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

Metainformation Management on the World Wide Web

Mark Madsen

Abstract

The World Wide Web represents an enormously valuable resource for the repository and retrieval of unlimited quantities of information. The most important outstanding problem is how to add the value required in the context of a particular use or user. Ideally, the information consumer would be able to determine in advance what kind of information exists on the Web that is relevant to their needs and how to access that information. On the World Wide Web, solutions to this problem are hindered by the technical difficulty of obtaining accurate metainformation describing potentially useful sets of information.

This document sets out an experimental design for the kind of metainformation management system needed to solve these problems. Key aspects of this solution are the storage of metainformation in URC (Universal Resource Characteristic) form and the use of the Web itself as the storage medium for both information and metainformation.

APM.1405.00.01 **Draft** 6th June 1995

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

Distribution:
Supersedes:
Superseded by:

Metainformation Management on the World Wide Web

Metainformation Management on the World Wide Web

Mark Madsen

APM.1405.00.01

6th June 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	**1**	**The Problem**
1	1.1	Overview
1	1.2	The business problem
1	1.3	The technical problem
3	**2**	**The Solution Requirements**
3	2.1	Functional requirements
3	2.2	Structural requirements
4	2.3	Components of the proposed solution
6	**3**	**Meta-Information Management**
6	3.1	The Form of Metadata Objects
6	3.2	The Meta-Information Management Engine
7	3.3	Metadata Query Processing
7	3.3.1	Query construction
7	3.3.2	Query encoding
7	3.3.3	Query processing
8	**4**	**The System Prototype**
8	4.1	The Elements of the Web
9	4.2	Building the Prototype
10	4.2.1	WWW Client
11	4.2.2	Changeling
11	4.2.3	URC Repository
11	4.2.4	Tcl Gateway
11	4.2.5	Safe-Tcl Interpreter
11	4.2.6	Broker
12	4.3	Future Developments

1 The Problem

1.1 Overview

The World Wide Web, and the information contained within it, provides a resource of impressive magnitude. Attempts to mine this resource are presently limited by the fact that technologies for the provision of information on the Web presently outstrip the technologies available for effective extraction of information from the Web, unless both the qualities and exact location of that information are already known to the agent attempting its retrieval.

The value associated with the resources contained in the Web consists primarily of the intrinsic value of the information represented by stored resources, and secondarily with the extrinsic value represented by services which provide required information, or access to classes of required information. Business value can therefore be extracted from the Web by services that handle categorisation, classification, and delivery (whether direct or indirect) of resources.

1.2 The business problem

The problem is to find effective ways of categorising and managing the information available on the Web. The business opportunities arising from solutions to this problem are twofold. Firstly, the management services and the results of those services can be sold. This kind of business is already active on the Web and is obviously growing. For example, there are now a variety of commercial resource-location indexes available, often based on the use of robots to acquire the metadata needed to locate resources. The second form is the sale of management systems tailored to the needs of particular kinds of information enterprises. This latter form of business activity is also now becoming part of the fabric of the Web, with the construction of secure servers and the socket security layer for financial transactions.

1.3 The technical problem

The technical problems posed by the management of metainformation relevant to the World Wide Web are as follows:

1. Maintenance of metainformation: this involves the storage, versioning, updating and consistency control of appropriate metainformation relevant to a given Web resource.
2. Resolution of metainformation into information: the need is for the system to generate paths leading to information directly from supplied metainformation.

3. Effective and efficient mechanisms and support facilities for searching through large quantities of metainformation. This process should be transparently automated wherever possible: the agent responsible should maintain a state which allows it to take account of the preferences, interests, knowledge and understanding of the client invoking that agent [MADSEN95].

2 The Solution Requirements

This chapter discusses and derives the requirements that the proposed Web-based service prototype must satisfy in order to deliver the business capabilities set out in the previous chapter.

2.1 Functional requirements

The additional functions that are not presently provided by the Web will need to be added through the development of external (distributed) facilities. These facilities will then need to be integrated with the Web and its supported protocols and capabilities. This level of integration can be achieved by use of an appropriate scripting language to provide the systemic glue that holds the components together. Reasons of availability, ready expertise, simplicity and rapidity of prototyping speak strongly in favour of Tcl as the appropriate glue language.

Functionally speaking, the system is intended to provide a platform for experiments with metadata-based search and retrieval on top of the WWW. It therefore needs to be able to perform the following:

- store, retrieve, update and manage metadata
- search the metadatabase for matches made on the basis of comparison with supplied metadata using supplied methods
- retrieve resources via a short interaction path from their metadata

These requirements in turn impose restrictions upon the structural requirements for the system.

2.2 Structural requirements

The single most difficult choice to make in developing any system to manage metadata is that of a suitable and flexible encoding for the metadata. This is because metadata is required to be close to the resource it describes in terms of user value, while being highly ordered from an automation viewpoint. For the purposes of the present experiment, the best choice is to use the tools already supplied for the WWW, where the intended standard for the encapsulation and encoding of metadata is called the URC (Universal Resource Characteristic).

Note: At the time of writing this draft, URCs as lists of attribute:value pairs had been abandoned in favour of a model with single-class inheritance. This latter model has not yet been firmed up, so for the puposes of experimentation, URCs are modelled as Internet RFC 822 type attribute-value forms.

2.3 Components of the proposed solution

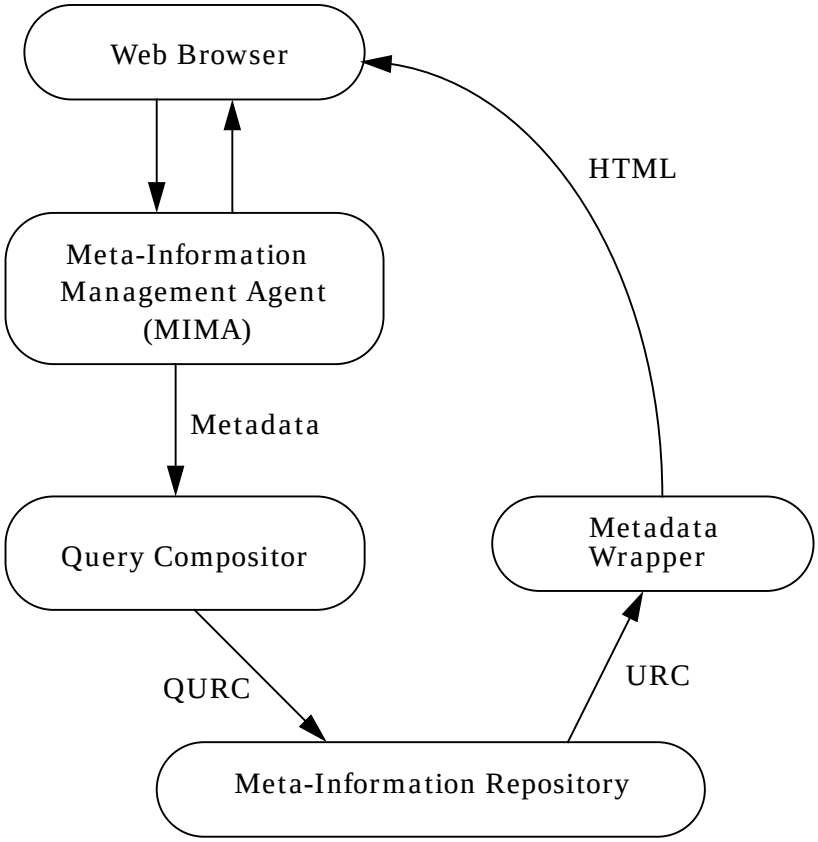
Solution of the problems posed in the preceding sections requires advances in the use of the existing technology: the facilities provided by the Web do not suffice on their own. The proposal is therefore to augment the resource storage and retrieval methods provided by the present Web by integrating a number of enhanced-technology components into the client's perceptual horizon. The essential elements are:

- A local metainfo repository which stores all locally held metainfo in the form of URCs
- A metainfo manager for accessing the metainfo repository and other services
- A query compositor, which wraps queries into the form of URC-like objects with their associated comparison methods
- A wrapping service, responsible for wrapping returned URCs into HTML that can be displayed by the WWW client
- A URC -> URL resolution service: in practice, this will make use of the HTTP facilites already supported within the WWW client

These components are shown in Figure 2.1, while the choices made in developing their representation in terms of existing available technology and requisite new components is discussed in depth in 4.

(This diagram is somewhat more general than is required for the present discussion, which is confined to the integration of metainfo management within the existing Web framework.)

Figure 2.1: The Components of a Meta-Information Management System



The flow of information and meta-information within the proposed prototype system. Here QURC represents a query-URC, in other words a legal URC which is constructed from the meta-information that is passed through by the MIMA. This QURC is used as the archetype against which the URCs stored in the meta-information repository are measured. The repository returns URCs via a metadata wrapping process which encapsulates the URCs into HTML documents so that they can be displayed by a standard WWW client/browser. If the user chooses, the links (URLs) contained within the URCs can be followed from the client on the basis of evaluation of the other metadata contained within the URCs.

3 Meta-Information Management

This section expands on the ideas sketched above by describing the ways in which metadata is both encapsulated and encoded, and the methods by which it is managed.

3.1 The Form of Metadata Objects

In the prototype system, metadata objects are constructed as URCs [DANIEL]. This has the advantages that URCs were planned from the outset to become the mode of representation of metadata within the WWW and are presently planned to become an IETF standard before the end of 1995.

The main disadvantage of URCs at the time of writing is that their exact formal structure has not yet been either formally laid down or standardised. It is hoped that this situation will change within the useful lifetime of the present document.

The way that advantage has been taken of the URC proposal is by using a simpler form for URCs than that which is expected ultimately to become standardised. The form which has been chosen is simple enough that it will correspond to a legal form of URC within any conceivable future standard. In simple terms, the decision was taken to use an RFC822 type of list containing attribute:value pairs. No URC is required to have any specific attributes, while the names of attributes are not fixed: they are identified as unitary strings following a newline and preceding the colon symbol. The only restriction on the values corresponding to attributes is that they must consist entirely of printable ASCII text and whitespace characters.

3.2 The Meta-Information Management Engine

The main ideas here are fairly general: the central idea is that, given the existence of a large, distributed, possibly heterogeneous repository of information, the best way to select, locate, and extract the items of information which are most relevant to the enterprise at hand is through the specification of appropriate meta-information which can be used as a reference comparator against which the meta-information describing candidate information items can be measured.

The overall functionality required from the system can thus be broken down into the sections corresponding to those shown in Figure 2.1. There is at least one meta-information repository (and in general many), which receives queries and responds with appropriate elements of stored meta-information. The queries originate with an agent and are packaged into the appropriate form by a query compositor. The response meta-information contains sufficient detail to allow retrieval of the appropriate information resource, which is then evaluated, and passed back to the agent if of sufficient quality. In this picture,

the agent may be controlled by a human, but need not be, and may be acting on behalf of another agent at a possibly different location. This can be made possible by having the agents use a trading service to get in touch with each other initially.

3.3 Metadata Query Processing

The exact details of metadata querying will depend on the detailed structure and storage of metadata within the system. At the same time, it is clearly necessary that a good querying scheme have the following general characteristics:

- robustness - it should not be necessary to alter the existing methods of the querying scheme in order to accommodate new forms of metadata, although it may be necessary to add new methods.
- scalability - it should be possible for the spreadout of queries to be subject to distributed management.

The first of these requirements depends on the system design for an efficient query construction, encoding, and processing methodology. Satisfying the second requirement is a task for the distribution layer.

The model of querying that is used here is based on that devised in [MADSEN94], modified to use URCs as the metadata representation. The querying components for this framework are taken in turn.

3.3.1 Query construction

This is achieved by using the input as the outline of a URC against which other URCs should be matched in order to determine their suitability as responses to the query.

3.3.2 Query encoding

The constructed query is encoded as a QURC (Query-URC), which is an object containing data in the form of a URC representing the target template against which other URCs will be compared, along with the methods that can be applied to search for matching URCs. These methods may be

1. standalone scripts in, for example, Tcl, Obliq, or Java
2. interface references to programs that may (or are expected to) be available on remote nodes

It is not intended that every query require the development of new query methods; rather, it is expected that most queries will incorporate methods scripts from a local collection of such scripts, or that they will use one of a set of widely available ‘standard’ methods.

3.3.3 Query processing

The encoded query is passed by the query manager of the node at which it originated to a known query broker. This broker then passes the QURC to a different query manager on the joint basis of

- the broker’s overall policy on query transferral
- forward information held by the broker on what queries are handled by different managers

4 The System Prototype

The goal is to prototype a system which puts the ideas expressed in the above sections into practice in a particularly relevant and timely fashion, by using the Web as the substrate on which the prototype lives and building the other necessary components as modules on top of this substrate. Functionality already present within the structure of the Web will therefore only be replaced when something extra is needed.

4.1 The Elements of the Web

The informational component of the World Wide Web consists essentially of 2 elements:

1. pages, which are HTML documents
2. links connecting pages, which are URLs.

In addition to these essential technological components, which form the base technology, there is a set of extant proposals for technology extensions that will coexist with these elements. Examples of these are URNs (Universal Resource Names) and URCs (Universal Resource Characteristics). Despite the similarity in their names, these objects have rather different reasons for existence. URNs are supposed to be stable and long-lived, as well as unique. (The current thinking is that this requires URNs to be allocated by some authority scheme - perhaps one similar to that used for the Domain Naming Scheme - but this point has not really been finalised.) URCs on the other hand are *de facto* representations for meta-information describing (and containing) URNs, URLs and other attributes that can logically be associated with HTML resources. In a way, a URC can be thought of as an object wrapper for URNs and URLs, but even more importantly it is an abstraction from these.

The obvious idea that arises from thinking about URCs is to build a meta-information system using URCs as the meta-information medium, the Web as the resource medium, and existing browser technology for the retrieval of URLs, all integrated inside a layered system like a version of that shown in Figure 2.1.

Before such a system can be prototyped, there are a number of issues that need to be resolved: these will now be discussed.

The main benefit of meta-information systems is the potential speed and accuracy of searching. To take advantage of this, one needs to define classes of search functions looking something like [MADSEN94]

Match[Paradigm-URC, Actual-URC] = function(URC-Fields)

with

Match:URCxURC -> result-type

It is crucial to notice that extensible, or even request-defined matching is possible using trusted-environment scripting technology such as is presently being constructed within the auspices of ANSA Phase III Task B3.

The functional sequence required for querying within this framework must be something like

1. construct proto-URC as the paradigm against which candidate URCs are to be tested
2. pass proto-URC to search engine, or search agent if appropriate
3. searcher retrieves matching URCs and ranks them
4. searcher passes URNs from highest ranked URCs to URN->URL resolution service
5. resolution service trades URNs back and forth until appropriate URLs can be identified[1], then returns these URLs to the search agent
6. the search agent passes the returned URLs to its Web browser service and retrieves the resources
7. the returned resource is evaluated, and the evaluation is wrapped into the proto-URC, which is then stored in the appropriate meta-information base.

4.2 Building the Prototype

In order to see how best to build the prototype of this system, it is necessary to keep in mind the goals of speed (the technology underlying the Web is still far from stable, so that meta-information systems need to be prototyped on a far more rapid timescale than that on which substantial changes can be expected) and extensibility (there is no point in starting over repeatedly from scratch). In the interests of these goals, it seems sensible also to reuse as much existing technology as possible in areas such as the Web browser and the trading service. For the latter we could draw upon Nigel Edwards' CORBA-IDL stub compiler [EDWARDS95], although even this is not essential to the first versions, which can be built around a primitive scripted interface to the Matchmaker Service [GIRLING94].

At the next stage of development, when the CORBA-IDL stub compiler can be used as an ORB interface for trading purposes, we will probably need to develop a framework for delivering every component to the trader inside an appropriate CORBA object wrapper. It will also be necessary to ensure that early development stages do not close off too many future enhancement strategies: generally it will be necessary to ensure that additional services can be introduced in a modular fashion by inserting them between existing interface slices.

A degree of ambiguity exists as to how meta-information itself is stored and handled. In the interests of clarity, it would be best to keep meta-information and Web resources totally separate, but this would require construction of an appropriate storage engine early on. Accordingly, it has been suggested that (at the prototype stage at least), the meta-information, encoded as URCs, be

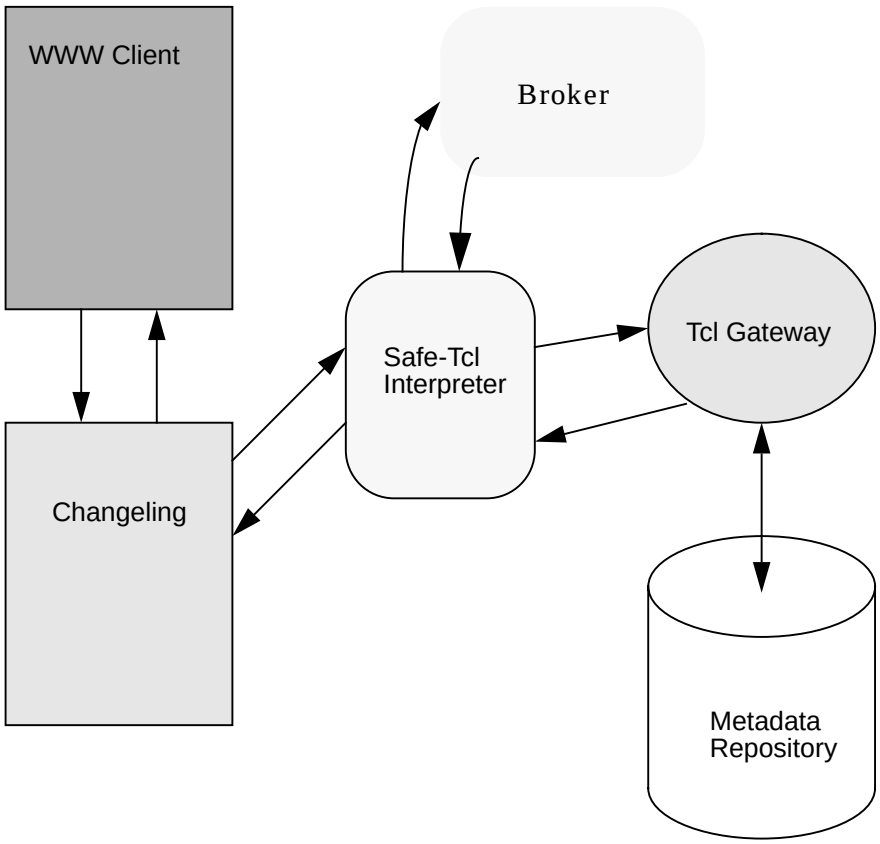
1. It may in fact be better to trade using parts of the meta-information contained within the returned URCs.

folded into HTML documents so that the meta-information management can be loaded onto a script-extensible Web server of the type presently being developed [MCCLENAGHAN95].

This is exactly the approach that has been adopted for a first cut of the code, with the benefit that user-specified search methods can also be defined in Tcl. These methods are then passed to the server as part of the query, and are run inside a Safe-Tcl environment with security parameters specified by the server according to the safety policy it operates.

The components of the prototype are shown in Figure 2.1.

Figure 4.1: The Components of the Prototype System



The prototype system broken down into its components. The central component is the Changeling Webserver which has been re-used to act as the query interpreter and interface to the local URC repository in addition to its webserving capabilities. Note that at this stage of development, the query script/agent is responsible for wrapping the retrieved URCs into HTML. The Broker cmponent stores forward information relevant to other webeservers or their metadata repositories.

Treating these components in turn:

4.2.1 WWW Client

Any client that supports the HTNL Forms interface can be used. Our experiments have mainly used NCSA Mosaic. All complications due to the

nature of the encapsulation of the metadata as URCs have been hidden within the forms interface.

4.2.2 Changeling

The Changeling extensible webserver is described in detail in [MCCLENAGHAN95]. The only modifications necessary for Changeling to support URC manipulation were those required to interface the metadata repository (since this exact location is, and should remain, invisible to the client). The extensibility of the metadata searching in the form of QURCs is supported by using the exstensibility of the HTTP methods already incorporated into Changeling.

4.2.3 URC Repository

Initially, the URC repository was modelled as a flat-file database using the Unix filesystem. This primitive model suffices only for local experiments where all URcs are held in a single repository.

The present plan is to use the MatchMaker as the appropriate model for the URC repository, and the requisite gateways have been written to allow access from Tcl scripts. The inclusion of the MatchMaker is obviously the first step towards implementing trading. The idea is that other MatchMaker modules can be used to store forward informaton for a variety of URC repositories. This allows a webserver whose query of its ow repository has failed to pass the query up to a query trading service for further processing.

4.2.4 Tcl Gateway

The Tcl gateway was constructed to allow easy access to the MatchMaker from Safe-Tcl. Since such gateways are simple and lightweight, it would have been an equally simple matter to write gateways for any desired scripting language. tcl was chosen for the reasons already mentioned, but it is hoped to extend the functionality of the system to include coping with J ava scripts in the future.

4.2.5 Safe-Tcl Interpreter

The experiments used a standard Safe-Tcl interpreter built using the standard Tcl/Tk toolkit.

4.2.6 Broker

This component is planned for future experiments on the following main fronts:

1. exploring scalability of URC matching/searching using different trading policies
2. testing the effectiveness of commercial approaches to metadata handling, such as selling resource access on the basis of metadata distribution
3. working with mobile agents that can be migrated from webserver to webserver, storing the state of their metadata searches internally.
4. mapping activities using agents that can evaluate metadata directly and autonomously [MADSEN95]

4.3 Future Developments

The further development of the ideas based on metadata as described in this document is continuing. In particular, the experiments that are listed under the description of the broker above will be performed as soon as the technology components become available. It is also hoped that appropriate IETF standards will become available on a short timescale. This will facilitate testing of and experimentation with these ideas over a larger scale.

References

[DANIEL]

Daniel, R.E. & Mealling, M., *URC Scenarios and Requirements*; <URL:http://www.acl.lanl.gov/URI/URCspec/>.

[EDWARDS95]

Edwards, N., *A Stub Compiler for CGI and HTTP: The Programmer's Guide*; **APM.1465**, APM Ltd., Cambridge U.K., May 1995.

[GIRLING94]

Girling, G. & Beasley, M., *The Property Repository*; **APM.1384**, APM Ltd., Cambridge U.K., December 1994.

[MADSEN94]

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

[MADSEN95]

Madsen, M., *Agents for Knowledge Resource Mapping in the World-Wide Web*; **APM.1473**, APM Ltd., Cambridge U.K., May 1995. In Working Notes of the 1995 CKBS-SIG Meeting on Intelligent Agents and the Next Information Revolution, edited by Michael Wooldridge and Michael Fisher.

[MCCLENAGHAN95]

McClenaghan, A., *The Changeling Web Server*; **APM.1453**, APM Ltd., Cambridge U.K., April 1995.