Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

An Overview of the Information Services Framework

Owen Rees, Nigel Edwards

Abstract

The provision and use of dependable information services in wide area networking presents business opportunities, and both business and technical challenges.

This document presents a strategy for identifying and answering the technical challenges, and for developing the technical components that will help in the exploitation of the business opportunities.

The strategy is presented as a collection of tasks in the context of the ANSA Phase III plan. Some of the tasks focus on the development and explanation of necessary concepts. Other tasks focus on the prototyping and demonstration of components.

The intended audience is ANSA team members and those who wish to a get a snapshot of the work. This is a living document: parts of it are likely to be under construction, other parts are likely to be changed. Comments and feedback are always welcome.

APM.1306.00.06 **Draft** 9th November 1994

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

Distribution:
Supersedes:
Superseded by:

An Overview of the Information Services Framework

An Overview of the Information Services Framework

Owen Rees, Nigel Edwards

APM.1306.00.06

9th November 1994

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	(01223) 515010
INTERNATIONAL	+44 1223 515010
FAX	+44 1223 359779
E-MAIL	apm@ansa.co.uk

Contents

1 Introduction

The provision and use of dependable information services in wide area networking presents business opportunities, and both business and technical challenges. This document presents a strategy for identifying and answering the technical challenges, and for developing the technical components that will help in the exploitation of the business opportunities.

The intended audience is ANSA team members and those who wish to a get a snapshot of the work. This is a living document: parts of it are likely to be under construction, other parts are likely to be changed. Comments and feedback are always welcome.

1.1 Stakeholder objectives

The objectives of the major participants in the business of information service provision are used to derive the technical challenges.

1.1.1 The information service provider's objective

Seamless provision of service from heterogeneous resources.

Seamless provision of service to a heterogeneous user community.

Managing evolving services.

Starting from existing resources: Internet based (WWW, Gopher, WAIS/ Z39.50, FTP, etc.), others (X.500, commercial DBMS, etc.).

Evolve towards heterogeneous resource capability.

Support new facilities not handled by existing systems: live audio/video, anything with Real Time requirements.

Attract users; the provider's resource is one in a multitude, the provider must make sure it is visible to potential customers.

Support user's search requirement.

Making the service sufficiently reliable to satisfy customers.

1.1.2 The information service user's objective

Finding relevant resources: meta-information about resources, technical (location, retrieval protocol, data format, etc.) and content (subject area, quality, impartiality, how up-to-date in both content and presentation).

Finding information in huge information space: human operated browser, local search engine, access to search service, "worm" searcher.

Finding relevant resources, without excess spend on search.

Notification of relevant item: "I see you are interested in cellular automata, there is a new archive of patterns at..."

Notification of technical change: “archive formerly at harbor.ecn.purdue.edu is now at ftp.aud.alcatel.com”.

Dependable use of service (?)

1.1.3 The information service agent’s objective

An agent is both user and provider, so has all the objectives described above.

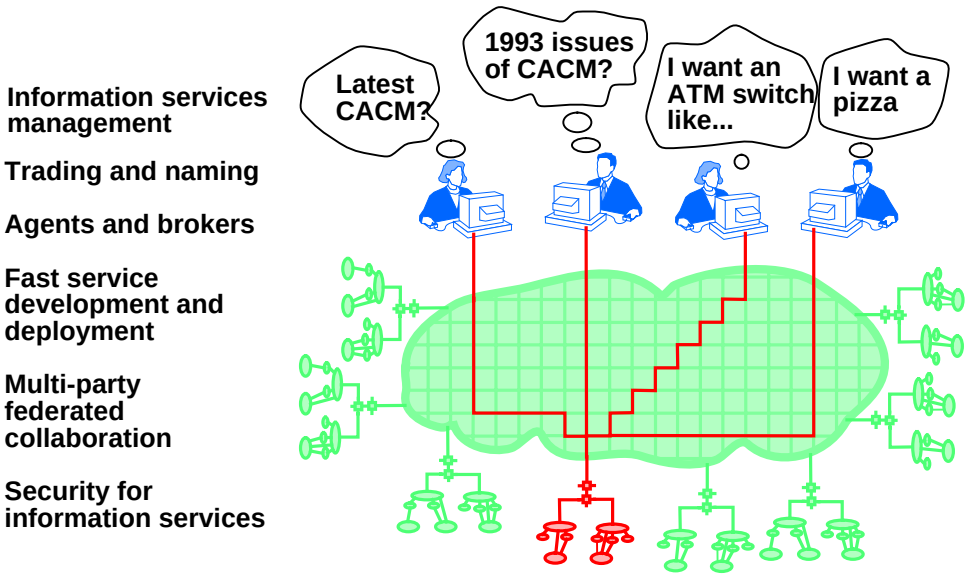
Must be able to act on behalf of user - authentication, remuneration issues.

Dependable service despite occasional/partial failure of provider.

1.2 Achieving the objectives

The theme of this work is to use distributed object oriented infrastructure (e.g. CORBA) to provide the virtual operating system for the global network. The results should say what this operating system should provide. Figure 1.1 illustrates the technical challenges that must be answered to achieve the objectives.

Figure 1.1: Technical challenges



1.2.1 Service provider

Needs tools for rapid deployment and management of services; use Distributed Object Oriented Platforms (e.g. CORBA or CORBA-like) §2.1.2 *Workpackage: B2*

Needs infrastructure that can support time guarantees in order to deploy live audio/video or other R/T service.

Support user’s search “worm” - script interpreter; e.g. TeleScript, safe-Tcl §2.1.3 *Workpackage: B3*

Common model of information resources to support seamless provision.

Federated naming model for resources; specifications for name interceptors.

1.2.2 User

Browsing tool, delivers many resources in common style.

Search tool; local follows references according to relevance and cost criteria, interface to provider's search tool (e.g. like WAIS), "search worm" constructor and reporting mechanism - needs script technology. Seamless integration, at least of how user describes area of interest. §2.1.3 *Workpackage: B3*.

1.2.3 Agent

Smart forwarding/limiting of searches? §2.1.3 *Workpackage: B3*

1.3 Information Campaign

How are the results of this work to be made visible?

1.3.1 Technology transfer to sponsors

Some possibilities:

Run a mailing list on which we send out notifications of achievements.

Provide remote access to our demonstrations (e.g. via WWW).

Provide code of demonstrations.

Make documents available.

1.3.2 Awareness campaign for non-sponsors

Be visible as "good chaps" on significant mailing lists and news groups (uri@bunyip.com, www-talk?, other?)

Overview of work available on WWW server.

Publish briefing notes.

1.3.3 Engage in related projects or activities

E.g. Volunteer to be W3O host in UK

2 Workplan: B-series - Services

This chapter is based on the workpackages set out in section 5.3 of APM.1275.02. Underlined items are links between the workpackages, and links to more detailed items later in the document.

2.1 B-series Workpackages

2.1.1 Workpackage: B1

Title: Object wrapped information services

Objective:

a) A CORBA wrapper for the HTTP protocol and associated "resources" (named by HTTP URLs) enabling CORBA-based implementation of WWW information services.§3.3.3.5 *CORBA/WWW interworking*, §4.2.8 *General CORBA/WWW interworking*

b) Use of trading and advanced naming concepts to extend capability of HTTP URLs to embedded CORBA based applications and mobile resources.§3.3.3.3 *Federation between Internet/URI and CORBA*, §4.2.3 *Representing CORBA Service IDLs (the trader) in the WWW naming space*

c) An infrastructure to support work on Management Engines, and Scripts and Agents. Used by B2 and B3

Method: This provides two ingredients for later workpackages: the information resources needed to drive the scenario in which the management engine is to be tested and an initial tcl/tk based infrastructure for the Management Engine.

Application animations based on WWW HTTP access and use of distributed object technology will require the application and demonstration of federation principles. The location of appropriate resources will require the application and demonstration of trading and naming principles.

Continuity: This work builds from 1993-4 plan deliverables D3: dependability engineering model and F4: interception.

Results: Software and prototype.

Timetable: Start: Now

End: 3/95

2.1.2 Workpackage: B2

Note: This workpackage description is very detailed, and may need to be refined when the results of other work become available (e.g. workpackage B1 and the distributed control workpackages).

Title: Management Engine

Objective: Develop architecture and concepts for

a) a “Management Engine” for dependability, providing the basic services to make the commercial information services highly reliable and available and also supporting in-service upgrades.

b) a “Management Engine” for distributed implementation of information services, e.g. facilities for cache management. This could be prototyped using Orbix to build a CORBA based cache manager for CORBA HTML files. §4.2.7 *Author/ resource management*

Both need to provide end to end services in a wide area network addressing such issues as:

a) what happens to customers if the local area network on which the service is sited becomes unavailable?

b) how to switch over to an alternative service?

c) how is such wide area redundancy managed?

In the longer term the management engine could be extended to include facilities for the following:

a) monitoring node load, thus enabling load balancing to achieve or maintain performance and QoS guarantees

b) high-level resource coordination to deliver end-to-end resource control in a wide area broadband network: making sure the service is as close as possible to where it is needed (by caching), making sure the broadband and local area networks collaborate to deliver the required end to end channel capacity.

Method: The work first builds on the infrastructure provided by the previous workpackage, to allow experiments with failure models and dependability mechanisms.

Secondly it build on results from the distributed control workpackages. Timed communications and rigorous quality of service guarantees make the detection of failures a more tangible task: it is not possible to distinguish between a failed service and a slow running service in the absence of bounded time communication. The binding architecture and prototype will allow time-outs (bounds on communication) to be fixed at bind time.

Continuity: There is a strong link with the current work on QoS architecture and engineering. Many of the topics were identified as important in a recent workshop, and documented in [APM.1220] and [APM.1233].

The work builds on 1993-4 plan deliverables D2: dependability management model and D3: dependability engineering model.

Results: Software and prototypes

Timetable: Start: 2/95

End: 10/95

2.1.3 Workpackage: B3

Title: Scripts and agents

Objective: The work will cover:

a) attribute and set based naming schemes for information services §3.3.2.2 *Meta information model for URCs*.

b) agents and information brokering.

c) script technology to coordinate multi-party business activities in a wide-area network dependably.

Method: This area grows the object wrapped information services to provide support for potentially mobile scripts and agents supporting multi-party information services and business processes.

Continuity: Builds on 1993-4 plan deliverables A8: Automated Transparencies for Dependability, D2: dependability programming model, and D3: dependability engineering model.

Results: Reports and examples

Timetable: Start: 12/94

End: 12/95

2.2 Secondary Objectives

Initial focus is WWW (HTTP/HTML).

Others Internet information resources exists, and the work could be extended to apply to them. The other resources include: (anonymous) FTP, gopher, WAIS, Z39.50

The concepts and management strategy could be applied to “native Internet” resources as well as being accessible via a CORBA interceptor. Current resource and infrastructure management in WWW is ad-hoc.

3 Concepts

This chapter describes the concepts which need establishing for the Information Services Framework (ISF). This framework will encompass systems with which we wish to interwork, and especially the Internet which already has vast information resources available. The components described in §4 *Components* will use the concepts described in this chapter, to form a coherent ISF.

3.1 Starting points

The following items are available as inputs to this work.

- ANSA Naming model
- other ANSA documents?
- IETF draft Uniform Resource Locator specification
- IETF draft Requirements for Uniform Resource Names
- discussions of the IETF Uniform Resource Identifiers working groups which take place on the mailing list uri@bunyip.com

3.2 Deliverables

The following kinds of output are expected from this work

- Report(s) describing the concepts
- Contributions to development of URI (esp. URN, URC specs)
- Application of concepts to components

The concepts can be considered sufficiently complete and sufficiently well explained when they can be used by people not directly involved in their development.

Potential users of these concepts, and the audiences for the reports are:

- Other ANSA team members
- Sponsors
- General IT community
- Standards organisations

3.3 Concepts and issues to be developed

3.3.1 Information model of URI area

3.3.1.1 Information model of URI elements

Build an information model describing the relationships between URIs, URLs, URNs and URCs. Currently there is much confusion about these concepts in the community.

3.3.2 Meta-Information

The IETF URI working group are developing a meta-information model for Internet resources. We need to participate in order to ensure that their proposals are compatible with our interworking objectives.

3.3.2.1 Should Meta Information be URNs, URCs or URLs?

This kind of meta information is information returned (in server HTTP response headers) about which other objects a hypertext object has links to; it should not be confused with the meta information model required for a trader to allow humans to evaluate the worth of data - a longer term aim.

The meta information could consist simply of the URNs of linked objects. A client then has the option of using the URL in the object, or using a service to resolve the URN. (It seems likely that the URN would only ever be used if the URL didn't work, unless the trader could resolve URN to URLs very quickly.)

If the meta information consisted of URCs (or parts of URCs e.g. URLs) of linked objects, browsers could examine URCs in the meta information to choose the most appropriate instance of a resource when a user selects a hyperlink.

Servers may or may not return meta information. If meta information is returned clients can ignore it. (All browsers are supposed to ignore headers which they do not understand.)

3.3.2.2 Meta information model for URCs.

Meta information allows humans and agents to evaluate the worth of data in a a resource: i.e. find the golden nuggets amongst the huge amount of muck (most people's golden nuggets are other's muck).

Web robots retrieve information, using it to create meta information which they store in the trader. Extend trading to allow trading on meta information (as well as URNs, URCs and URLs).

3.3.3 Naming and location issues

3.3.3.1 Resolution

There are several kinds of name resolution to be considered; in particular:

- URN -> URL lookup
- URN -> URC lookup
- URL -> URN/URC lookup

Trader could do URN -> URL lookup and URN -> URC lookup. A URL -> URN/URC lookup may well fail because it may not be possible to determine which trading domain to search from a URL. E.g. suppose ANSA document

instances are stored at HP. The URLs to access these will be something like: http://www.hp.com/projects/labs/bristol/ansa/**. In general there is no way to tell from this URL that the trading request (for a URL -> URN lookup) should be directed to the trader for the domain “ansa.co.uk”.

3.3.3.2 Nature of links

- Should links in objects be URLs or URNs?

A link is an html anchor or hypertext link to another object. If links are URNs you always have to go to a (trading) service to resolve the name. Existing browsers cannot cope with URNs. So it seems as though links in html will continue to be URLs; URNs will appear as headers (meta-information) returned by the server to browsers.

3.3.3.3 Federation between Internet/URI and CORBA

The descriptions above have focused on Internet/URI issues. In order to interoperate between WWW and CORBA, we need a similar understanding of the concepts in a CORBA context, and then an explanation of how the concepts in the two worlds are related.

- What is a resource in WWW, in CORBA; how are these related?
- What is a name in WWW, in CORBA; how are these related?
- How is a CORBA resource named from WWW?
- How is a WWW resource named from CORBA?

3.3.3.4 Resource References

What options are available for passing resource references in WWW? What is feasible within the technology? What is acceptable to the community?

An idea to explore; adapt the interface reference idea to WWW, and prototype the result:

- URN plus multiple URL, like interface references
- Handle RR, URN, URL
- Server sends RR in HTTP headers
- Browser keeps RR, when URL selected, look up and use RR
- NCSA Mosaic and CERN httpd for prototyping
- Relocator: refresh stale RR (passive - delivers new information when requested)

3.3.3.5 CORBA/WWW interworking

What concepts are needed to support §4.2.8 *General CORBA/WWW interworking*?

Ideas to explore:

- generic interceptor - where does it get mapping rules?
- CORBA IDL as input to HTML form generator §4.2.2 *CORBA IDL and WWW Scripts*
- dynamic or static generator - what options are feasible?

3.3.4 Author/resource management

Concepts to support prototyping §4.2.7 *Author/resource management*

3.3.4.1 Authoring tool issues

What does an author write to identify destination of a link?

- Current WWW requires that author writes URL.
- Tool to check existence?
- How are relative links handled?
- Set of documents with cyclic reference chains (e.g. previous/next)

Destinations with multiple instances; suppose §3.3.3.4 *Resource References* in use.

- Author uses URN/URL, tool finds RR?
- When? At moment of selection? When document completed?

Install document into server

- HTTP has a PUT method - is it used?
- Common practice today is ad hoc - write into file - URL/URN not used

3.3.4.2 Maintenance tools

Track resources - maintain integrity of links - active RR refresh.

- Test links from here?
- Inform others who have links to here?

Consistency of multiple instances

- Where is the membership list?
- Urgency of update - is it the same for all instances?

Part of this task is to identify existing tools, such as those supplied by EIT.

4 Components

This chapter describes the components to be prototyped for the Information Services Framework.

4.1 Deliverables

The following items are available as inputs to this work.

- Concepts as described in §3
- CORBA platform (assume Orbix for now)
- Public domain CORBA IDL compiler

The following kinds of output are expected from this work

- Prototypes

4.2 Experiments to develop components

In progress:

- §4.2.1 *Trader*
- §4.2.2 *CORBA IDL and WWW Scripts*
- §4.2.3 *Representing CORBA Service IDLs (the trader) in the WWW naming space*

Tentative; needing development of concepts

- §4.2.5 *Resource reference implementation*
- §4.2.6 *Relocator*
- §4.2.7 *Author/resource management*
- §4.2.8 *General CORBA/WWW interworking*

4.2.1 Trader

4.2.1.1 Access to trader from WWW

Trading: make available to non-CORBA (e.g. WWW) browser *§2.1.1/b*

Build an infrastructure which allows the current prototype to be invoked from WWW. The objective of this is to allow trading on URLs. Will need to build simple scripts to allow posting of offers in traders and also searching of offers in traders.

Note: Current status: an infrastructure has been built which makes many of the enhanced trader's interfaces available through WWW, showing that trading is feasible in WWW. The work is now awaiting the results of §3.3.1 (URI Information model) and §4.2.2 which (hopefully) will allow much of the interception code to be generated automatically.

4.2.1.2 *Trader as URC repository*

Adapt the trader to support the information model of URI built in §3.3.1.

The goal: make the trading service a prototype URC repository for WWW

Even if this is not widely adopted by the WWW community we should learn a lot about wide area federated trading - lessons which should be applicable to other problem areas.

4.2.1.3 *Prototype federated trading of web resources.*

Federations are partitioned by URN domains. There is potential for many levels of federation: as well as federating with other traders a trader might federate with other URC services (e.g. some have proposed using the whois++ service for URCs).

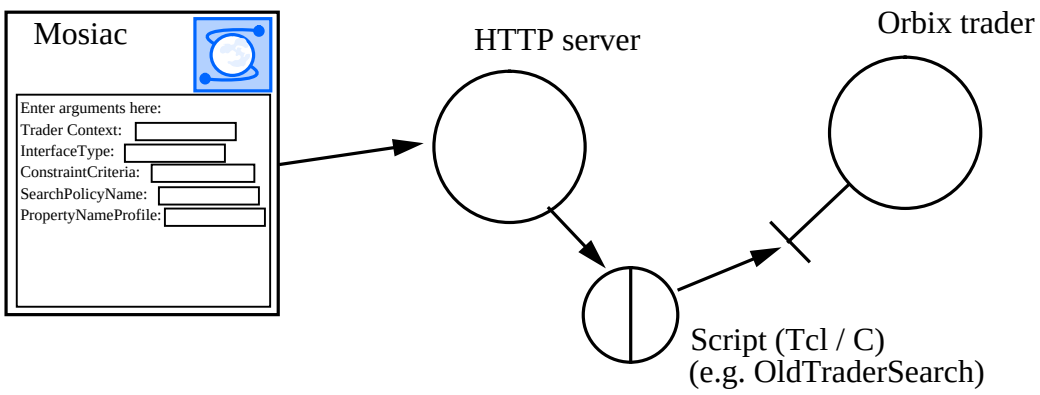
4.2.1.4 *Role of trader in authoring*

Trader should verify that URLs exist when they are installed. Periodically ping them, or rely on management service to inform the server of the change (URC or group manager)?

4.2.2 CORBA IDL and WWW Scripts

The experience of making the trader accessible via WWW was very similar to trying to write a CORBA application without a stub compiler. A script needs to be written for each operation supported by the trader. The function of the script is to unmarshal the arguments (from standard input), invoke the appropriate trader operation and marshal the results (into standard input). The scripts are invoked by a WWW server. Human users select which operation to invoke and provide arguments by using HTML forms. We had to be very careful to keep the form consistent with the script (i.e. right number of arguments etc.). The architecture is shown in figure 4.1[1]

Figure 4.1: Trader WWW Architecture



It seems likely that a stub compiler could automate much of this. From an interface definition a stub compiler could:

1. The figure shows NCSA Mosaic as the browser – there are many other browsers that could be used in this case. Source code for the browser will be required for some later experiments; NCSA Mosaic is the most sophisticated browser available in source form.

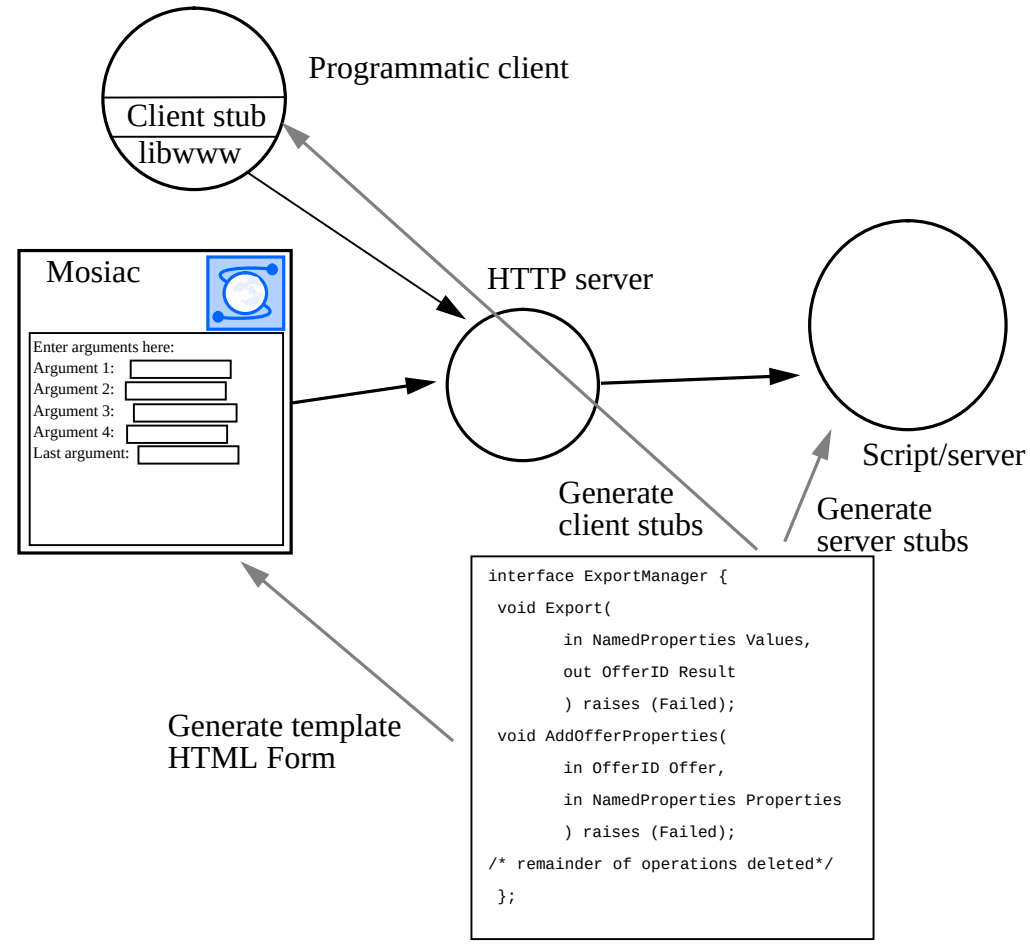
- Generate a template HTML form (adding explanatory text and adjusting input field size would be optional).
- Generate the code to unmarshal the arguments of the script when they are received from httpd (analogous to server stubs).

This could be taken a little further. The stub compiler could also:

- Generate code to marshal results (programmers wouldn't have to worry about explicitly writing the results in stdout - it would be done automatically).
- Generate client stubs so that people could write programs to invoke scripts.

The full architecture is shown in figure 4.2. The script is made to look like a CORBA service: it may or may not invoke a third party (e.g. trader). Potentially, the HTML form document could be generated on the fly from the IDL.

Figure 4.2: Script generation



Starting from a CORBA IDL compiler (public domain front ends exist), the end result may well be to add CORBA functionality to WWW. It would mean that CORBA/WWW interworking becomes the same problem as CORBA/CORBA interworking.

The nature of the underlying WWW technology suggests that commercial CORBA offerings such as Orbix will have much to offer paying customers in

terms of robustness and performance (the latter is likely to be an order of magnitude better). One of the benefits of this work will be to help make the demand for CORBA technology ubiquitous.

Note: Status: We are just starting this.

4.2.3 Representing CORBA Service IDLs (the trader) in the WWW naming space

There are several options for structuring a CORBA service within WWW. One would be to build a single script to access the service; this is consistent with the idea of automatically generated server stubs discussed in §4.2.2. Script parameters would select the interface and operation as well as contain parameters for that operation.

An alternative would be to construct the script space to reflect the interface structure of the service. In the file space all interfaces would have their own directory and each operation would be a separate script, e.g.:

```
http://www.ansa.co.uk/trader/OldTrader.idl
http://www.ansa.co.uk/trader/OldTrader/search
http://www.ansa.co.uk/trader/SubProfileManager.idl
http://www.ansa.co.uk/trader/SubProfileManager/SetProperty
```

4.2.4 Publishing document

Build scripts to allow publishing of information: create multiple instances and export the URC to the trader.

A detailed scenario.

1. Parse (HTML) source to detect links.
2. Use trader to lookup URCs of linked objects (should also check they exist?).
3. Use trader to lookup reference of candidate servers at which to store objects. (Note this implies that the trader stores interfaces to HTTP servers as well as URCs, URNs and URLs.)
4. Construct URN for new resource - how does this get done? Domain specific URN generator? Could the trader do it?
5. At each candidate server post object (server can refuse to allow post or except post and return a URL for the object)
6. Construct URC of new resource (will consist of URN + URLs of each instance returned by servers in response to posting request + information about linked objects (what information about linked objects?)
7. Construct meta information for new resource (=URC) and use the HTTP “link” method to post this to the servers which are storing instances of that resource. Servers can ignore the meta information or store it to return it as part of the header information to be returned when they serve the object to a client.

4.2.5 Resource reference implementation

URN plus multiple URL, like interface ref

Handle RR, URN, URL

Server sends RR in HTTP headers

Browser keeps RR, when URL selected, look up and use RR

NCSA Mosaic and CERN httpd for prototyping

4.2.6 Relocator

Relocator: refresh stale RR (passive - delivers new info when requested)

4.2.7 Author/resource management

Track resources - maintain integrity of links - active RR refresh

Install document into server

Author uses URN/URL, tool finds RR on installation.

Consistency of multiple instances

Prototype scripts which manage availability of resource instances: maintain integrity of offers in the trader when resources fail and are updated. (Group Managers or URC Managers.)

4.2.8 General CORBA/WWW interworking

Retrieve resources across boundary

build gateway that translates RR<->URL and makes proxy invocation

5 Issues

5.1 Data Protection

Information providers will be subject to the Data Protection Act in the UK, and any equivalent legislation that may be in force elsewhere. Does this have any impact on the architecture?

Issue raised: 21st October 1994 at ANSA project meeting.

5.2 Relationship with dependability

The position of dependability in the plan needs to be clarified.

Issue raised: 21st October 1994 at ANSA project meeting.

The current position is that dependability will be addressed more specifically when the pieces of infrastructure identified in this document have been built. In addition we will need a set of applications which need to be made dependable and (concrete) requirements. WWW looks like it will be a fruitful source of applications.

Note that one of the reasons for building the infrastructure described here is to improve the dependability of current access technology. For example, if a browser fails to retrieve the resource instance identified by a particular URL, at the moment there is no way to switch over to an alternative instance of that resource.

5.3 Acronyms

This document contains many acronyms that are neither expanded nor explained. This makes it hard to follow for those not directly involved in the work.

Issue raised: 21st October 1994 at ANSA project meeting.

5.4 Generality

The plan focuses on WWW (URI). Will the results be applicable in other contexts?

Issue raised: 21st October 1994 at ANSA project meeting.

5.5 Other initiatives

Who else is working in this area? How does their work relate to ours? Is there a danger that we will be left behind, and our work be rendered irrelevant by other initiatives?

Issue raised: 21st October 1994 at ANSA project meeting.

There is group actively researching a general purpose directory service for the internet called “whois++”. This service has much in common with trading. However, it has some neat features which trading lacks; conversely trading has some neat features which whois++ lacks. At the moment we do not have the effort to explore this relationship. Relevant documents on whois++ are: [FALTSTROM 94], [GARGANO 94] and [WEIDER 94].

5.6 Trading for Commercial Purposes

Trading for commercial services needs to be “fair”: how do you ensure that a trader is not biased towards one particular vendor, returning that vendors offers more frequently than its competitors.We do not have the effort to look at this.

5.7 Trading and Information Indexing

What is the relationship between trading and indexing services such as Harvest? (Harvest includes technology “worm/robot” technology which automatically creates indexes of home pages in WWW. Currently the robot has discovered over 10,000 home pages and created a single searchable index for them.)

5.8 Cutting HTTP as a Protocol into CORBA

If HTTP we a protocol available to CORBA (or e.g. ANSAware) CORBA clients could access WWW services. This is perhaps not so interesting for the retrieval of web pages. However, it would allow CORBA clients to access other servers behind HTTPD servers accessed via CGI (Common Gateway Interface) scripts.

This would also allow WWW browsers (e.g. Mosaic) to invoke CORBA services (presumably the CORBA service would have to look like a CGI script to the browser).

Given that the work described in §4.2.2 *CORBA IDL and WWW Scripts* deals with describing WWW services using IDL. A more general approach might be to use whatever technology is developed to solve CORBA/CORBA interoperability.

6 Diary

6.1 ANSA core team review 21 October 1994

Version 00.04 discussed at a project meeting. Issues raised were added to §5 of version 00.05.

6.2 Trader WWW Demonstration 21, 24, 25, 26 October 1994

Trader accessed and demonstrated through WWW to various Hewlett-Packard entities in the USA.

6.3 CORBA IDL Compiler Retrieved and Built 4 November 1994

Sun's public domain front end CORBA IDL compiler was retrieved and built on HPUX (non-trivial). No we need to figure out how to build a back-end for WWW!

6.4 Trader WWW Demonstration to HP Bristol 21 November 1994

Trader demonstration scheduled for John Taylor at HP Bristol as part of internal review.

6.5 Jon Crowcroft 25 November 1994

Jon Crowcroft of UCL visiting to discuss all aspects of ANSA (including this work).

6.6 Technical Committee 7,8 December 1994

What do we need to do for this?

6.7 Joe Sventek 16 December 1994

References

[FALTSTROM 94]

"How to interact with a Whois++ mesh", P. Faltstrom, R. Schoultz, C. Weider, Internet draft 24 July 1994, <draft-ietf-wnils-whois-mesh-00.txt>

[GARGANO 94]

"Whois and Network information Lookup Services Whois++", Joan Gargano, Ken Weiss, Internet Draft June 26 1994 <draft-ietf-wnils-whois-lookup-01.txt>

[WEIDER 94]

"Architecture of the Whois++ Index Service", Chris Weider, Jim Fulton, Simon Spero, Internet Draft July 1994, <draft-ietf-wnils-whois-03.txt>