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, Mark Madsen, Ashley McClenaghan

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.1306.00.19 **Draft** 16th May 1995

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Superseded by: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, Mark Madsen, Ashley McClenaghan

APM.1306.00.19

16th May 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	(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 requirements for context and content based searching, such as through provision and distribution of appropriate metadata resources.

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.

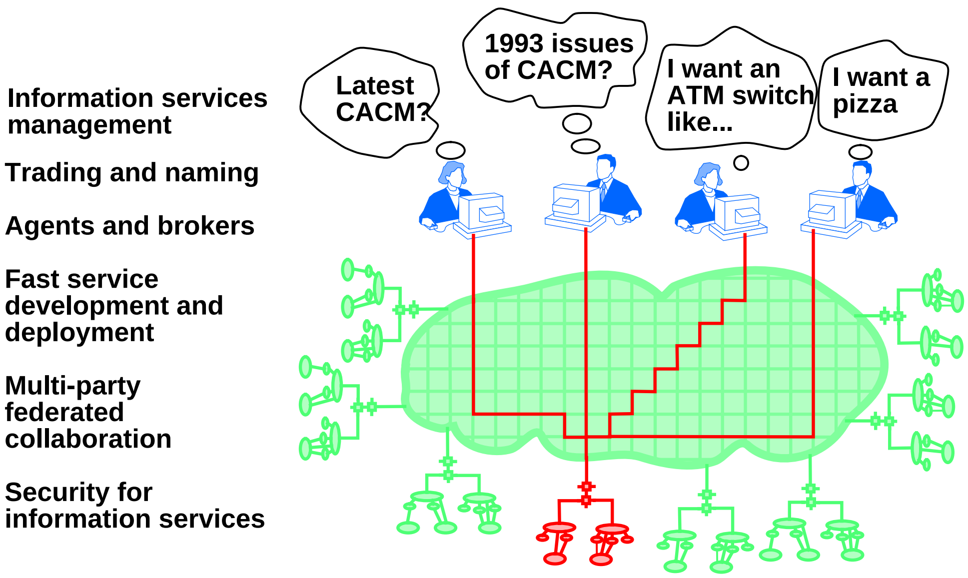
Must be able to act on behalf of user - authentication, remuneration issues.

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 (such as uri@bunyip.com, www-talk, and others)

Influence IETF developments through direct interaction with key developers.

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. §3.3.4 *CORBA/WWW interworking*, §4.2.4 *A Management platform for the internet* §3.3.4.1 *Web re-engineering* §4.2.5 *Re-engineering the World Wide Web*

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.*

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

Note: As noted at the Technical Committee Meeting December 6th 1994, this task is unlikely to be finished by 3/95. However we should be able to demonstrate significant progress by then.

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.

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: see §3.3.2.2 *Meta information model for URCs*.

b) agents and how they can implement information brokering, meta-information capture and management, and Web resource mapping.

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, together with the necessary meta-information management systems. The resulting systems will display the capabilities necessary for 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, Harvest, and Whois++.

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.

2.3 Meta-information

The above mentions meta-information because the chapters which describe the work in progress (§3 and §4) refer to meta-information directly and indirectly. Workpackage B1 needs to put in place the basic concepts in infrastructure to support the use of meta-information. Workpackage B2 will involve building the meta-information services identified in B1. Part of workpackage B3 will involve building agents and scripts which can exploit these meta-information services, in addition to capturing and generating meta-information directly. Meta-information services are described in more detail in §5 and in [Madsen 95].

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. There are at least two distinct kinds of meta-information.

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.

A second kind of 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?

3.3.4 CORBA/WWW interworking

3.3.4.1 *Web re-engineering*

In order to re-engineer the World Wide Web, (§4.2.5 *Re-engineering the World Wide Web*) there are some conceptual problems to solve. These are the design issues for the IIOP migration work.

- Stream behaviour
  - — render as data arrives, concurrent retrievals
  - — extend IDL, IIOP?
  - — IDL for HTTP depends on solution
- Addressing
  - — resource named relative to protocol by URL
  - — how do we know that IIOP is available?
  - — migration to URN is the same problem, not a solution

There may be other issues to resolve as well.

3.3.4.2 *Management platform*

What concepts are needed to support §4.2.4 *A Management platform for the internet*? Can HTTP be represented in the CORBA computational model (almost certainly yes)? If so how?

Ideas to explore:

- generic interceptor - where does it get mapping rules?

- dynamic or static generator - what options are feasible?

Analysis of the HTTP protocol is in progress. Several CORBA IDL mappings are possible, with different emphasis on capturing the concepts in HTTP or the representations of those concepts. There are also several possibilities for representing the many optional aspects of HTTP, and these are being explored.

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*

Tentative; needing development of concepts

- §4.2.4 *A Management platform for the internet*

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.

An infrastructure has been built which makes many of the enhanced trader's interfaces available through WWW, showing that trading is feasible in WWW.

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).

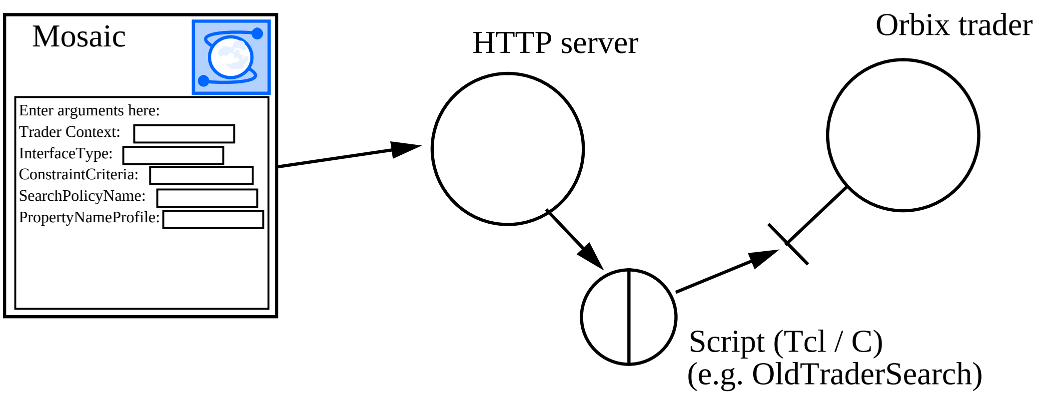
4.2.1.4 Role of trader in authoring

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:

- 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:

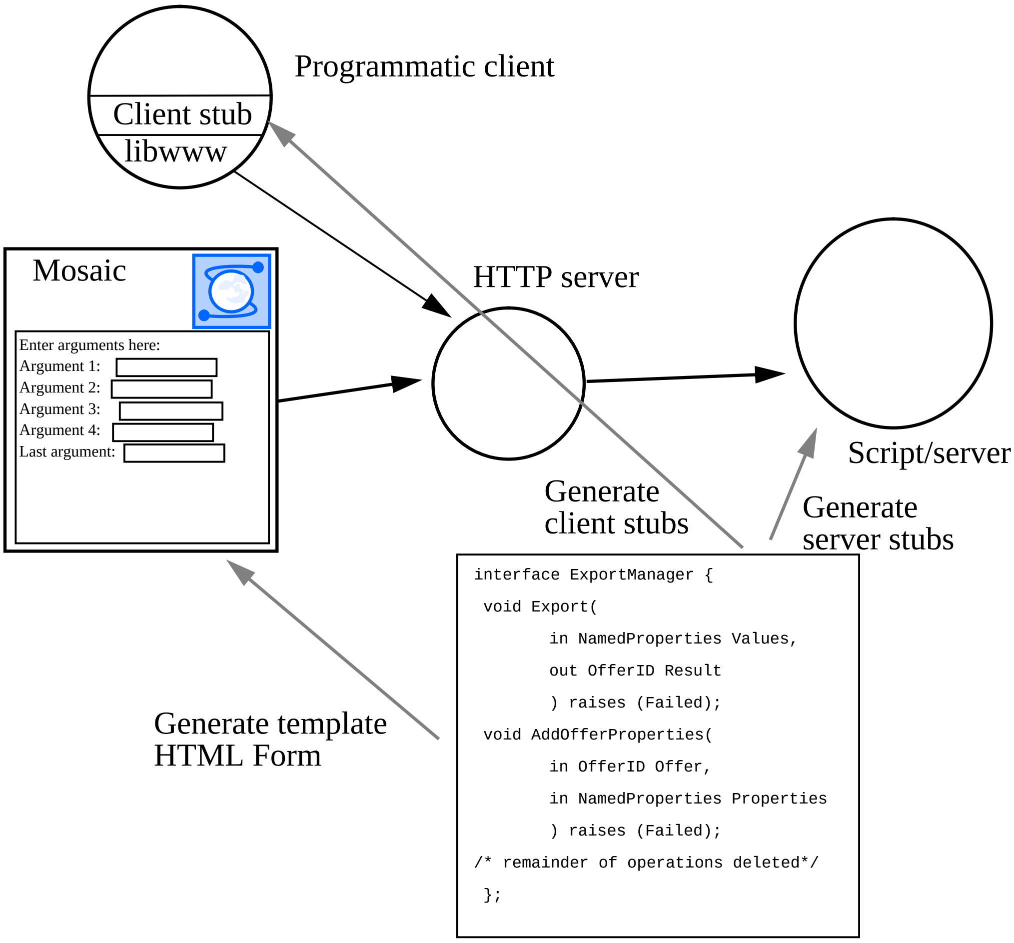
- Generate code to marshal results (programmers wouldn't have to worry about explicitly writing the results in stdout - it would be done automatically).

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 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.

The above is implemented and documented in [EDWARDS 95b] and [EDWARDS 95c].

4.2.3 Making HTTP an Extensible RPC Protocol

The draft HTTP specification suggests that the authors' intention is to turn HTTP into a general RPC protocol [BERNERS-LEE 94]. We are in an excellent position to show how this can be done: the work in §4.2.2 puts in place many of the basic components needed.

Certain methods will be standardised (e.g. GET, PUT, HEAD, POST, DELETE etc.); it is proposed IANA (Internet Assigned Numbers Authority) controls these and deals with applications to standardise new methods.

However, clients and servers should be free to define their own methods. This facilitates innovation and market differentiation — something which will be important for commercial servers. In addition by providing, an easy way of programming clients we will make it easier for vendors to provide a bespoke client for their web server to take full advantage of any additional functionality they have added. Currently the only way to drive any additional functionality is by using html form technology.

One question which arises with this proposal is: "how does a user get hold of the bespoke client application in the first place?" In the short term the answer is probably by ftp; in the long term it would be really nice if the client application could migrate automatically to the users machine, perhaps using some kind of agent technology.

Current management practice in the web is somewhat ad hoc. HTTP defines the methods "PUT", "POST" and "DELETE" which are intended to allow for replacing, creating and removing objects from the server. However, the precise semantics of these methods are usually determined by the local environment. Thus it is not possible to provide generic routines in the server; instead these routines, if they are implemented at all, are implemented as scripts which are forked by the server. This means that any state associated with the script has to be stored externally; usually this is on the local file system.

Our proposal involves restructuring the web server to make it look more like a traditional CORBA server. This would make it very easy for people to insert their own PUT, POST and DELETE methods inside the server program itself; of course they would still be free to use a script as before. The benefit of having the routine internal to the server will be improved performance and stateful interaction.

In essence HTTP is defining a standard supertype for all HTTP object interfaces, where the supertype contains the standardized messages. Something which needs attention is the generic nature of operations with names like PUT and GET. They often take a parameter which defines the type and name of variable being put or got. This is close to the CORBA DII style of interaction and leads you into playing with the "Any" data type, which is an area where CORBA and ANSA diverge (in as much as ANSA has a story to tell.)

It would not be difficult adapt the backend of the CORBA compiler in §4.2.2 to do this. As part of the work for §4.2.2 we have made a modifications to the existing HTTP protocol stack in libwww so it will support POST instead of just GET and HEAD. (This experience suggests it would not be too hard to generalise this work.)

To make execution of CGI scripts efficient we are going to modify the CERN HTTP server so that the CGI script can co-exist in the same process rather than as a separate process which gets forked each time a request occurs. The

CGI scripts can be thought of as server specific HTTP methods. Hence this will produce a version of the CERN server which can be extended to support new server specific HTTP methods. The signatures of the methods will be defined in CORBA IDL.

The results of §4.2.2 have already produced a "programmatic client" which can be used to drive these new methods.

The only question that remains is how to allow browsers to take advantage of these new methods. We can already generate (template) html forms from CORBA IDL which allow browsers to invoke operations. However, the main reason for wanting to extend HTTP with new methods which use more complicated types is to support new programs rather than existing browsers. (Humans are really only likely to type in strings and numbers.)

These new programs will provide new services to network users as well as manage the network.

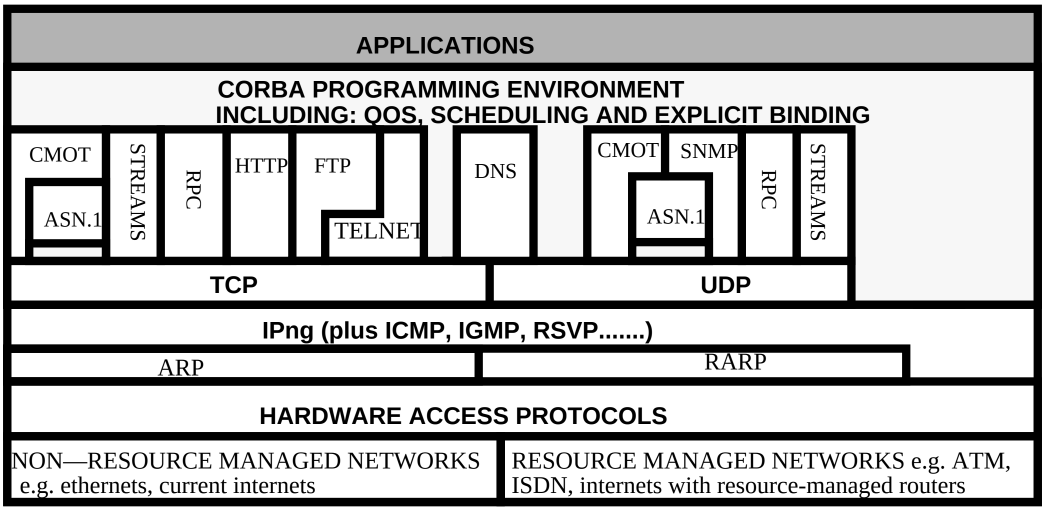
This work could be positioned as a better way of doing CGI, but any modified server should support both the old and new style interactions.

This work has not yet been started, but some detailed ideas are discussed in [EDWARDS 95c].

4.2.4 A Management platform for the internet

However consider figure 4.3 taken from [EDWARDS 94b] which shows how the architecture described in [COMER 91] might develop given the emergence of new protocols such as RSPV [ZHANG 94] and IPng [HINDEN 94] (or IP version 6).

Figure 4.3: A programming environment for the internet



We hypothesize that there is scope for a programming environment for the internet which makes the various internet protocols available to the programmer in a consistent manner. Furthermore, we hypothesize that CORBA could become this environment. Once we have completed the work in §4.2.3 we will know how to represent HTTP in the CORBA computational model. The next step would be to cut this protocol into an existing CORBA platform (or ANSAware). If this is successful we could investigate cutting some of the other protocols in the above diagram into the platform.

This will enable various kinds of internet services to be made available through CORBA. However, much more importantly, it enables CORBA applications to talk to other applications and services which use these protocols, thus enabling CORBA to become the programming platform of choice for building management applications in the emerging information superhighway.

4.2.5 Re-engineering the World Wide Web

4.2.5.1 Overview of the problem

HTTP the HyperText Transfer Protocol designed for the World-Wide Web accounts for an increasing proportion of the traffic on the Internet. The specification of HTTP/1.0 describes HTTP as "a generic, stateless, object-oriented protocol which can be used for many tasks".

The strength of HTTP is its simplicity, but this is also its weakness. It is easy to build simple clients and servers for HTTP, but the protocol is, by its nature, wasteful of resources. HTTP uses the underlying TCP protocol in a way that compounds the latency problem.

The simplicity of HTTP is in implementation for basic document delivery. Although intended to be usable for many tasks, there are deeply embedded assumptions about the nature of the objects, and a complex collection of optional modifiers. HTTP is extensible, but there is no framework of types or interface definitions within which to position extensions.

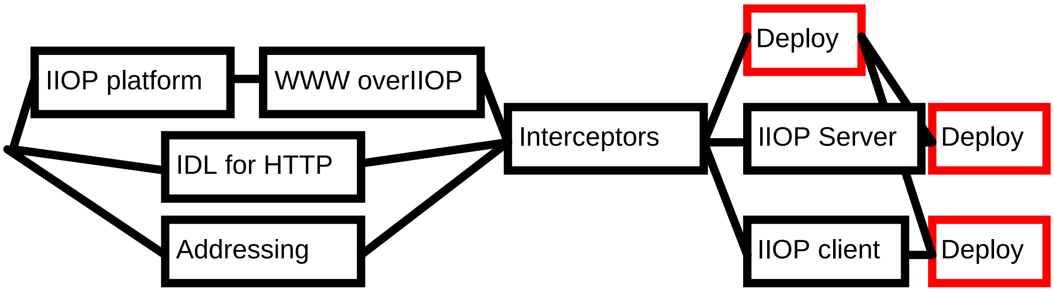
There are proposals to improve HTTP, and to modify it so as to mitigate some of the latency and performance problems. The need for improvements is seen as urgent, but the progress of these proposals has been slow. These proposals do not address the issue of a framework for extensibility.

There is now an opportunity to exploit in WWW the RPC protocol design experience that has culminated in the CORBA Internet Inter-Operability Protocol (IIOP). IIOP offers a superior base both for current WWW requirements, and for the evolution towards a web of distributed objects.

4.2.5.2 Strategy for migrating WWW onto IIOP

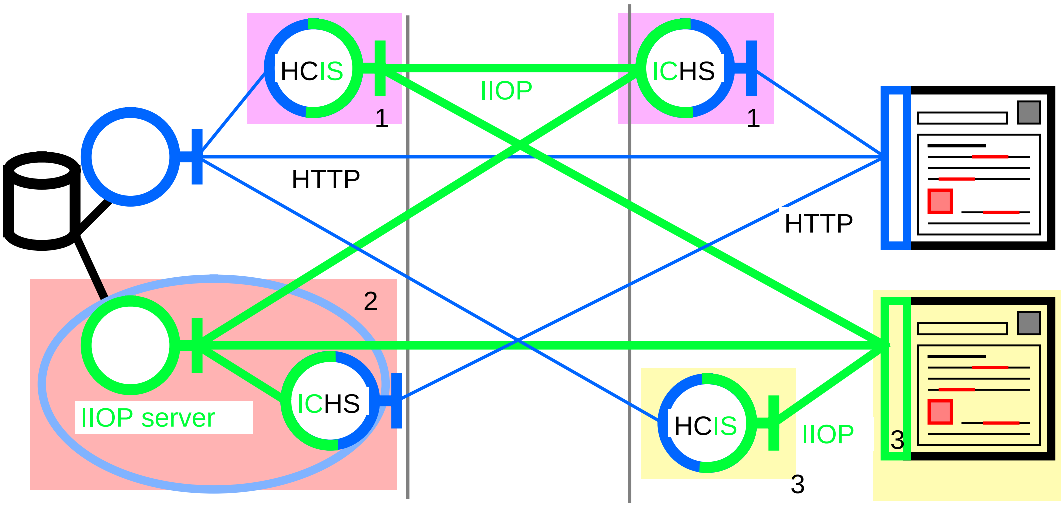
Figure 4.4 shows an outline of a strategy for migrating WWW onto an IIOP base that addresses the immediate problems while offering the opportunity for evolution.

Figure 4.4: WWW IIOP migration strategy



It is essential that there is a migration path from the current HTTP based technology. Figure 4.5 gives an overview of the migration path.

Figure 4.5: Overview of the evolution path



4.2.5.3 *The critical success factors*

- Time is of the essence
  - deploy before HTTP-NG entrenched
- Anything you can do, I can do better
  - must not lose any functionality
  - render as data arrives?
- Availability
  - basic version available free by FTP
  - all platforms – portability
- Interoperability
  - must interwork with HTTP
  - transparent redirection of URLs

5 Meta-Information

This chapter describes work on the ways in which meta-information needs to be incorporated into the larger arena of information management.

Initially, it is proposed to use the storage of meta-information as the basis for a prototype system to fit in with some of the aims of the Information Services Framework.

The idea behind this work is manifold. Firstly, to extend the scope of the application of current work to the World Wide Web. Secondly, to provide a practical testbed for the application of meta-information methodology to the design of a management engine [Madsen 95]. Thirdly, to extend and broaden the available expertise on the capture and generation of meta-information [Madsen 94].

5.1 Deliverables

The following kinds of output are expected from this work:

- A usable understanding and evaluations of the kind of meta-information support systems already in existence.
- The design of an extensible Web-based meta-information management system.
- A prototype of this design and the accompanying experience with the creation of an integrated management engine.
- Positive influence on the work of the IETF (Internet Engineering Task Force) towards the development of stable futureproof standards for metadata to be used in the WWW.

5.2 Summary of Relevant Projects

5.2.1 Evaluation of meta-information support components

This will be an overview study of the principal projects currently being developed elsewhere. This study will be directed towards mining the resource base comprised by the work of those projects. Primary questions to answer are:

- What capabilities are present in prototypes produced by existing research projects? Do these solve technical problems currently set out in the workpackage descriptions?
- How might these capabilities be integrated with the capabilities of other systems? Can uniqely new services be created by such integration?
- What kinds of meta-information do they require, and how do they go about generating, acquiring and classifying it?

- Are these capabilities scalable? If not, can they be re-engineered into a form with good scaling properties using ANSA technology?
- What is the relationship between distributed trading and scalable indexing services?

The main targets for this aspect of the work that have been identified are:

- The Harvest Project at the University of Colorado. Harvest consists of a synthesis of a number of technologies, which together are used to create searchable specialist WWW indexes. Chief components among these are the *gatherer* and the *broker*. The role of the gatherer is to begin from a set of specified URLs and perform a depth-limited breadth-first indexing search (keyphrases as well as keywords are used). The basic Harvest idea is that ultimately, each information provider should run their own gatherer. These gatherers are relatively lightweight processes compared to current Web robot technologies, creating indexes that are only a few percent of the size of the base information sets, and doing so with orders of magnitude less WWW traffic. The role of the brokers is to exchange meta-information about the gatherer indexes to which they have access with other brokers, and to filter and distribute search requests. A typical broker maintains contact with a number of other brokers and at least one or two gatherers. Because brokers are as distributed as the specialist indexes of which they are aware, the load problems associated with centralised indexes such as those maintained by Lycos, World-Wide Web Worm, The Webcrawler, Jumpstation, and others do not occur.
- The Microcosm/Multicosm Project at the University of Southampton. Microcosm is an architecture for hypertext systems that dislocates the links between documents from the contents of those documents, with the effect that the hypertext links can be located in searchable linkbases, and have associated metadata incorporated within the linkbase. The Multicosm architecture extends the Microcosm ideas to distributed systems, and also incorporates multimedia support within the architecture.
- The KQML Agent-Communication Language Project of the DARPA Knowledge Sharing Initiative External Interfaces Working Group.
- Obliq, a distributed scope language being developed by Luca Cardelli at DEC Systems Research Center, Palo Alto.

Obliq and KQML are important because it may eventually be neccessary to take advantage of the facilities they offer that are otherwise only available within Telescript, which is a proprietary agent-language technology belonging to General Magic.

It is also worth noticing that the Harvest architecture is closely related to some of the early proposed uses of agents (“mediators”) in managing very large database systems [Wiederhold 92].

5.3 A Meta-Information Management Prototype

5.3.1 Meta-information repository and management engine

This will be capable of interfacing with Web browsers and the CORBA IDL trader. The aim is to be able to issue a request for a resource by its URN, and have this request resolved appropriately in transparent fashion.

It is envisioned that this repository will benfit from the incorporation of linkbase technology based on that used in Microcosm.

5.3.2 Initial tasks

Initial tasks include the following:

- Build a meta-information repository incorporating extensible querying. The query process will be based on that developed within the architecture of [Madsen 94]. Extensibility will be assured by building the repository and its query mechanisms in Tcl.
- Construct a meta-information management engine, also using Tcl/Tk, that can use the meta-information repository to retrieve URNs, resolve them to URCs, query meta-information stored within URCs, and retrieve the implied URLs. This will also require:
- A basic resolution service capable of URN->URC and URN->URL resolution. It is expected that this will draw on the CORBA IDL technology that has already been developed for the WWW.
- A control interface that will handle the lower-level problems to do with passing resource identifiers to and from a WWW browser. This is straightforward with, for example, NCSA Mosaic because it has a script port. However, an issue that arises is how much distribution support is desirable to incorporate within this aspect of the prototype.

(See also §3.3.2.)

5.4 IETF Standards

This aspect of the metadata work is directed at developing appropriate contacts within the IETF and providing them with useful input for draft standards. The focus of the work has been how metadata can, and should be, encoded within URCs (Universal Resource Characteristics). The status of these contributions is as follows:

- A contribution on metadata-based searching has been passed to Ron Daniel (Los Alamos National Labs) for inclusion in the “URC Scenarios and Requirements” draft. This draft should be completed in time for the Stockholm meeting of the IETF in July.
- Following email exchanges on the structure of URCs and the ways they encode metadata, Ron Daniel has asked MSM to co-author the rewrite of the URC specification draft.
- Ron Daniel has begun synthesizing a metadata taxonomy [DANIEL 95] as part of the general attack on Web-based aspects of the metadata problem. This draws on [MADSEN 94] along with material from Ron Pfaff (also at LANL).

5.5 Related Work

The metadata research is being conducted in close connection with the Scripts and Agents work, and some aspects of the work of the Federation Group.

6 Scripts and Agents

This chapter describes the work on scripts and agents.

6.1 Deliverables

- An extensible, script-based Web server prototype.
- An evaluation of technologies and products for programming the Internet.
- Identify and develop opportunities for using new Internet technologies.

6.2 Components

6.2.1 An extensible, script-based Web server prototype

Changeling is a prototype, script-based Web server which supports an extensible set of methods.Methods are implemented as Safe-Tcl scripts. Changeling clients may install new methods, or inspect, invoke or remove existing methods via the HTTP protocol while the Changeling server is running.

This prototype is being used to explore the issues of safe interpretation of scripts, and the dynamic updating of servers (in-service upgrade). The prototype will also be used as a basis for other Information Services related work, such as URC lookup and resolution.

6.2.2 An evaluation of Internet programming technologies

Recently a number of new technologies for programming on the Internet have been released. These include :

- scripting languages such as Sun's Java [HotJava], General Magic's Telescript [GM-TeleLess93][GM-TeleProg93], (Safe-)Tcl [Tcl/Tka][MIME/Safe-Tcl], Silicon Graphics' Virtual Reality Modelling Language
- protocol developments such as S-HTTP and SSL
- model developments such as OSF's DCE based Web.

This objective of this evaluation is to learn about and experiment with these new Internet technologies.

6.2.3 Identify opportunities for Internet technologies

The Internet is becomming important to business organisations as a means of integrating their IT systems, of providing customer services, of advertsing and of data gathering. New Internet programming technologies are opening up new business opportunities. The purpose of this deliverable is to help our sponsors develop and use Internet programming technologies to create new business services on the Internet.

Ideally we would like to work with one or more of our sponsors to help develop Internet based services using the latest Net technologies such as Java, Tcl, VRML, etc. This would give us the chance to evaluate these technologies and assess if they can be integrated and used to support real business services.

6.3 Relationship to other work

With respect to the extensible Web server prototype: the HTTP protocol is defined to be extensible. This is not yet widely exploited, but may be an important aspect of WWW/CORBA interoperability. The extensible HTTP server can be used to explore the extensibility in order to understand how it relates to WWW/CORBA interoperability.

With respect to Internet programming technologies: scripts and agents are expected to play an important role in the exploitation of metadata [MADSEN 95a] and this work area therefore retains close connections to that described in Chapter 5.

6.4 ANSA documents

- Scripts and Agents: Introduction and Work Proposal [McClengahan95a]
- The Changeling Web Server [McClenaghan95b]

7 Issues

7.1 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.

7.2 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.

7.3 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.

7.4 Generality

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

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

7.5 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].

The following is our preliminary conclusion from a very quick skim through the above documents.

Whois++ is a proposal for a distributed indexing service in the internet. Data is stored as (attribute, value) pairs and constraint based searching is possible (the constraint language looks very simple).

The two main differences between this and trading appear to be:

- The approach to federation
- Traders are intended for use by programs; the whois++ seems to be intended for use by humans (e.g. you telnet into it as you telnet into an archie server now). However, whois++ can be configured for program access. (This is how people are proposing it will support WWW.)

The approach to federation is very different. Instead of linking to other whois++ servers (like traders link to other traders), there is a distributed indexing service. Each whois++ server is attached to an index server which will index it and other index servers. Index servers themselves may be attached to other servers etc. After reading the whois++ documentation our conclusion is that it would not take much to turn a trader into a whois++ server. If we did this we may well have something much superior to what the specifiers had in mind and better than most other implementations

7.6 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.

7.7 Place Holders for future work

The following section lists possible items for future work which we do not intend to work on at present.

7.7.1 Author/resource management

Concepts to support prototyping §7.7.4 *Author/resource management*

7.7.1.1 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 URCs are in use.

- Author uses URN/URL, tool finds URC?
- 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

7.7.1.2 *Maintenance tools*

Track resources - maintain integrity of links - active URC 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.

7.7.2 Publishing documents

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.

7.7.3 Relocator

Relocator: refresh stale URC (passive - delivers new information when requested)

7.7.4 Author/resource management

Track resources - maintain integrity of links - active URC refresh

Install document into server

Author uses URN/URL, tool finds URC 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.)

8 Diary

8.1 ANSA core team review 21 October 1994

Version 00.04 discussed at a project meeting. Issues raised were added to §7 of version 00.05.

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

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

8.3 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). Now we need to figure out how to build a back-end for WWW!

8.4 Trader WWW Demonstration to HP Bristol 21 November 1994

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

8.5 Jon Crowcroft 25 November 1994

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

8.6 Technical Committee 7,8 December 1994

Three presentations on this document explaining §3 and §4 as well as future work (slides in [EDWARDS 94a], [REES 94] and [EDWARDS 94b]).

8.7 Joe Sventek 16 December 1994

Discussed this work and how it might be applied to HP - lots of possibilities.

8.8 December 23 1994: CORBA IDL and WWW Scripts

ANSAware Echo service and clients successfully ported to become WWW clients and servers.

8.9 January 18th 1995

Presentation to BNR as part of a set of presentations on the ANSA work program.

8.10 January 20th 1995

Visit by Steven Hayles of Leicester University. Steven talked about the STILE project which exploits web technology in novel way allowing lecturers to create a hypertext system containing information relevant to courses. Of particular interest is the use of scripts made to extend the web's functionality to allow easy creation and management of the information.

8.11 February 1st 1995

Visit to Leicester University to learn more about STILE.

8.12 February 6th 1995

Visit to HP Labs Bristol taking them through [EDWARDS 94b].

8.13 February 28th, March 1st 1995

Technical committee meeting. The following presentations were made: [EDWARDS 95a], [MADSEN 95a], [MADSEN 95] and [McClengahan95a].

8.14 March 8th, 1995

Attendance at BCS World Wide Web Seminar Meeting.

8.15 March 28th, 1995

Visit to Multicosm Project Group in the Department of Electronics and Computer Science at the University of Southampton.

8.16 April 5th 1995

Presentation of APM.1451 at ANSAworks describing the work discussed in §4 and §3.

8.17 April 10th to 14th

Third International World Wide Web conference. RTOR attended and sat on the security panel.

8.18 May 3rd and 4th1995

Visit to OSF RI (NJE representing HP). Some of the work in this document was discussed

8.19 May 2nd 1995

APM.1473 contributed to Proceedings of CKBS-95 meeting.

8.20 May 25th and 26th 1995

ANSA Technical Committee and WWW workshop to discuss “WWWng”.

References

[BERNERS-LEE 94]

"Hypertext Transfer Protocol — HTTP/1.0", T. Berners-Lee, R.T. Fielding, H. Frystyk Nielsen, <draft-fielding-http-spec-01.ps>.

[COMER 91]

"Internetworking with TCP/IP", Volume I; Principles, Protocols and Architecture, D.E. Comer, Prentice-Hall 1991 (second edition).

[DANIEL 95]

"Metadata Taxonomy", <URL:http://www.acl.lanl.gov/URI/URCspec/taxonomy.html>

[EDWARDS 95a]

"A CORBA IDL Compiler for the World Wide Web", N.J. Edwards, APM1419, February 1995.

[EDWARDS 95b]

"A Stub Compiler for CGI: The Programmer's Guide", APM.1465, APM Ltd., Cambridge, U.K. May 1995 (in preparation).

[EDWARDS 95c]

"Object Wrapping (for WWW) - The Key to Integrated Services?", APM.1464.01, APM Ltd., Cambridge UK, April 1995.

[EDWARDS 94a]

"Information Services Framework — Overview (Slides)", N.J. Edwards, APM1368, December 1994.

[EDWARDS 94b]

"WWW and CORBA — Slides", N.J. Edwards, APM1370, December 1994.

[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>

[GM-TeleLess93]

Scott Kronick, "Telescript Language Lessons - The Fun Book", General Magic Inc., 2465 Latham Street, Suite 100, Mountain View, CA 94040, USA, June 1993, Magic 270

[GM-TeleProg93]

Susan Rayl, "Telescript Programming", General Magic Inc., 2465 Latham Street, Suite 100, Mountain View, CA 94040, USA, April 1993

[HINDEN 94]

“IP Next Generation Overview”, R.M. Hinden, October 1994, <draft-hinden-ipng-overview-00.txt>

[HotJava]

Sun Microsystems Labs., “HotJava Home Page,” <URL:http://java.sun.com/>,March 1995[McClengahan]

[MADSEN 94]

“Metadata Systems: Integrative Information Technologies”, M. Madsen, I. Fogg, C. Ruggles, Libri 44(3):237-257, 1994.

[MADSEN 95]

“The Application and Management of Distributed Meta-information”, M. Madsen, B. Crawford, APM.1409, February 1995.

[MADSEN 95a]

“Agents for Knowledge Resource mapping in the World-Wide Web”, M. Madsen, APM.1473, in Proceedings of Intelligent Agents and the Next Information Revolution: CKBS-95, May 1995.

[McClengahan95a]

Ashley McClenaghan, “Scripts and Agents: Introduction and Work Proposal”, 1434.00, APM Ltd., Cambridge UK, 1995

[McClenaghan95b]

“The Changeling Web Server”, Ashley McClenaghan, APM.1453, April 1995.

[MIME/Safe-Tcl]

Nathaniel Borenstein, Marshall T. Rose, “MIME Extensions for Mail-Enabled Applications: application/Safe-Tcl and multipart/enabled-mail”, November 1993

[REES 94]

“Naming and Trading in the World Wide Web (TC Presentation)”, R.T.O. Rees, APM1367, December 1994.

[Tcl/Tka]

John S. Ousterhout, “Tcl and the Tk Toolkit”, Addison-Wesley, 1994

[WEIDER 94]

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

[WIEDERHOLD 92]

“Mediators in the Architecture of Future Information Systems”, IEEE Computer, March 1992, pp38-49.

[ZHANG 94]

“Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification”, L. Zhang, R.Bradem, D. Estrin, S. Herzog, S. Jamin, July 1994, <draft-ietf-rsvp-spec-03.ps>.