** [APM.? 95].

### 1.4.2 Future work

From the above, the following main topics are identified for future work (in decreasing order of priority):

1. **Trader federation (§2)** [APM.1310 95].
2. **Domains and domain management (§3)** [?].
3. **Scenarios** [APM.1095 94]: The work on trading, interception and domains will provide the basic components, guidelines and recipes for constructing generic domains. These will be tested in a variety of scenarios.

Note: Note that the document becomes more fuzzy as we go down the list!

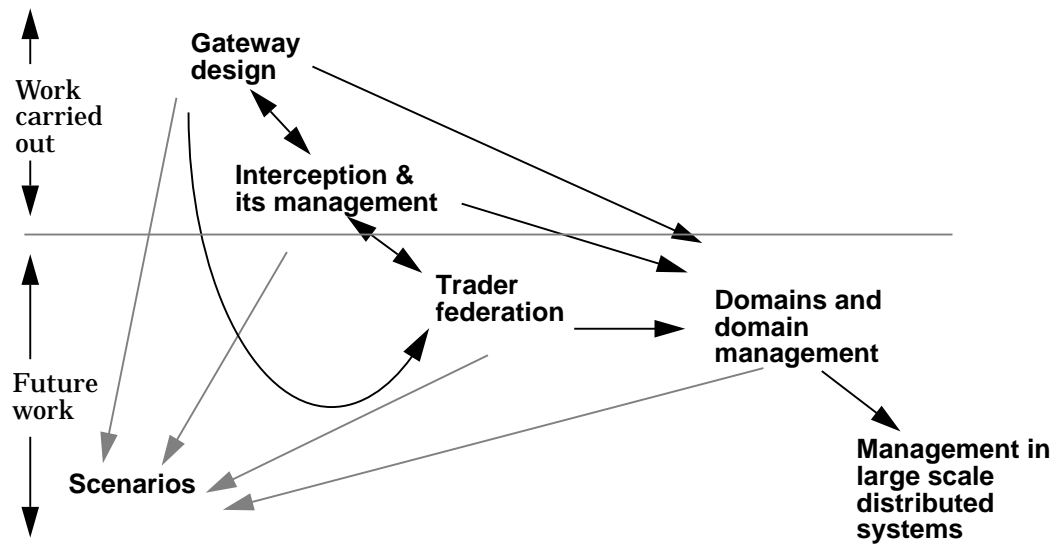
### 1.4.3 Relationship to other Phase III work

Note: Any more information on links?

Trading, interception and domain management work is related to:

- QoS issues such as real-time and dependability: QoS domain management and monitoring
- management of distributed applications such as the WWW
- type conformance: necessary for gateways to deduce whether interface references are being passed between clients and servers (to nth degree)
- meta information: ways of describing differences between domains
- ?

Figure 1.1: Work topics and relation between them



## 1.5 Division of work within areas

The work described in the rest of this document is divided according to the following categories:

- *analysis and modelling* of the issues: derive the principles, guidelines and recipes for an architecture which will enable us to create gateways and domains in different environments
- *design and implementation* of interceptors/gateways, re-design of traders, factories, domain managers (e.g. node managers) in the context of specific platforms such as Orbix, ANSAware
- *prototype* implementation for experimentation and demonstrations
- *input* to standards initiatives: e.g. OMG, and interaction with consortium members.

## 1.6 Background material

The work to be carried out relies on input from:

- ANSA past and current work [APM.1005 93] [APM.1384 94] [APM.1387 94] [APM.1303 95] [APM.1406 95]
- Description of existing platforms: ANSAware [ARM 93], Orbix [IONA 93a], DCE [OSF 92]
- ORB Inter-operability submissions [BROWNELL 94].

## 2 Trader federation

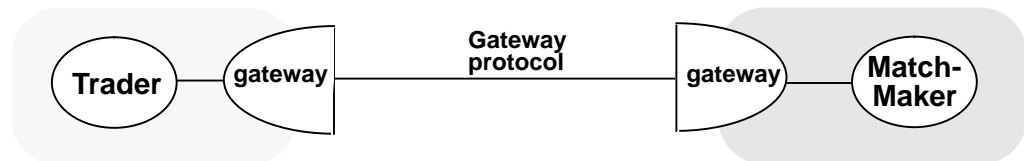
### 2.1 Introduction

The **Objective** of the Trader federation work is to use the experience and concepts developed in the work on gateways and interception to further investigate trading in large scale distributed systems.

Trading can be regarded as the technology necessary in order to find compatible objects, while interception is about enabling, disabling and monitoring the interactions between objects which do not belong to the same domain. Each technology complements and expands the scope of the other in large scale distributed systems.

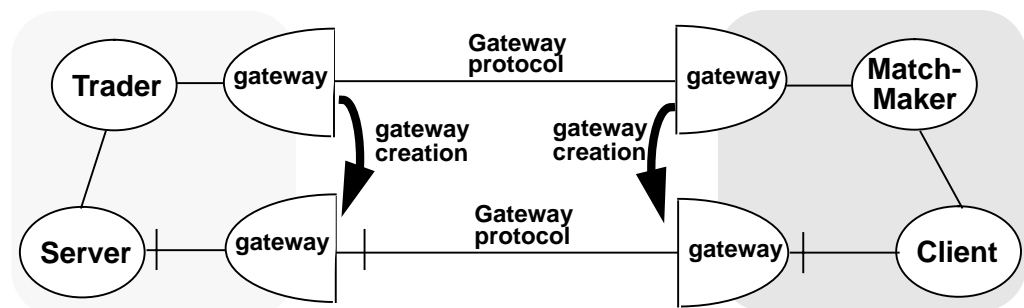
In order to allow federation of different domains, it will be necessary to link their traders together (Figure 2.1). The domain boundaries will have to be bridged between the traders as they would have to between any two clients and servers.

Figure 2.1: Connecting traders in different domains



The information passing between traders concerns interface references and hence potential links between objects in the different domains. Where the boundaries crossed between the traders also concern the potentially new links, some action must be taken to ensure that the appropriate gateways are eventually put into the right place in the newly formed interaction path (Figure 2.2).

Figure 2.2: Setting up links between clients and servers in different domains



### 2.1.1 Background

This work builds on:

- previous experience of building traders in homogeneous environments (e.g. ANSAware Trader [ARM 93] and Orbix Match-Maker [APM.1384 94])
- past work described in APM.1005: “A Model of Trading and Federation” [APM.1005 93] and APM.1387: “Designers’ Introduction to Trading” [APM.1387 94].
- the work on gateways and the interception model
- experience gained from building gateways between the Trader and Match-Maker.

### 2.1.2 Content

The issues which will be investigated:

- linking traders
- minimal trading configuration
- trading domains
- visibility (advertising/search policy) issues
- trading and QoS issues
- trading as an interception process
- domain hopping (indirect paths)
- extension to trading model
- trading and CORBA repositories

## 2.2 Linking traders

---

Further investigation is necessary to determine:

- the different possible ways of connecting traders together:
  - explicitly
  - implicitly through context properties
  - through their database
- the implications of the different ways
- the mechanism and information necessary in traders to enable to different ways of connecting them
- trader linking bootstrapping problem (trading traders)
- trader responsibility and gateway responsibility issues.

## 2.3 Minimal trading configuration

---

What is the minimal/default configuration of traders in a domain: per domain of any type, per node, per organization

---

## 2.4 trading domains

---

When connecting traders the following must be dealt with:

- differences between traders (interface definition, property description semantics and languages, advertizing/search policy description)
- heterogeneous technical domains
- heterogeneous administrative domains.

Should trading domains be defined accordingly and is the notion useful?

---

## 2.5 Visibility (advertizing/search policy) issues

---

Note: This section is a mess!

Trading is about advertizing and searching for advertized services. Some of the information given to the trader concerns bounding the scope of the federation:

When information about services can be advertized in traders of different (administrative) domains, it is possible to control it by:

- traders enforcing domain trader advertizing/search policy (coupled with client/server policy)
- interception carried out at domain boundaries by gateways can:
  - enable/disable/monitor information to pass
  - set up a gateway and pass information about the gateway rather than the service
  - semantics
  - language for specifying trader advertizing/search policy
- what are the rules/implication for one trader advertizing itself in another trader versus a trader advertizing some of the services advertized in it, in another trader

This is concerned with the management of service/client visibility and involves:

- domains will dictate to their traders the in/out visibility they wish to impose
- clients/servers may further constrain the visibility within the confines of the domain they belong to
- domain management may intervene and make special cases
- how to specify visibility
- what are the rules/implication for one trader advertizing itself in another trader versus a trader advertizing some of the services advertized in it, in another trader
- link between domino advertizing and client/server advertize/search request
- Trader advertizing/search policy: how should clients and servers specify the extant/scope of their export/import operations

### 2.5.1 Specifying Advertizing/Searching policy issues

Advertizing/search policy should be specified in as much as possible abstract form. Not in terms of explicit internal structure (making internal representation/structure visibly externally - goes against encapsulation principles) of trader database/repositories

Advertizing in terms of boundaries and their attributes. Classes of Trader advertizing/search policy

- do not leave this domain at all:
  - don't cross any security domains at all
  - don't leave host/node boundaries
  - don't leave administrative physical boundary (e.g. outside department/ building/organization)
  - don't leave this reliability domain
- leave this domain only if you go into a domain with the following attributes:
  - only cross a security boundary if it goes into another of rank=5
  - only give me server if it has reliability rating of XXX
  - search in any domains where remuneration strategy is YYY
  - search in any domains who is banking with the listening bank
- leave this domain only if you let me control/monitor the activity on the links created on a per invocation basis (this is the i don't care where you pass the information to but i am not necessarily going to honour requests).

---

### 2.6 Trading and QoS issues

When combining different QoS domain using communication infrastructure with specific service guarantees, who should check and how the following:

- check end-to-end QoS requirements and guarantees
- take into account (possibly transparent) gateways and their effect on service guarantees.

---

### 2.7 Trading as an interception process

Using the difference between client and server properties to construct gateways. This will require modifications to current trading process (see suggestions in).

This has implications on the type of gateway produced and inserted. It is unlikely that properties which are of importance to client and server can be dealt with in this manner, e.g. security.

- Important boundaries must be dealt with:
  - close as possible to object (e.g. stubs) and immediately not deferred
- not so important issues can be marked:
  - options
  - left to trader to deal with and optimize if possible



## **2.8 Domain hopping (Indirect paths)**

---

The following problems may arise when using gateways between domain boundaries without allowing for examination of the paths created and their optimization:

- crossing multiple domain boundaries (creating a longer path than necessary)
- multiple crossing of the same domain boundaries (Circularity of references problem).

## **2.9 Extension to trading model**

---

### **2.10 Trading and CORBA repositories**

---

A look at the:

- different ways in which implementation repositories and interface repositories are to be used in a trading implementation.
- what is missing in CORBA and ANSAware
- 

Input to the mapping of ANSA and CORBA document.

### **2.11 Design and implementation**

---

The design and implementation work will provide input to a description of the mapping between CORBA and ANSA architecture.

#### **2.11.1 Enhancements to Trader and Match maker**

As result of above investigation it will be possible to enhance the Trader and Match-Maker to implement the different schemes of linking trading processes.

---

## 3 Domains and domain management

---

### 3.1 Introduction

---

**The Objective** of this work is to provide a model and the components necessary to create and maintain technical and administrative domains in a computerized business environment.

The re-visit of the concept and implementation of domains will tie together the experience from the model of interception, building gateways and the investigation into trader federation, into a coherent model.

This would be used to provide an engineering model of how to build and manage domains using trading and interception techniques.

#### 3.1.1 Problems with the concept of domains

The concept of **domains** [SLOMAN 89] is intuitively appealing in any complex system. However, there were a few things missing in previous discussions of domains:

- relationship between domains - cannot be dealt with in isolation
- dynamic links created across domain boundaries (this was particularly lacking in the context of distributed systems)
- how are the boundaries created and maintained - although the notion of a domain manager is conceptually correct in that it indicates the source of policy enforcement, is not enough - the processes, mechanisms and information to carry out the policy were missing. These are now identified as gateways, traders, factories and resource managers.

With the experience gained in the work on interception and trading we are now in an ideal situation to re-examine and make progress with the concept and implementation of domains.

Figure 3.1 shows a model of two domains interacting with one another. It is important to note that the model can be implemented in different ways. For example, gateways can be put in stubs or they can be separate entities shared by many objects. The decision of whether to use one implementation or another will depend on the type of boundary crossed and the importance of the domain.

#### 3.1.2 Background

Sloman's papers:

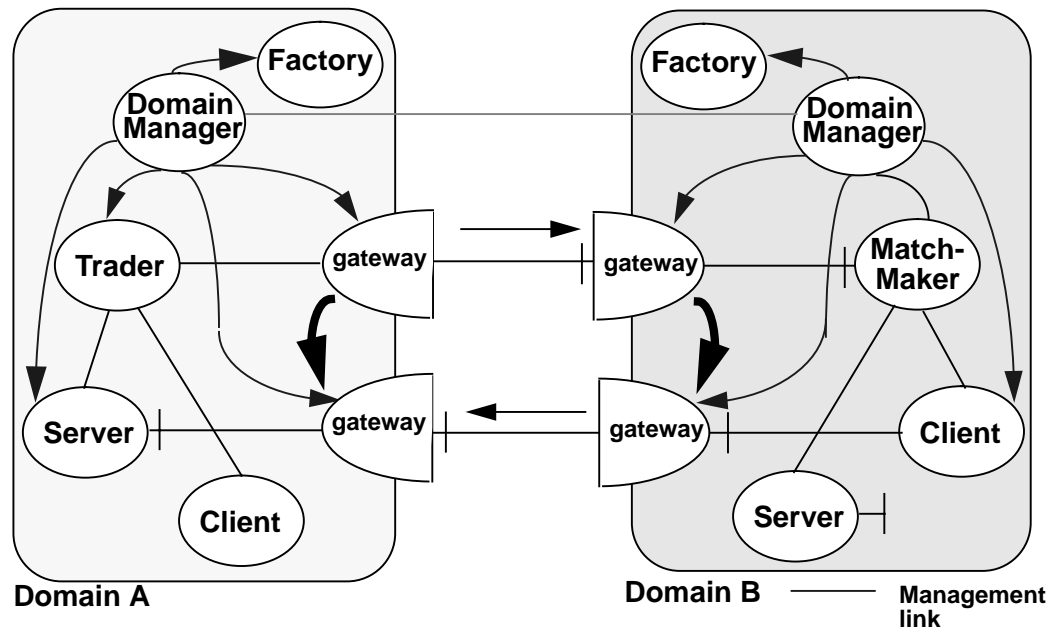
#### 3.1.3 Content

The investigation will look at the following issues:

- heavy-weight model of domains (on a per property basis)
- light-weight model of domains (optimization of the heavy-weight model)

- domain management
- cloning issues
- information needed to construct domains and gateways

Figure 3.1: Domain management configuration



### 3.2 Heavy-weight model of domains

Note: Are domain boundaries dynamic things?

In particular we shall examine:

- heavy weight model of a domain (per property) consisting of the following components and the relations between them:
  - domain manger
  - domain trader
  - domain factory
  - domain gateways
  - domain clients and servers

### 3.3 Light-weight domain model

- how to combine different types of domains and their management
- optimization of the heavy-weight model

combining different domains

trading properties as a means of extending boundaries (resolution strategy):

looking for compatibility

interception: looking for/anticipating differences

---

### **3.4 Domain management**

---

Will be concerned with:

- authority, management and policy issues
- domain membership
- ownership and control of gateways

Where domains will incorporate more than one type of property - the management will have to look after the inter-play between the different boundaries dealt with.

---

### **3.5 Information needed to construct domains and gateways**

---

#### **3.5.1 Design of domain and management**

In view of investigation of domain management (§3) a re-visit the design of the components necessary to create and manage domains will be necessary:

- Traders
- Factories
- Node managers
- Domain managers

Use a scenario (e.g. SimpleBank example) involving two administrative domains and construct the necessary gateways, linking them to local traders, factories and domain managers.

There are two ways in which this can be done:

- modify existing ANSAware components
- build similar Orbix components.

---

### **3.6 Applying domain management to different scenarios**

---

Examples from the scenarios will be used to test the models and components by looking at:

#### **3.6.1 security domain**

Note: High on priority list!

#### **3.6.2 remuneration domain**

#### **3.6.3 Organizational domain**

#### **3.6.4 Design and implementation**

Experiment with the implementation of the above domains.

## 4 Miscellaneous issues

---

### 4.1 Introduction

---

***The Objective:*** to investigate issues not completely covered in the previous chapters and where necessary to concentrate on them as topics in their own right.

The following provide an initial tentative list of issues:

- gateway management: gateway maintenance and destruction issues
- tools for the construction of gateways and domains
- garbage collection
- gateways, transparency and QoS
- circularity of reference
- relocation
- migration
- garbage collection
- management of distributed applications: WWW

---

## 5 Prototype implementation

---

### 5.1 Introduction

---

**The Objective:** to verify the models developed and to provide practical input to the investigation. The prototypes will be used as demonstrators. Where appropriate, the resulting code should be made available.

### 5.2 Suggested prototypes

---

It is expected that more detail of prototypes will become available when the work progresses.

#### 5.2.1 Specific gateway implementation

A number of experiments are suggested:

- Echo client-server: ANSAware-Orbix (available) [APM.1303]
- ANSAware Trader - Orbix Match Maker:
  - immediate resolution method: gateway factory issues

#### 5.2.2 Deferred resolution method

A number of experiments are suggested:

- using proxyExport() operation
- ANSAware Binder and Ifref re-design: deferred resolution method
- WWW - CORBA ?

#### 5.2.3 Half-Gateway management facilities

A number of experiments are suggested:

- gateway management protocol:
  - between half-gateway
  - with clients/servers
- monitoring activation for reporting about traffic
- producing generic gateway components.

### 5.3 Trader federation

---

### 5.4 Domains and domain management

---

#### **5.4.1 Applying the domain experience to construct domains of different types**

### **5.5 Monitoring and visualization**

---

The appropriate monitoring facilities are provided to enable visualization of the prototypes. These form part of the generic components of gateways.

### **5.6 Tools**

---

Automation of the process of gateway construction:

- generating skeleton gateways
- generating transformations from property descriptions.





---

## 6 Work management

---

---

### 6.1 Introduction

---

---

### 6.2 Dependency between topics

---

#### 6.2.1 Internally to topic

#### 6.2.2 Other areas of APM work

Links to other APM work:

- CORBA & AW to WWW gateways
- Metadata for describing properties of WWW services
- AST for types

---

### 6.3 Effort estimate

---

---

### 6.4 List of deliverables

---

#### 6.4.1 Planning

- APM.1404: "Federation Future Work Plan" [APM.1404 95]

#### 6.4.2 Presentations

- TC presentations:
  - APM.1249: "TC presentation June 94 Interception" [APM.1404 95]
  - APM.1371: "Interception Model slides" [APM.1406 95]
  - APM.1413: "Using the Interception Model" [APM.1406 95]

#### 6.4.3 Documents

- documents describing interception model:
  - APM.14???: "Passing Interface references Across Domain Boundaries" [APM.1404 95]
  - ?
- documents describing principles of gateway design, with recommendations, prototyping results, possibly even interfaces:
  - APM.1406: "Introduction to Application-Level Gateways" [APM.1406 95]

- **Trader federation:**
  - APM.1310: “A Gateway between Traders” [APM.1310 95]
- documents describing domain management model
- documents describing prototyping
  - APM.1303:”ORBIX-ANSAware Gateway Design & Implementation” [APM.1303 95]
  - APM.1299:”Gateway visualization” [APM.1275 94]
- prototypes of specific problems [?]
- Mapping CORBA concepts to ANSA architecture [?].

#### **6.4.4 Code and coding modules/principles**

- framework: general components evolved by prototyping. Software for sponsors to experiment with
- tools for generating mapping code (?)

#### **6.4.5 Demonstrations**

- visualization of simple Echo client-server
- visualization of Trader to Match-Maker gateway and immediate resolution
- visualization of deferred resolution:
  - proxyExport
  - binding resolution

---

## References

---

[ANSA 91]

ANSA: A Systems Designer's Introduction to the Architecture, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., April 1991.

[APM.1001 93]

Rees, R.T.O., "The ANSA Computational Model", APM.1001, Architecture Projects Management, APM Ltd., Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1993.

[APM.1003 93]

van der Linden, R., "The ANSA Naming Model", APM.1003, APM Ltd, Poseidon House, Castle Park, Cambridge CB3 0RD U.K., February 1993.

[APM.1005 93]

Deschrevel, J-P., "The ANSA Model for Trading and Federation", APM.1005, Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1993.

[APM.1091 93]

Beasley, M.D.R., "Advanced Trading", APM.1093, Architecture Projects Management, Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1993.

[APM.1095 94]

Gomer, T. Editor, "Micro-Scenarios for Federation", APM.1095, Architecture Projects Management, Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1110 94]

Beasley, M.D.R., "New Trader - Analysis and Design", APM.1110, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1193 94]

Beasley, Cameron, Gray, Hoffner, Linden, Thomas, "Federation Manifesto", APM.1100, Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1275 94]

"1994-1996 ANSA Workplan", APM.1275, Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1229 94]

Cameron, E.J. & Linden, R.V.D., "Information Moedl for Federatiion", APM.1129, Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1314 94]

Otway, D. J., "The ANSA Binding Model", APM.1314, Architecture Projects Management, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1299 94]

Hoffner, Y., "Gateway Visualization", APM.1299, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1303 95]

Hoffner, Y. & Crawford, B., "ORBIX-ANSAware Gateway Design and Implementation", APM.1303, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[APM.1310 95]

Hoffner, Y., "A Gateway Between Traders", APM 1310, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1342 95]

"An Overview of the Description and Distribution of Services", APM 1342, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[APM.1371 94]

Hoffner, Y., "Interception (TC Decmber 94 slides)", APM.1371, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1384 94]

Girling, G. & Beasley, M., "The Property Repository", APM 1384, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1387 94]

Hoffner, Y. "A Designers' Introduction to Trading", APM.1387, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1994.

[APM.1404 95]

Hoffner, Y. & Crawford, B., "Federation Work Plan", APM.1404, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[APM.1406 95]

Hoffner, Y., "Introduction to Application-Level Gateways", APM.1406, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[APM.1413 94]

Hoffner, Y., "Using the Interception Model (TC March 95 slides)", APM.1413, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[APM.? 95]

Hoffner, Y., "Passing Interface References Across Boundaries", APM.14, APM Ltd., Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1995.

[ARM 93]

"The ANSAware 4.1 manual set", Architecture Projects Management, Poseidon House, Castle Park, Cambridge CB3 0RD U.K., 1993.

[BROWNELL 94]

Brownell, D. (SunSoft), "Universal Networked Objects", Revised ORB 2.0 Interoperability Submission by BNR, Expersoft, IBM, ICL, IONA Technologies, and SunSoft. 1994.

[IONA 93a]

**Orbix: Programmer's Guide**, Iona Technologies Ltd., Dublin, Republic of Ireland, 1993.

[IONA 93b]

**Orbix: Advanced Programmer's Guide**, Iona Technologies Ltd., Dublin, Republic of Ireland, 1993.

[ODP 93]

"Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive Model", ISO/IEC CD 10746-3 (Committee Draft), ISO/IEC JTC1/SC21/WG7, June 1993.

[OMG 92]

"The Common Object Request Broker: Architecture and Specification", Document Number 91.12.1, Object Management Group and X/Open, 1992.

[OSF 92]

OSF, DCE Application Development Guide, Open Software Foundation, 11 Cambridge Centre, Cambridge, MA 02142, USA, 1992.

[SLOMAN 89]

Sloman, M. "Domains...",

