The Aquarelle Resource Discovery System

Alain Michard[a], Vassilis Christophides[b], Michel Scholl[c,a], Mike Stapleton[d],
Dale Sutcliffe[e], Anne-Marie Vercoustre[a]

[a] INRIA - BP 105 - F-78153 Le Chesnay Cedex.

[b] FORTH-ICS, P.O. Box 1385, GR 711 10, Heraklion, Crete, Greece

[c] CNAM, 292 rue St Martin, F-75141 Paris Cedex 03

[d] System Simulation Ltd., 250M Bedford Chambers, WC2E 8HA, London, UK.

[e] CLRC-RAL, Chilton Didcot, OX1 10QX OXON, UK

Abstract

Aquarelle is a three-year project supported by the Telematics Applications Programme of the European Union, aiming at designing a resource discovery system on the Internet, applied to cultural heritage documentation. The system relies on the Z39.50 protocol to support access to heterogeneous databases, including SGML document repositories. Its most original features are direct linking from SGML documents to database records, an advanced link management facility, and query broadcasting to dynamically selected databases

***Keywords*:** resource discovery; Z39.50; link management; heterogeneous databases; cultural heritage.

.

1. Objectives

Many types of resources are made available through the Internet: structured documents (e.g. SGML documents), textual documents with a relatively weak structure (e.g. html pages), unstructured texts (e.g. postscript or pdf files, character files), database records, and various kinds of files containing images, sounds, graphics, etc., which can be considered as unstructured as far as information retrieval is concerned.

A *resource discovery system* (RDS) is a facility offered to end-users to help them to retrieve the resources which are relevant to their current needs. In our view, a RDS should -as far as possible- have the following features:

- It should hide the distribution of data from the end user: the Internet (or a sub-part of it) should appear to the user as a global information repository.
- It should be able to retrieve structured as well as unstructured information. Information structure, whenever available, should be exploited to allow more accurate retrieval.
- It should process information which is organised and indexed to support a retrieval process (e.g. databases), as well as information which is not authored with the objective of facilitating retrieval (e.g. HTML pages on the Web).
- It should support heterogeneity of information structure, including heterogeneity of database schemas and of SGML Document Type Definitions (DTD), and this heterogeneity should ideally be hidden to the end-user.
- It should support heterogeneity of information sources: HTTP servers, various database management systems, and classical information retrieval systems (such as library OPACs) should be considered as primary content sources.
- It should support multilinguality of textual information.

- It should not overburden network resources by transmitting entire documents rather than index data or content labels.
- It should guarantee that, when some resource is identified and proposed to the user as a result of a query, the resource effectively exists and can be downloaded, i.e. the RDS should not propose dangling links.
- It should support some query refinement facility.

Our objective is to design a RDS, taking as a testbed the implementation of a distributed documentation system on the cultural heritage. Taking into account the characteristics of this application domain, we have decided to relax some of the desirable features mentioned before: the Aquarelle RDS is not scalable to the whole Internet, and we will consider as primary information either *databases,* or collections of *SGML documents*. We have not tried to support direct retrieval of unstructured information e.g. html, postscript or pdf files, unless they have been previously indexed in a database.

2. The Application Domain

Aquarelle aims at designing an information system enabling professionals to share across European borders cultural information related to paintings, sculptures, historical sites and monuments, musical instruments and furniture. The Web is more and more used by museums and cultural organisations for broadcasting cultural contents (see for instance the sites of the Louvre [1] or Uffizi [2]), as well as for exchanging cultural information at an international level. Current Web technology provides the necessary standardisation and global communication infrastructure, but existing tools suffer from several limitations: the general Web lacks a powerful RDS adapted to professional needs, and the HTML document model is much too weak for representing the rich structure of complex collections of digital documents.

Museum curators, urban planners, commercial publishers and researchers should be able to collect information relevant to their needs or interests notwithstanding the information location and organisation. In addition, each author of a given information component should be able to link directly a part of his/her own creation to another information asset created and updated by another author. Linking, annotating and commenting on relevant pieces of information belonging to different sources will bring much more than simple access to existing information: it will add value to the information content itself. The overall Aquarelle architecture is designed to relieve users from the cumbersome manual task of maintaining cross-references as well as to support the high precision required in referencing and retrieval.

In order to develop such hypermedia networks of multimedia documents, Aquarelle relies on two main sources of cultural information: existing primary material, called *archive data,* such as records, drawings, maps or text bases provided by the different cultural organisations (museums, galleries, etc.), and secondary material, referred to as *folders*, in the form of SGML documents, describing, commenting on and referring to archive data, as well as adding new information. Folders, in the Aquarelle sense, are considered as containers gathering a structured collection of specific information elements (archive data), which can be semantically linked together (intra or inter-folder references). The following are examples of folders :

- folders on architectural sites, monuments or objects, etc., such as those actually produced by the "Inventaire General" of the French Ministry of Culture [3];
- folders prepared by museum curators as supporting documentation for an exhibition e.g. Treasures from Mount Athos [4]
- folders created by a commercial publisher to edit a city guide.

In order to create a folder, users can start to retrieve, via queries on archive and folder servers, cultural heritage information related to their study or research interest. Furthermore, they can browse through the retrieved folder's structure and hyperlinks to identify particular objects of interest. Then the user can insert into the new folder references to the relevant objects, via an SGML editor. Users will "cycle" between editing, retrieval and browsing until they achieve a satisfactory product [5]. Therefore, queries on data available in archive and folder servers play a central role within the Aquarelle information discovery system.

Since there has been little coordination between the cultural organisations in setting up archive servers, existing primary material is currently available in various forms and structures (records, texts, images, drawings, databases, etc.), and it is managed by quite different platforms and systems : Data Base Management Systems (DBMS), Information Retrieval Systems (IRS), Knowledge Base Systems (KBS). In addition, folder texts and their references to primary material should conform to the folder SGML structure defined by a Document Type Definition (DTD). Aquarelle is designed to support an open number of DTDs according to the user needs. For the first evaluation phase, the project adopted the CIMI CHIO DTD (Cultural Heritage Information On-Line [6]) and that of the "Classeur d'Inventaire" (CI) from the French Ministry of Culture [3]. Other DTDs conforming to the emerging XML standard will be considered in a later phase of the project.

Aquarelle does not impose a data schema on the primary material, which ranges from strongly-structured (in record-oriented relational bases or graph-oriented object bases), to semi-structured, where the structure is looser, or irregular, or implicit (in SGML or bibliographic bases), and unstructured raw data (images or drawings).

Querying these heterogeneous and multilingual data sources, in a transparent way for end-users, is the central issue of the Aquarelle project.

3. Relationship to other work

A number of tools are available to facilitate discovery of resources. The most widely used fall into three categories: *search-services,* which are based on centralised automatic full-text indexing of textual files, *content-routing systems,* which are based on some content labelling to assist the gathering of information from content-servers without downloading the entire documents, and *subject-based directories* based on some manual classification of a collection of digital resources.

Search services (e.g. Altavista [7], Lycos [8], Excite [9]) are very well-known and are used daily by a very large public. They are based on centralised automatic full-text indexing of textual files. Their common drawback is that, to index the Web or some part of it, they must first fetch all the documents from the primary content-servers, hence overloading the Internet infrastructure. Their common limitation is first that of "blind" full-text indexing, leading to low recall ratio (because synonyms are not handled) and high noise ratio (due to homonyms, and terms considered as "out of context"), and second their inability to reference information stored in databases and made available through the automatic generation of HTML pages.

Subject-based directories (e.g. Yahoo [10]) try to alleviate to the limitations of full-text indexing by adding some manual classification of the collection of digital resources. The obvious limitation of this approach is the human cost when the digital collection becomes large and/or multilingual.

Content-routing systems gather information about other servers and provide some form of query-based mechanism for users to find out about servers relevant to their query. They use compact descriptions of content-servers, called content-labels, which are collected from the primary content-servers and stored on the RDS to process user queries and determine which are the relevant content-servers for a given query.

One of the first implemented content-routing system was Harvest [11]. In Harvest, the broker plays the role of a content-router, gathering simple content-labels in the form of SOIF (Structured Object Interchange Format) objects.

A more sophisticated content-routing system is Discover [12]. It offers a single point of access to over 500 WAIS servers, providing two key services: query refinement and query routing. Query refinement helps a user to improve a query fragment to describe his interests more precisely. Once a query has been refined and describes a manageable result set, query routing automatically forwards the query to the WAIS servers that contain relevant documents. Abbreviated descriptions of WAIS sites, the content-labels, are used by the query refinement and query routing algorithms. Discover did not support heterogeneity of primary content-servers, which were restricted to WAIS. Aquarelle can be considered as a content-routing system which can cope with more heterogeneity of primary databases.

Agent-based Information Filtering and Gathering Systems (e.g. [13][14][15][16]) rely on more or less "intelligent" programmes to assist the user in discovering and/or filtering distributed information. Agents exploit the interaction history of one or several users as made available through hotlists, browsing history, past queries, etc. They embed some machine-learning techniques to increase their competence by adapting to the user's interests, which may change over time, while at the same time exploring new domains that may be of interest to the user. Using this background information on the user's interests and habits, the system autonomously collects related documents and URLs from search services. Agent-based systems do not fetch and index directly the documents on the Net, which is the role of the search services. They try to help the user by automatically querying these search services and building a personalised index matching the user's interests.

Information Integration Systems are a new generation of RDS currently emerging [17]. The most advanced of them are based on *mediators* [18][19]. A mediator embeds the knowledge that is necessary for processing a specific type of information. In a sense, a mediator is a logical view of the data found in one or more sources. Data do not exist at the mediator level, but one may query the mediator as if it stored data; it is the role of the mediator to access its sources and find the answer to the query.

Several mediator-based architectures have been proposed, e.g. TSIMMIS [20] from Stanford University, and the Information Manifold project from AT&T [21]. Other Information Integration projects are variants of the above mediator-based architectures: HERMES [22], SIMS [23][24], DISCO [25] and the Internet Softbots [26][27] are examples of such variants.

One related fundamental issue in the integration of heterogeneous information systems, is the modeling and querying of data spread across the information sources to be integrated. [28] presents the advantages of a semi-structured view of data to be queried and integrated, and gives a comprehensive state of the art of the area. OEM/LOREL [29] and CPL [30] are the leading projects following this approach. While their models rely on a blurring between data and its structure, POQL [31][32] is based on an incomplete knowledge of the underlying strongly typed schema. Possible long term extensions to Aquarelle fucntionality based on such methodologies and architectures are discussed in [33].

4. Methodology

Aquarelle aims at providing an information retrieval service for searching across different database systems with different data architectures. It does this by presenting the user with a common vocabulary, including a set of access points for the purpose of phrasing a query. The Aquarelle cultural partners (content-producers) use over 300 fields to store the data in their respective databases [34]. Individual databases may use over one hundred different ones to store the information at the required level of detail. Due to the specific requirements of the respective institutions there is little commonality at this level. Searching these databases using the native fields as access points would provide as high a precision in the search as possible, but may lead to frustratingly low recall if the user is not familiar with the dataset. Our choice is to provide higher level access points for specifying queries. This approach will improve the recall at the cost of reduced precision. The common set of access points are mapped to the target datasets to perform the actual query.

Interaction between a user and individual databases is mediated by an access server. Apart from providing various services that will be mentioned later, the first role of this access server is to offer to the user a query interface structured along this set of common access points. This set appears as a virtual data structure in terms of which any query can be expressed. It can be viewed as a "Universal Relation" shared by clients and servers, which can be queried, and which must be mapped onto each real archive or folder data structure The access server transmits the query to the relevant content servers, where it is processed according to the specific mapping that is defined between the access-points and the local data structure implemented in the database.

Data servers may return results in two formats: "Brief", in which hits are listed in a very compact format using only two or three fields of the retrieved records, and "Full", in which full records are sent back. It is the role of the access server to process result sets and present them in a readable format to the user. After having received the "Brief" result set, the user may either select some of the hits and obtain the corresponding full records, or refine his query.

4.1 The AQLCIMI Z39.50 Application profile

Communication between the access server and the Aquarelle Data Servers is done by means of the ANSI/NISO standard Z39.50, adopted as international standard ISO 23950 [35]. Aquarelle uses Z39.50 in the context of the Internet environment. Specifications for the use of Z39.50 over TCP/IP are provided in RFC 1729 [36]. The Aquarelle profile specifies the subset of Z39.50 features, options, and parameters that the access server may use in a dialogue. Notably, it specifies the units of information that can be queried, that is the access points, which in the Z39.50 terminology are called the "Use Attributes".

The Aquarelle Z39.50 profile [37] is based on Draft version 3 of the CIMI profile [38], a companion profile to the Digital Collections profile [39]. The CIMI profile designed within the CHIO Project [6] is itself under revision. However, project CHIO was seen to be of particular relevance, as it aimed to support a similar, though narrower constituent community to Aquarelle. Furthermore, as well as identifying the access points, it defined them as both Z39.50 USE attributes and SGML tags. This provided input into the specification of the facilities of

Aquarelle Folder Servers as well as Archive Servers. The main difference was that Project CHIO addressed primarily information related to museum objects, while Aquarelle addresses information relating to sites and buildings as well. A subsidiary goal is interoperability of Aquarelle clients and CIMI servers and vice versa. For this reason, the Aquarelle Profile is based on the CIMI Profile, where possible, but makes fewer requirements in some areas, and is supplemented by extensions to support Aquarelle-specific functionality. Most of the extensions fall into two categories: first, support for the Aquarelle architecture (with the Access Server and emphasis on SGML documents); and second, the extension from CIMI's emphasis on museum information to include more diverse forms of cultural heritage, such as architecture. Where the Aquarelle Profile's extensions to the CIMI Profile have wider application outside the bounds of the Aquarelle project, it is hoped that they will be absorbed back into a future version of the CIMI Profile, so that there will eventually be a single profile shared by both projects.

Choosing the Z39.50 protocol implies that our underlying information model is "flat-records-oriented" as is generally the case in classical IRS: the concept of "record" or document in the traditional IRS view is intrinsically "flat", compared to the view of records or objects in DBMS (or of document fragments in modern SGML-based IRS). Finally, retrieval (using attribute sets) of relevant records and presentation of results are two distinct actions in IRS, while in DBMS the user must specify the required fields (*'select'* clause in SQL) together with the filtering condition (*'where'* clause in SQL). It is worth also noticing that, unlike DBMS, the structure of returned records is fixed (i.e., query independent) as it is (pre-)defined in the Z39.50 Profile.

These semantical differences in data representation and access are the main obstacles to issuing structured queries and retrieve complex data via the Z39.50 protocol. The query capabilities of the Z39.50 wrapped data sources are limited to a) Boolean filters and b) projection on fixed attributes; they do not allow associative access (joins). We shall discuss in section 7 some possible ways through which these limitations could be bypassed in future versions of the system.

4.2 Surrogate records

Current developments in this area, such as the Dublin Core project [40], are described in terms of defining metadata records which can act as a surrogate for the primary object or digital record. The architecture of some data services is based on holding surrogate records for the primary data records centrally and searching those rather than the primary data, for example SCRAN [41], ADAM [42], ELISE [43] and VAN EYCK [44]. The Aquarelle Access Server does not hold surrogate records at the object level. It may hold surrogate records describing complete archives for the purpose of providing finding aids and other directory services. However, the set of access points can be seen as defining a virtual record which can act as a surrogate for the primary data and can be queried as if it held the data. Archive Servers can be implemented using either a gateway or a dedicated server holding surrogate records. Hence data services based on holding surrogate records, such as those mentioned, fit easily into the Aquarelle architecture, where they act as Archive Servers.

5. Architecture and functionality of Aquarelle

The architecture of the Aquarelle Information System is shown in figure 1. At the logical level, it is composed of:

- Two user applications which will support all interaction with the end-user. The first one is an "off-the-shelf" Web browser (e.g. Netscape 3 or later), and the second one is a SGML editor and browser, used as a "helper application" of the Web browser, with some added specific functionality for publishing SGML documents on the system.
- An access server, which is offered to a user as the single entry-point to the Aquarelle network.
- An arbitrary number[1] of "archive databases", which may be in practice supported by various database management systems, information retrieval systems, or knowledge-base systems.
- An arbitrary number[1] of "folder servers", offering access to SGML documents.

As far as information structure is concerned, the major difference between folder databases and archive databases is that folders may embed hypertext links pointing either to other folders or to archive records, and that archive

[1] Typically between 10 and 200. The architecture is not designed to be scalable to thousands of data-servers connected to a single access-server.

records cannot embed hypertext links. In other words, browsing is supported only from SGML folders, and an archive record may only appear as a terminal leaf in the hypertext graph.

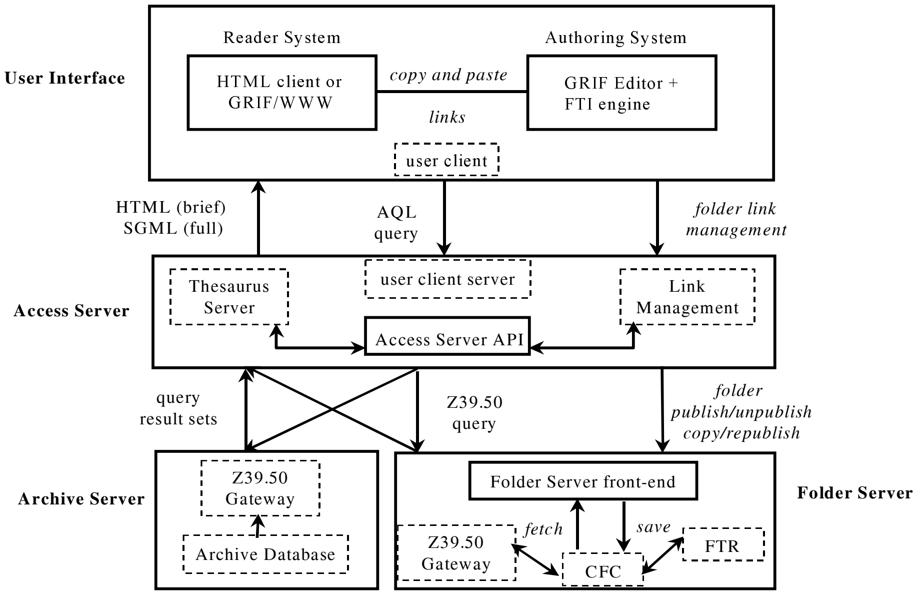


Fig. 1. The Aquarelle Architecture

5.1 The access-server

At the heart of the Aquarelle system is the access server , providing connections between the user clients and the archive and folder servers. It controls access to the Aquarelle system through the user management functions which include the storage and manipulation of user profiles. It supports the services provided by the user client, namely resource discovery, query handling, result management, folder publication, and one-to-one connections with servers, through specific functions. It provides a uniform interface to archive and folder servers based on the search and retrieval protocol Z39.50. It also provides an interface with a thesaurus browser to assist users in selecting query terms. Finally the consistency of hyperlinks in folders is guaranteed by the Aquarelle link management module. The access server embeds the following components:

5.1.1 The user-client server

The user-client server is made up of a Web server and of a set of CGI programs (java applications) that process the user requests, invoke the correspondent functionality in the access server, and encode the returned data before sending them to the Web browser. The Graphical User Interface (GUI) is a set of static and dynamic HTML pages; the static pages are accessed directly through the Web server, the dynamic pages are generated on the fly by the CGI programs. To prepare a query, the user is presented with a set of HTML forms. The form is submitted by HTTP and interpreted by CGI scripts in the User Client module. This converts the query to the Aquarelle Query Language (AQL), described in section 5.2, and passes it to the User Session.

5.1.2 The Z39.50 client.

The User Session has the opportunity to modify the query: for instance, to apply various terminology resources to translate or expand the terms. The modified query is still expressed in AQL and passed to the ZClient. The ZClient converts the query from AQL to Z39.50, using the Aquarelle profile, and broadcasts it to the currently selected data servers. Each data server interprets the native protocol and responds to the ZClient with a GRS-1 record. The GRS-1 record can contain structured data or SGML documents composed to one of the supported

DTDs described in section 5.3. The ZClient collates the responses and encodes them as SGML, if necessary, and returns them to the User Session.

5.1.3 The link-management sub-system.

The system architecture distinguishes between workspace links, which can be freely created, cut and pasted by folder authors, and Aquarelle links, which are centrally managed URIs, stored in a link database. It is the role of the link-management sub-system firstly, to transform workspace links into Aquarelle links when folders are published (that is when they are made available to querying and browsing on a folder server), and, secondly, to resolve the link when a user activates a link traversal.

An additional requirement for the link-management sub-system was to provide users with permanent identifiers for retrieved archive records. The permanent id or retrieval key is provided by an archive server to point at an item that has been found in an archive as a result of a query. When the result of the query is first made available, each item can be identified by an index within a result set, which is itself identified by a result set id. However, the lifetime of the result set id cannot be guaranteed, and so the retrieval key is necessary so that the item can be pointed at for longer than an Aquarelle session. The form of the retrieval key is determined by the archive server such that a subsequent query, on the access point *access$local_number* (use=12) with the value of the retrieval key, will return the full form of the item that is pointed at. The retrieval key is returned to the access server, where it is converted into a workspace link for return to the user client. The workspace link can be inserted into a folder, that is being created (or edited), to point at the item found by the query.

Syntax of workspace and Aquarelle links, as well as the process for handling them are described in [45] and [46].

5.1.4 The thesaurus server.

Most archive databases are filled in using standardised terminology resources, such as authority lists (e.g. list of artist names, or of geographic names), or thesauri (e.g. the Art and Architecture Thesaurus [47]). These terminology resources are, of course, in various languages and it is important to help users to select adequate query terms especially when they intend to query databases in other languages than their mother one. From his client application, the user is able to select a thesaurus from the list of the available thesauri, and to navigate through the thesaurus hierarchies and browse the selected terms. The user can view the top terms of a thesaurus (root terms) and the terms connected to a given term through broader-term and narrower-term relations and obtain definition of each term (scope note). For the selected term, all ancestors and all children (limited to two levels) are displayed. Each term having children (more specific terms) is an anchor that allows browsing recursively.

Besides thesaurus browsing, the system supports searching of specific concepts. Term matching and normalisation are meant to help the user with additional entry points in the thesaurus. Term matching identifies terms on the basis of their scope-note containing a given word or sub-string. Term normalisation retrieves in a thesaurus the canonical form of a term from a non-canonical form input by the user.

5.1.5 The access-server kernel.

The access-server "kernel" is basically a session-manager, handling user-sessions and ensuring data transmission between the various components mentioned before. It also manages users' profiles and access rights. It stores the metadata describing the content of the various data sources.

5.2 The Aquarelle Query Language

The Aquarelle Query Language (or AQL) is the language that is used in the Aquarelle system for passing queries between the user client and the access server, and between the user session and the Z39.50 client (within the Access Server). Aquarelle users typically express queries by filling in terms and selecting options on an HTML form. The completed form is then submitted to the User Client module, which compiles the query into AQL and submits it to the Access Server for transmission to the target data servers. AQL is based on CCL (as defined by ISO 8777 and its equivalent Z39.58). However, CCL and Z39.50 have different models of queries, which means that some translation or adaptation must take place before CCL can be used to prepare Z39.50 queries. Further, it is necessary to specify which option is used where ISO 8777 and Z39.50 differ. CCL was chosen as the basis for AQL as:

- It provides a similar expressive power to typical Z39.50 queries.
- If it became desirable to add a command-line query language to the Aquarelle system, using a known query language, like CCL, would mean that a solution was already available.
- AQL need not be visible to the user in normal use. However, using a protocol that can be expressed as plain text and read by people is a useful feature when integrating and testing the system.

The abstract query expressed in AQL consists of a set of query elements that are joined by Boolean operators (and, or, not) and brackets. All the Boolean operators are binary operators ('not' is defined as 'and-not'); they have equal priority and are evaluated left-to-right. Brackets may be used to specify a different order of evaluation. Each query element consists of just a term, i.e. a word, or of one or more qualifiers separated by commas, followed by an equals sign and a term. The term may consist of more than one word, in which case it should be enclosed in double quotes.

The qualifiers are translated into Z39.50 attributes from the Aquarelle profile. This is a one-to-one mapping. As a result, the qualifiers belong to one of seven types representing the seven attribute types in the attribute set of the Aquarelle profile, namely "access" representing the attribute use (acces-points), "relation" representing a string comparator, and four attributes types representing position, structure, truncation, completeness and authority. A maximum of one qualifier of each type is allowed in each query element. If there are more than one qualifier in a query element, one must be "access". More complex queries (such as looking for a term on multiple access points) require a Boolean combination of query elements. The qualifier name has two parts separated by a dollar sign, the first part representing the qualifier type and the second the qualifier value : if the qualifier name is "access", then the qualifier value is an attribute name, if it is "relation" then thequalifier value is a string comparator, if it is "trunc" then the value is a truncation specification, etc.

For example,

```
access$personal name=''Alberto Giacometti'' and access$title,
trunc$both=Man
```

where "both" means left and right truncation, would find artworks by Alberto Giacometti containing “Man” somewhere in their title.

It should be noted that AQL is based on a *flat* view of information corresponding to the set of access-points defined in the Z39.50 Profile. To illustrate this point, let's consider a more complex example. Usually a number of actions (creation, acquisition, donation, possible modifications, etc.) are associated with cultural objects, where an action is characterised by at least one actor and an event. An event is determined by a time and a place.

A user may want to retrieve the work of Alberto Giacometti made when he joined the Surrealists in 1930. This query can be expressed in AQL as follows:

```
access$personal name=''Alberto Giacometti'' and access$event=creation
and access$date=''1930:1940''
```

The user will receive as a response to this query a number of records satisfying the search conditions. But nothing guarantees that the retrieved cultural objects have, indeed, been created by Alberto Giacometti in the given period. In fact, Alberto Giacometti may have donated the retrieved painting or sculpture to a museum, or even restored it. Any logical association between actions, events, persons, dates and places has been lost. Flattening the information structure makes it possible to provide a uniform vision of heterogeneous data structures, at the cost of losing some of the information represented through the original structures. We will consider in section 7 some ways of limiting the resulting loss of information for the end-user.

5.3 Results formats

Results from queries are made available to the user-client as SGML documents conforming to one of the following DTDs:

- Aquarelle Results Format (ARF DTD). It is used to transmit documents in Brief Format from archive and folder databases, and documents in Full Format from archive databases.
- Folder Profiles (FP DTD). It defines the content and structure of metadata records describing SGML folders. Folder Profile content is automatically extracted from the folder itself, is stored as a separate document on the folder server and is used to support folder retrieval.

- Classeurs d'Inventaire (CI DTD). A DTD for folders documenting the architectural heritage and related objects, that is immovable heritage items, for which georeferencing is possible. It supports embedding a collection of folders, represented as typed links, into another folder with the geographic inclusion relation.
- CIMI DTD. A DTD defined in Project CHIO of the CIMI [6] to represent museum exhibition catalogues. It is based on the Text Encoding Initiative (TEI) DTD [48].

5.4 The archive servers

Archive Servers provide information about individual objects or sites. The model of an Archive Server is designed so that a museum collection management system, a photograph library catalogue, or a data service system could act as an Aquarelle Archive Server. They are expected to be able to return a record about each object or site held in the database. They follow the conventional information retrieval model for database access.

- Mapping from Z39.50 to the local host database

  As well as mapping the access points to the local data structures, the Archive Server maps the responses of the host database into the record structures of the Aquarelle Profile supported by the Archive Server. The 'Brief record' is precisely defined so that summary displays showing the results from different servers will be consistent. The 'Full record' allows more freedom for the database owner to specify which native fields they wish to make available and how they are mapped to the tag-set elements of the profile.

- Archive Server architectures

  To encourage the acceptability of the Aquarelle data service it was decided that it must be possible to implement an Archive Server without affecting how the data provider's primary data is stored. This can be achieved by both the server approaches described below. Different data providers may find one or other approach more attractive. Archive Server implementers are not restricted in how they implement the servers so long as the interface they present to the Aquarelle ZClient conforms to the Aquarelle Z39.50 Profile. Two approaches are represented in the project.

  The gateway approach provides the target capability by adding a gateway layer which converts dynamically between the Z39.50 protocol and the native query and retrieval facilities of the host database. The target gateway implements the relevant Z39.50 Profile, and will make use of local storage to hold result sets and other working data in the short term. It does not store significant amounts of indexes or primary data. It maps the host database's searching and retrieval capability on to the Z39.50 Profile as well as it can, rather than augmenting it in any fundamental way. The "underlying database" is accessible according to the mapping implemented by the Gateway. The gateway approach probably requires less local resources than a dedicated target.

  The dedicated target approach uses a separate database to provide the Z39.50 target capability. This target database may simply hold indexes, or it may hold sufficient data to be able to respond to the Z39.50 queries without reference to the primary database. The target database can implement the supported Z39.50 profiles as fully as required, providing search and retrieval capability not supported by the primary database. To implement this approach, an appropriate interface must be established between the target database and the primary database. Strategies for updating the target database can be periodic or continuous depending on the capabilities of the primary database. The target database acts as a proxy for the "underlying database" which is not directly accessible to Z39.50 clients. This approach allows content owners to "publish" a selection of their data to a dedicated target which may be operated outside a firewall.

  The first generation of Archive Servers described in [49][50][51] can all act as gateways to the primary data. The Mistral [50] and Index+ [51] systems can also act as dedicated servers holding surrogate records derived from the primary data.

- Registering archives

  An archive owner must register the archive with the Aquarelle service. In the first version, this is done by conventional means, and the entries in the Directory Services database describing the Archive are made by the Access Server administrators. Various approaches are under investigation to allow some or all of these details to be registered automatically. For instance, the use of the Z39.50 'Explain' service gives the potential for the Access Server to interrogate Archive Servers about the content of the archive. Currently, support for the Explain service is not widespread. At the time of writing, the Aquarelle and CIMI Z39.50 Profiles do not require support for EXPLAIN.

5.5 The folder servers

Complementing the information available from the Archive Servers, Aquarelle also provides access to "folders", i.e. SGML hypertext documents typically providing information relating to groups of objects. Folders may include links to other folders, and to object data known to the Aquarelle system. Aquarelle provides a unified interface for searching for and browsing folders in conjunction with object information.

Folder Servers are an original creation of the Aquarelle project. The storage model and functionality are described in [52]. They hold the primary data, the folder text, as an SGML document. A folder server stores for each SGML document a companion surrogate document called "Folder Profile" (note this use of the term "profile" is unrelated to the Z39.50 use). The folder-profile DTD provides for a simple set of descriptive fields, that can be searched directly in the same manner as object data held on Archive Servers. The folder itself is returned as an SGML document encapsulated in GRS-1 record syntax.

Folders can contain links to records held on Archive Servers. The Aquarelle Z39.50 Profile provides a mean for obtaining a unique identifier which can be used for this purpose. In many respects, Folder Servers and Archive Servers can be treated simply as data servers serving different types of content.

6. Information retrieval process description

Fig. 2 indicates how queries and responses are handled:

(a) The users, at their workstations, are presented with an HTML form. They can formulate their query by filling in fields on the form, where the access points are implied by the selection of fields. Alternatively, they can use a conventional query language, where the access points are given explicitly.

(b) The form is submitted by HTTP and interpreted by CGI scripts in the User Client module. This converts the query to the Aquarelle Query Language (AQL) and passes it to the User Session. AQL supports all the Aquarelle access points (see section 0).

(c) The User Session has the opportunity to modify the query: for instance, to apply various terminology resources to translate or expand the query terms. The modified query is still expressed in AQL and passed to the ZClient.

(d) The ZClient converts the query from AQL to Z39.50 using the Aquarelle profile, and broadcasts it to the currently selected data servers. The Aquarelle access points are expressed as Z39.50 USE attributes.

(e) The data server gateway interprets the Z39.50 query into the host database query language and submits it according to the local protocol. The access points are now mapped onto one or more host database fields.

(f) The host database responds to the data server gateway according to the local protocol. The data server interprets the native protocol and responds to the ZClient with a GRS-1 record. The GRS-1 record can contain structured data, or SGML documents composed for one of the supported DTDs.

(g) The ZClient collates the responses, encodes them as SGML if necessary, and returns them to the User Session.

(h) The User Session returns the SGML document to the User Client, which translates it into HTML and passes it to the User Workstation using HTTP.

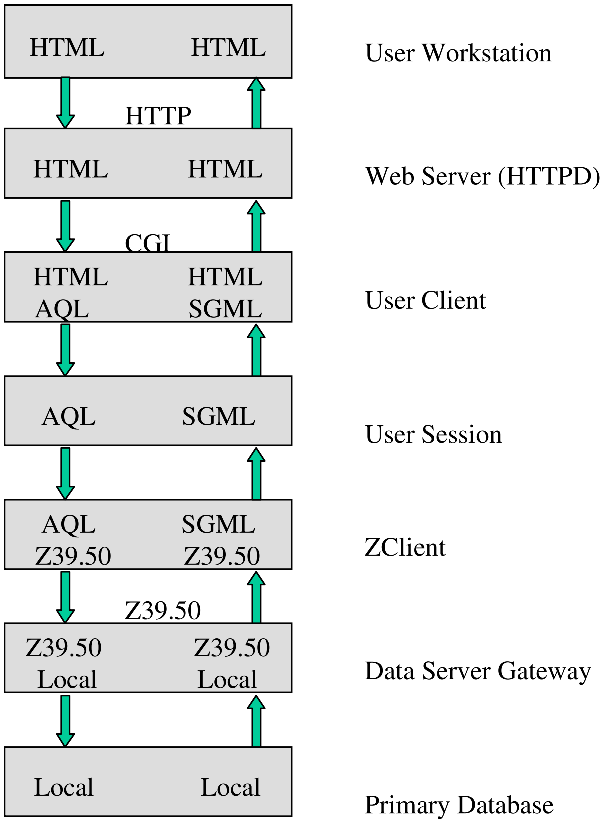


Fig. 2. Query and response handling

Each protocol and the conversion between them has to be specified so that the user query and the response are transmitted faithfully.

7. Open issues

Sticking to the Access-Point flat view of information, we discuss in this section how to improve (i) the functionality of the access-server, (ii) the user-interface, and (iii) access to the data servers. Fig. 3 illustrates some of these short-range improvements which are compatible with the current Aquarelle architecture.

7.1 Improving the access-server functionality

7.1.1 Improving data-servers registration

In the present version of the system, registration of a database on the access-server is done manually. The access-server manager must collect and store various data about each archive: identification of the database and of the server, description of content using one or several keywords denoting cultural domains, thesauri and authority lists used in the database, etc. This manual registration may appear as a limitation to the scalability of the system.

A possible improvement may be obtained through the use of the EXPLAIN feature of Z39.50. This facility may be used to manage a database with information, such as name, description, protection, cost of the databases which are available for search, or more specific data, such as access points details, schemas, record syntax, etc. Explain categories, stored as tables have to be specified in the search syntax. Categories of main interest for us are:

- TargetInfo for information about the target;
- DatabaseInfo for searching a specific database managed by the target;
- SchemaInfo describes the schema of a database. Databases with different structures could be managed by the same target;

- TermListInfo includes all descriptive information about the database access points which can be usefully exploited for querying.

A Z39.50 search operation can then specify the request for all databases managed by the target which receives the query : for example, a request such as ExplainCategory='DatabaseInfo' AND Availability='yes' gives as a result the references of all the databases managed by the target. The expression :

```
        ExplainCategory='TermListInfo' AND DatabaseName='!database name?'
```

would give as a result all the term lists associated with the databases managed by the target.

7.1.2 Better integration of metadata

Metadata can be classified into three categories : (i) information that can be derived directly from contents such as structure of the data, (ii) information that can be derived by system tools, such as size, format, etc., (iii) information that cannot be derived directly from data content and structure. For instance, a Subject description of a folder is situated at a more abstract level than its Title.

The discussion of "metadata for digital objects" was started by librarian communities trying to provide uniform semantic descriptions for accessing unstructured data (images or drawing bases), or semi-structured data (HTML pages where the SGML tags serve only for presentation purpose). In all cases, it is currently understood that schema (in the database jargon) is part of the metadata. In addition, the query capabilities of the data sources (i.e., support of boolean or SQL queries), as well as constraints on the source's contents (e.g., data available only for a specific time, place, etc.), is also part of the metadata. In the current design, access points represent metadata from all sources, i.e. structure as well as information that cannot directly be derived from the data contents. However, at the access server level, there is no way to decide whether a given source has a valid instance of a given access-point.

To enable data servers to manage their metadata and export them to the Access Server, the use of the Z39.50 Explain facility can be considered (see subsection 7.1.1). The metadata can then be used by the Access Server for filtering the most appropriate data sources to answer a user query. It can also be an important means for increasing knowledge of data sources at the user level, as a help to issue queries. Metadata should make it possible to present to the user a structured view of the global knowledge represented by the whole set of data-servers, based on clustering data servers which share valid access points and corresponding terminology authorities.

7.1.3 Automatic translation of query terms

When an end-user selects some query terms in a given language, and intends to broadcast his queries to databases which may be in different languages, it would be of great help if the system could automatically translate the selected terms to all the target languages, using multilingual thesauri whenever they exist for the relevant domain. Although an API has been specified and implemented for automatic term translation, it is not yet integrated in the current version of the system. In fact, as multilingual thesauri are presently very rare and have limited coverage, this functionality was not considered of very high priority. However, its integration may act as a strong encouragement to cultural organisations for the production of multilingual thesauri.

7.1.4 Distribution of mediation services

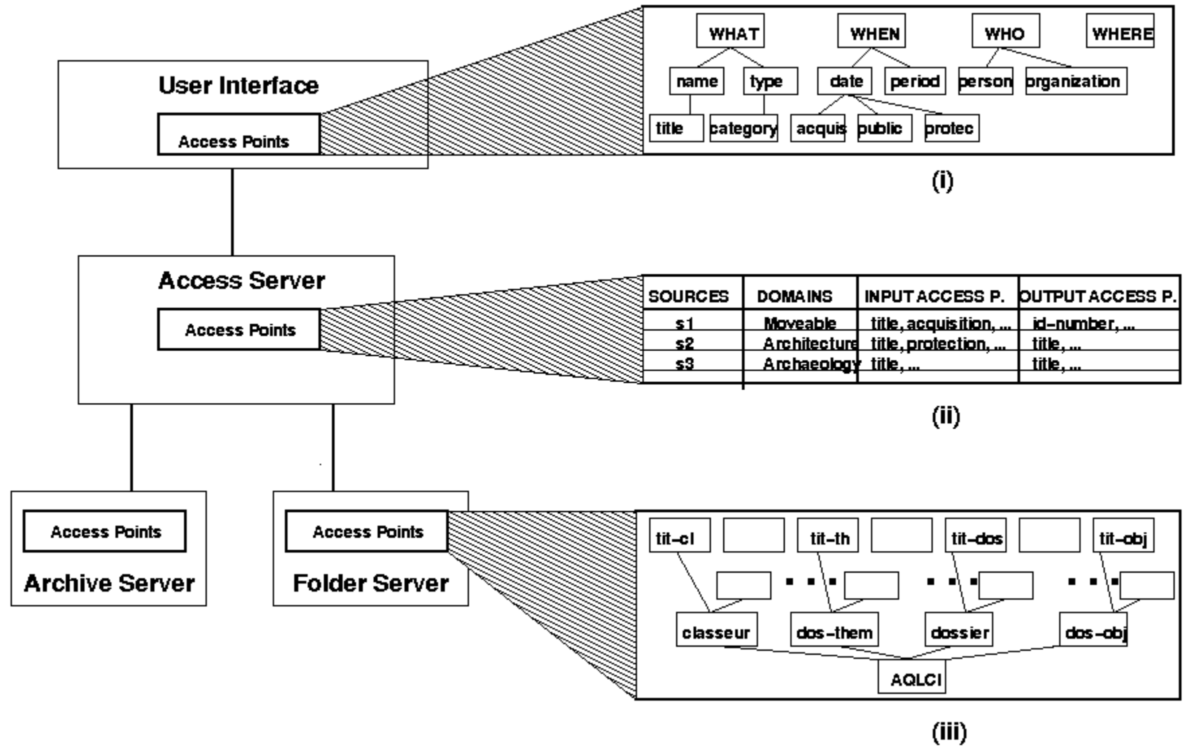
In the present version of the system, the access server is not distributed, but replicated. Each content database has to be registered on each of the access servers. For the sake of scalability, the access server should be distributed. This implies the need to define mechanisms for forwarding queries from one access server to another, which is a challenging issue.

7.2 Improving the user-interface.

More semantics about the data sources should be available to the end user. In particular, the user should have the possibility of browsing the metadata related to data sources. More knowledge about the metadata of data sources not only allows the user to refine his/her needs (and therefore improve precision), but also allows him/her to express queries exploiting the structure of the target information bases. In other words, if the user could obtain precise information on the mapping of the access points onto the various elements of the folders or archives structures, he would likely express much more precise queries than in the present situation where this mapping is hiden. This perspective may be exploited both at the user interface level to present the meta information of specific databases (see section 0) and at the Access Server level via a common database schema giving a global view of the available information.

In order to enrich the current user interface, conceptual modelling based on access points should be explored. Part (i) of figure 3 shows a simple example of conceptual modelling. This classification should help the user in his/her choice of the relevant access points and values to choose for his AQL queries.

Finally, it must be stressed that the user interface should depend not only on the data sources available, but also on the user perspective. This may imply sophisticated user-interface customisation, and various user-helps adapted to various levels and natures of knowledge of the data servers which the user wants to restrict his/her attention. Clearly, there is a need for a variety of user interface paradigms depending on the user expertise: naive user or curators, urban and regional planners, publishers and researchers, etc.



SOURCES	DOMAINS	INPUT ACCESS P.	OUTPUT ACCESS P.
s1	Moveable	title, acquisition, ...	id-number, ...
s2	Architecture	title, protection, ...	title, ...
s3	Archaeology	title, ...	title, ...

Fig. 3. Structuring the user interface

7.3 *Improving access to the data servers*

7.3.1 Structure-based queries to specific data sources

We have argued that knowledge of the actual data structure at the user level might help the user in query formulation (without modifying AQL). We assume here that some users might be interested in accessing specific archive or folder servers whose structure they want to discover, or of which they are aware (see figure 2, case iii). This is not a uniform access to heterogeneous data sources, but, on the contrary, an enrichment of the user interface for the user who wants a narrower and richer access to a more restricted range of servers.

Experience from existing Web interfaces to databases (see for instance JOCONDE [53] and MERIMEE [54]) assisting users to formulate structured queries tailored to specific cultural DBMS demonstrates that when the database structure is made visible at the user-interface level, users are able to issue complex queries fully exploiting all the characteristics of this structure. For instance, in a folder server with SGML documents describing architectural objects (i.e. CI DTD), one may issue queries like: "Find building photographs whose caption contains Place" or "Find the buildings whose bibliographic references all concern the Cognac Region". As we have seen in section 5.2, such queries cannot be expressed easily or at all with the current flat view of information supported by AQL.

Since we do not wish to modify the current Aquarelle architecture, these queries might be expressed using the native query language of the target system, and could be sent via the Z39.50 protocol as uninterpreted "strings" (i.e. Z39.50 type 0 queries). The same is also true for returned query results.

7.3.2 XML

With the emergence of the XML standard [55][56][57], it is likely that true XML browsers and editors will soon become widely available, hence making it unnecessary to transcode SGML results into HTML for displaying them. The ARF, FP and CI DTDs have been designed with this evolution in mind, and can be turned into XML DTDs very easily. This transformation seems much less feasible for the CIMI DTD which inherits of some of the advanced SGML features of the TEI DTD. A possible solution will be to send only well-formed documents, and to rely completely on a style-sheet mechanism, such as CSS [58], to display folders and result-sets.

Moreover, the proposal for a standardised Resource Description Framework [57] may make it possible to design more interoperable data structures for representing metadata than in the current version. This evolution may prove very helpful for supporting distribution of the access-server, as well as to permit querying folder-servers from other client applications than the Aquarelle access-server. This development will be considered for the next version of the Aquarelle system.

8. Acknowledgements

Aquarelle is a European Project in which 22 different organisations actively participate. In addition to the authors who have directly contributed to the present paper, we would like to acknowledge the highly valuable contribution of P. Bonhomme (Inria), C. Bounne (Grif s.a.), V. Bouthors (Bull), P. Carrara (CNR-ITIM), D. Dawson (Museum Documentation Association), M. Döerr (ICS-FORTH), F. Malézieux (Euroclid), A. Marrara (Finsiel), M. Padula (CNR-ITIM), S. Piperidis (ILSP), A. Rizk (Euroclid), L. Romary (Inria), O. Signore (CNR-CNUCE), M. Taylor (SSL), O. Toche (Ministère de la Culture), as well as the active participation of many other colleagues working in the same organisations.

References

[1] Bienvenue sur le serveur Louvre. http://mistral.culture.fr/louvre/

[2] The official Web Server of the Galleria degli Uffizi in Florence, http://www.uffizi.firenze.it/welcomeE.html

[3] Les dossiers de l'Inventaire Général, http://aquarelle.inria.fr/Inventaire

[4] Hellenic Ministry of Culture, Treasures of Mount Athos, http://www.culture.gr/2/21/218/index.html

[5] M. Doerr, V. Christophides and I. Fundulaki, The specialist seeks expert views - Managing digital folders in the Aquarelle project. *Museums and the Web 97*, pages 261-270, Pittsbourg, Pennsylvania US, Archives & Museum Informatics 1997.

[6] Consortium for the Computer Interchange of Museum Information (CIMI), Cultural Heritage Information Online (CHIO) Project: http://www.cimi.org/projects/chio.html

[7] http://www.altavista.digital.com/

[8] http://www.lycos.com/

[9] http://www.excite.com/

[10] http://search.yahoo.com/

[11] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber and M.F. Schwartz, The Harvest information discovery and access system. In *Proceedings of the Second International World Wide Web Conference*, Chicago, Illinois, Oct. 1994.

[12] M.A. Sheldon, A. Duda, R. Weiss and D.K. Gifford, Discover: a resource discovery system based on content routing. In *Proceedings of the Third International World Wide Web Conference*, Darmstadt, Germany, April 1995. Available at : http://www.igd.fhg.de/www/www95/proceedings/papers/82/html-files/discover.html

[13] Alexandros Moukas, Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. *Proceedings of the Conference on Practical Application of Intelligent Agents & Multi-Agent Technology*, London, 1996. Available at: http://moux.www.media.mit.edu/people/moux/papers/PAAM96.ps

[14] Marko Balabanovic, Yoav Shoham and Yeogirl Yun, *An Adaptive Agent for Automated Web Browsing.* Stanford Working-paper No SIDL-WP-1995-0023. http://www-diglib.stanford.edu/diglib/WP/PUBLIC/DOC55.ps

[15] William P. Birmingham, *An Agent-Based Architecture for Digital Libraries*. D-Lib Magazine, July 1995,

[16] Chen H. Chung, Y. Ramsey, M. Yang, C. Ma and P. Yen, Intelligent Agents on the Internet, *Proceedings of the 30th Annual Hawaii International Conference on System Sciences* (HICSS-30), Maui, Hawaii, January 7-10, 1997.

[17] J. Ullman, Information Integration using Logical Views. In *Proceedings of the 6th International Conference on Database Theory*, Delphi, Greece, January 1997.

[18] I3 Reference architecture, http://dc.isx.com/I3/html/briefs/I3brief.html

[19] G. Wiederhold, Mediators in the architecture of future Information Systems, IEEE Computer, pages 38-49, March 1992.

[20] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Irelandand, Y. Papakonstantinou, J. Ullman and J. Widom, The TSIMMIS Project: Integration of heterogeneous information sources. In *Transactions of the Information Processing Society of Japan (IPSJ)*, Tokyo, Japan, October 1994.

[21] T. Kirk, A. Levy, Y. Segiv, and D. Srivastava, The Information Manifold. *In AAAI Spring Symposium Series on Information Gathering from Heterogeneous Distributed Environments*, Stanford University, March 1995. http://www.isi.edu/sims/knoblock/sss95/proceedings.html

[22] V.S. Subrahmanian, Sibel Adali, Anne Brink, James J. Lu, Adil Rajput, Timothy J. Rogers, Robert Ross and Charles Ward, HERMES: A Heterogeneous Reasoning and Mediator System. http://www.cs.umd.edu/projects/hermes/overview/paper/index.html

[23] Yigal Arens, Craig A. Knoblock and Chun-Nan Hsu, Query Processing in the SIMS Information Mediator. *Advanced Planning Technology*, Austin Tate editor, AAAI Press, Menlo Park, CA, 1996.

[24] SIMS: An Information Mediator: http://www.isi.edu/sims/

[25] Anthony Tomasic, Louiqa Raschid and Patrick Valduriez, Scaling Heterogeneous Databases and the Design of Disco. In *Proceedings of International Conference on Distributed Computing Systems*, 1996, available from: **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**

[26] Oren Etzioni and Daniel Weld, A Softbot-Based Interface to the Internet, *Comm. of the ACM*, July 1994. available from http://www.cs.washington.edu/research/projects/softbots/www/publications.html

[27] Robert B. Doorenbos, Oren Etzioni and Daniel Weld, A scalable comparison-shopping agent for the World-Wide-Web, Agents'97 Conference Proceedings, ACM Publisher, 1997, available from http://www.cs.washington.edu/research/projects/softbots/www/publications.html

[28] S. Abiteboul, Querying semi-structured data, In *Proc. of the International Conf. on Database Theory*, pp. 1-18, Delphi, Greece, January 1997, LNCS 1186, Springer-Verlag.

[29] S. Abiteboul, D. Quass, J. McHugh, J. Windom and J. Wiener, The Lorel query language for semi-structured data. *Journal of Digital Libraries*, 1, April 1997, pp. 68-88.

[30] Peter Buneman, Susan Davidson and Dan Suciu, Programming Constructs for Unstructured Data, In *Proc. of the 1995 International Workshop on Database Programming Languages* (DBPL), Gubbio, Italy, Sept. 1995, Springer-Verlag, available at : http://www.springer.co.uk/ewic/workshops/DBPL5/

[31] V. Christophides, S. Abiteboul, S. Cluet and M. Scholl, From Structured Documents to Novel Query Facilities. *SIGMOD'94*, pp. 313-324, Minneapolis, Minnesota, USA , May 1994.

[32] S. Abiteboul, S. Cluet, V. Christophides, G. Moerkotte and J.Simeon, Querying Documents in Object Databases, *Journal of Digital Libraries*, 1, pp. 5-19, April 1997.

[33] Vassilis Christophides, Paola Carrara, Marco Padula, Michel Scholl and Anne-Marie Vercoustre, Querying Heterogeneous Data in the Aquarelle project. Project deliverable No D4.6, 20 Sept. 97.

[34] D. Dawson, Data Structures in Cultural Databases : A survey. Aquarelle Deliverable No D5.5, 28 Sept. 1996.

[35] National Information Standards Organization, ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. Bethesda, MD, NISO Press, 1995. Available at: http://lcweb.loc.gov/z3950/agency/document.html

[36] Clifford A. Lynch, RFC 1729, Using the Z39.50 Information Retrieval Protocol in the Internet Environment. 1994, available at: http://ds.internic.net/rfc/rfc1729.txt

[37] M. Stapleton and M. Taylor, Data structures in cultural databases and Aquarelle Z39.50 Application Profile. Aquarelle Deliverable No D5.6, 20 Oct. 1997.

[38] CIMI Profile Development Working-group, The CIMI Profile: Z39.50 Application Profile Specifications for use in project CHIO, 1996, Available at: http://lcweb.loc.gov/z3950/agency/profiles and ftp://ftp.cimi.org/pub/cimi/CIMI_Profile/

[39] Library of Congress, Z39.50 Profile for Access to Digital Collections, May 3 1996, http://lcweb.loc.gov/z3950/agency/profiles/collections.html

[40] Dublin Core Metadata Element Set: Reference Description. Revision of January 15 1997. Available at: http://purl.org/metadata/dublin_core_elements

[41] Scottish Cultural Resources Access Network (SCRAN), http://www.scran.ac.uk

[42] ADAM, The Information Gateway for Art, Design, Architecture and Media. http://www.adam.ac.uk

[43] ELISE, Electronic Library Image Server for Europe, E.C. DGXIII Project No1008, http://severn.dmu.ac.uk/elise/

[44] VAN EYCK, Visual Arts Network for the Exchange of Cultural Knowledge, E.C. DGXIII Project No1054, http://www.hart.bbk.ac.uk/van_eyck.html

[45] D. Sutcliffe, Querying Service: Technical description , Aquarelle Deliverable No D4.1, 21 Nov. 1997.

[46] E. Fras, Folder Publisher, Link Server and Directory Services, Aquarelle Deliverable No D4.3, 15 Sept. 1997.

[47] Getty Information Institute, Art & Architecture Thesaurus (AAT). 2nd ed. New York: Oxford University Press on behalf of the Getty Art History Information Program, 1994. Bibliographical references. ISBN 0-19-509155-8.

[48] ACH/ACL/ALLC (Association for Computers and the Humanities, Association for Computational Linguistics, Association for Literary and Linguistic Computing), Guidelines for Electronic Text Encoding and Interchange (TEI P3). Edited by C.M. Sperberg McQueen and Lou Burnard. Chicago: ACH/ACL/ALLC, April 8 1994. 2 volumes, xxvi + 1290 pages

[49] O. Signore and M. Loffredo, Z 39-50-SQL gateway : technical description , Aquarelle Deliverable No D5.1, 10 Oct. 1997.

[50] Vincent Bouthors, Jean-Yves Dupuis and Nhan Tran Huu, Z39.50 Gateway for Mistral and ARF DTD Specification , Aquarelle Deliverable No D5.2, 16 Sept. 1997

[51] M. Taylor and M. Stapleton, The Z 39-50 version of Index+ : technical description, Aquarelle Deliverable No D5.3, 8 Oct. 1997.

[52] Cyril Bounne, Vassilis Christophides, Martin Döerr, Eddy Fras, Irene Fundulaki, A. Kementsietsides and Y. Velegrakis, Aquarelle folder server and editor technical description. Aquarelle Deliverable No D6.4, 10 Oct. 1997.

[53] Ministère de la Culture, La base documentaire Joconde, http://www.culture.fr/cgi-bin/mistral/joconde

[54] Ministère de la Culture, La base documentaire Mérimée, http://www.culture.fr/cgi-bin/mistral/merimee

[55] Tim Bray, Jean Paoli and C. M. Sperberg-McQueen, editors. Extensible Markup Language (XML), W3C Working Draft 07-Aug-97, http://www.w3.org/TR/WD-xml-lang

[56] Tim Bray and Steve de Rose, editors. Extensible Markup Language (XML): Part 2. Linking, W3C Working Draft July-31-97, http://www.w3.org/TR/WD-xml-link

[57] Ora Lassila and Ralph R. Swick, editors. Resource Description Framework (RDF) - Model and Syntax. W3C Working Draft, Version 1, 02-Oct-1997, http://www.w3.org/TR/WD-rdf-syntax/

[58] Håkon Wium Lie and Bert Bos, Cascading Style Sheets, level 1 , W3C Recommendation 17 Dec 1996 , http://www.w3.org/TR/REC-CSS1