1. Introduction

This document specifies the Z39.50 profile used by Aquarelle software within the framework of the Aquarelle architecture described in [10].

Communication between the Aquarelle Access Server's ZClient (hereafter the "client") and the Aquarelle Data Servers (hereafter "servers") is by means of the ANSI/NISO standard Z39.50 (recently adopted as international standard ISO 23950), which specifies a protocol for searching in, and retrieving data from, databases on remote machines. The complete definition of Z39.50 can be found in the in ANSI/NISO standard [1]. Terms such as "attribute set", "tag set" and "APDU" are defined and exemplified in that document.

Aquarelle clients and servers use Z39.50 in the context of the Internet environment. Specifications for the use of Z39.50 over TCP/IP are provided in RFC 1729 [3].

This profile specifies the subset of Z39.50 features, options, and parameters needed to support Aquarelle's requirements for searching and retrieval of cultural heritage information. It addresses Z39.50 search and retrieval but does not specify user interface requirements, nor the internal structure of databases that contain the digital information objects.

Appendix A of this document describes the Aquarelle attribute set and the associated qualifiers for AQL, the Aquarelle Query Language [15]. Appendix B gives some guidance in mapping databases onto this attribute set and the available tags. Aquarelle cultural partners may wish to restrict their attention to these appendices and those sections of the main document referred to by the notes.

Numbers appearing in square brackets [like this] indicate references to the source documents listed in Appendix D.

As well as the normative material, this document contains annotations, interpretations, and indications of what may change in future versions. This material is set in small type on the right-hand side of the page,

like this.

These annotations are provided in the hope that they will be useful to implementors: they do not constitute a part of the Aquarelle Profile itself.

2. Aquarelle Versions

This document describes the profile for version 0.2 of the Aquarelle system. The profile will be revised when the requirements for version 0.3 are agreed. Anticipated changes are flagged in the commentary to the text.

Version 0.2 is due for completion by the end of July 1997. Version 0.3 will incorporate some changes to this profile, and will include folder publishing and unpublishing. It is due for completion on 20th

September 1997. Subject to testing and minor enhancements, version 0.3 should become version 1, and is due for delivery to the Quality Manager on 13th October 1997.

3. Interoperability

3.1 Aquarelle Interoperability

The primary interoperability goal of Aquarelle clients and servers is to interoperate with one another. This profile primarily addresses this central requirement.

3.2 CIMI Interoperability

A subsidiary goal is for Aquarelle clients to interoperate with CIMI servers and vice versa.

The Aquarelle project's use of Z39.50 is very similar to what is described in the CIMI Profile [7] and the subsequent Implementors Agreement [8]. 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 (especially in the area of the attribute set).

Most of the extensions fall into two categories: firstly, support for the Aquarelle architecture (with the Access Server and emphasis on SGML documents); and secondly, 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.

So implementors should feel free to suggest changes and extensions to this profile (including the attribute set defined in Appendix A) which are not compatible with CIMI.

3.3 Other Z39.50 Interoperability

The third and least important interoperability goal is for Aquarelle servers to be searchable by other Z39.50 clients, and vice versa. Among non-Aquarelle, non-CIMI Z39.50 clients and servers, those which have the best chance of interoperating usefully with Aquarelle software are those which implement some or all of the Collections Profile [6], in particular the GRS-1 record syntax.

4. Z39.50 Specification for the Aquarelle Profile

4.1 Protocol Version

The Aquarelle Profile requires that clients and servers support Z39.50 Version 2 as specified in the Z39.50 standard [1]. Version 3 of the protocol may be negotiated during Init when both client and server can support it.

It may become desirable to use features of Z39.50 which are only available in version 3 of the protocol: for example, the ability to formulate a query using multiple attribute sets; the facility to provide a "complex element specification" instead of an element-set name; use of the searchResponse APDU's AdditionalSearchInformation parameter. In this case, clients and servers may be mandated to support Z39.50 Version 3.

4.2 Z39.50 Objects

The Aquarelle Profile requires that clients and servers support the following Z39.50 objects:

Object	OID	Client	Server
Aquarelle attribute set	{ANSI-standard-Z39.50 3 8}	X	X
Bib-1 diagnostic set	{ANSI-standard-Z39.50 4 1}	X	X
GRS-1 record syntax	{ANSI-standard-Z39.50 5 105}	X	X
tagSet-M	{ANSI-standard-Z39.50 14 1}	X	
tagSet-G	{ANSI-standard-Z39.50 14 2}	X	X
Collections tag set	{ANSI-standard-Z39.50 14 5}	X	X
CIMI tag set	{ANSI-standard-Z39.50 14 *Note 1*}	X	X

Notes:

1. The CIMI tag set has not yet been assigned an OID.

Clients and servers wishing to achieve maximum interoperability with non-Aquarelle software may elect also to support the BIB-1 attribute set and the USMARC record syntax.

4.3 Z39.50 Services

4.3.1 Minimum Support

Both clients and servers are required to support three Z39.50 Services: Init, Search and Present. Standard Z39.50 Init Service negotiation procedures control the use of all services.

Client and servers may, of course, elect also to implement other Z39.50 services so as to improve interoperability with non-Aquarelle Z39.50 software.

Z39.50 segmentation of records is not required. Clients and servers should negotiate reasonable values when negotiating the minimum value of the maximum record size parameter.

It might be worth actually specifying a value here. How big will SGML documents get? 10k? 100k? A megabyte? (Images and other media objects are transferred by URL, so they don't contribute significantly to the record size.)

Or perhaps segmentation should be required?

4.3.2 Init

Servers may choose between three auth types: They may elect to make no requirements of clients, or to require a specific single string to be sent, or to support a set of groupId/userId/password triples.

Aquarelle clients may use the idAuthentication parameter of the InitializeRequest APDU to pass authentication information to the server. They can satisfy servers' three authentication types using the anonymous, open and idPass choices of IdAuthentication respectively.

The Aquarelle Access Server will make metadata available to clients, so that they know which of these options is required by each server.

Servers should accept the absence of any authentication information as being equivalent to `anonymous` authentication.

In the interests of short-term interoperability during the version 0.2 phase, servers should either make no authentication demands or require the `open` authentication string "aquarelle".

Servers may return a diagnostic if authentication fails. The return of the diagnostic in the userInformationField parameter is governed by a Z39.50 Implementors Agreement [2].

4.3.3 Search

4.3.3.1 Search Overview

Clients must be able to generate Z39.50 Type-1 and/or Type-101 (RPN) queries.

When Z39.50 Version 2 is in force, Type-101 queries are necessary to express proximity conditions. Proximity support is not required for version 0.2, but may be required in future versions.

Servers must understand both Type-1 and Type-101 queries.

Servers must support named result sets.

Servers must support a minimum of two concurrent result sets.

Servers must support the inclusion of references to existing result-sets in new queries: for example, a search which ANDs an existing result-set with an additional search term.

The last two of these requirements enable iterative search refinement. This feature is not used in Aquarelle version 0.2, so the requirements for two concurrent result-sets and result-set combination with new terms may be relaxed for the moment.

4.3.3.2 Database Names

Each server must make available for searching a single database called "aquarelle". Data providers who wish to publish multiple databases must run multiple servers on different ports and/or hosts, each providing access to a single database.

This restriction is imposed by the current Aquarelle Access Server's metadata architecture, which makes the assumption that each server publishes a single database. This will change after version 1.0.

4.3.3.3 Attribute Sets

Clients and servers must support the Aquarelle attribute set, which is defined in Appendix A. Servers need not, however, support all attribute types and values in the attribute set, provided they return a suitable diagnostic. See Appendix A for details of which attributes are mandatory and which are optional.

4.3.3.4 Proximity Operation

Clients may use a proximity search with the unit ELEMENT to express a search for terms within the same semantic unit. For example, if an object was sculpted by John and modeled by Jane, then a search for

```
 (author="john" AND role="sculptor")
PROX with unit ELEMENT, distance 0
 (author="jane" AND role="model")
```

would find that object, but not objects sculpted by Jane and modeled by John, whereas using an AND operator in place of the PROX would find both records.

Servers must support this meaning for proximity in searches.

Although it won't be used in version 0.2.

4.3.3.5 Date Searches

Dates occurring as search terms should, when possible, be transmitted in normalised form with the attribute 4=5 (structure=normalised date) attached to the search-term. The normalised form is as defined for GeneralizedTime in Section three, paragraph 30 of the ASN.1 standard [5], except that the only mandatory portion of the string is the four-digit representation of the year.

For example:

1968	The year 1968
196803	March 1968
19680312	12th March 1968
196803120434	34 minutes past 4 in the morning, 12th March 1968
19680312043452.2	34 minutes and 52.2 seconds past 4am, 12th March 1968
19680312043452.2Z	34 minutes and 52.2 seconds past 4am, 12th March 1968, UTC (Universal Co-ordinated Time) – previous examples have been local time.

196803120434+0200	34 minutes past 4am, 12th March 1968, in the timezone two hours ahead of UTC.

If the client is not able to normalise the date, then it may transmit it in whatever form it has, without the structure attribute; and the server must interpret it as best it can.

Dates should generally be translated to the canonical form at as high a level as possible, since only at high levels can the software know configuration information such as the natural language of the human using the system, local date-formatting conventions, etc.

4.3.3.6 Range Searches

Range searches are specified by a single search-term, carrying the attribute 3=3 (relation=equal), and with the range end-points separated by a colon (:).

This notation is unambiguous when searching against an access-point for which values can not include colons – for example, numeric co-ordinates in a spatial referencing system, or GeneralizedTime-encoded dates.

For example, a search for dates in the range 1966-1984 would be encoded as a search term with attribute 3=3 and value “1966:1984”

The means of expressing a range-search at the Z39.50 level may change soon dependent on the outcome of range-search discussions at forthcoming ZIG meetings. Candidate representations include overloading of the proximity operator with distance zero, and a new type of sequence-of-terms concept, which can contain both end-points of a range, and can be annotated with a new relation=between attribute.

4.3.4 Retrieval

4.3.4.1 Tag Types

The following tag types are available for use in GRS-1 records:

Type	Usage
1	Elements from tagSet-M.
2	Elements from tagSet-G.
3	Reserved for tags locally defined by a server.
4	Elements from the Collections tag set.
5	Elements from the CIMI tag set.

Tag-type 3 is not used in version 0.2

tagSet-M and tagSet-G are defined in Appendix TAG of the Z39.50 standard [1], in sub-sections 2.1 and 2.2 respectively. The Collections tag set is defined in Collections Profile

[6], section 4.5. As yet, there is no formal definition of the CIMI tag set.

It may become apparent that servers have data which can not be described by any of the tags in tagSet-M, tagSet-G or the Collections or CIMI tag set. In this case, additional tags may be proposed for addition to the CIMI tag set, or a new Aquarelle tag set created.

4.3.4.2 Record Structure

In order to provide more structure to the data, and to maintain interoperability with the CIMI and Collections profiles, Aquarelle will adopt the hierarchical record structure described in section 7.4.3.2 of the CIMI Profile [7]. The current flat record is a compromise to facilitate rapid development of Aquarelle version 0.2.

The record structure for Aquarelle is defined as follows:

Element	Occurrence Repeatable?	Tag type Tag value	Datatype Semantics
typeOfDescriptiveRecord	mandatory no	4 1	INTEGER Defined in Collections Profile
typeOfObject	mandatory no	4 12	INTEGER Defined in Collections Profile
categoryOfObject	mandatory no	4 12	InternationalString Defined in Collections Profile
server	*Note 1* no	4 24	InternationalString Defined in Collections Profile
db	*Note 1* no	4 25	InternationalString Defined in Collections Profile
recordID	*Note 1* no	4 37	OCTET STRING Defined in Collections Profile
alternativeIdentifier	*Note 1* no	4 38	OCTET STRING Defined in Collections Profile
title	*Note 2* no	2 1	OCTET STRING Defined in tagSet-G
date	optional	2 8	InternationalString Defined in tagSet-G
description	optional	2 17	InternationalString Defined in tagSet-G
creator	mandatory	5 1	InternationalString Defined in CIMI Profile
objectTitle	*Note 2*	5 2	InternationalString Defined in CIMI Profile

currentLocation	optional	5 3	InternationalString Defined in CIMI Profile
subjectDescription	mandatory	5 4	InternationalString Defined in CIMI Profile
museumObjectId	optional	5 5	InternationalString Defined in CIMI Profile
nationalityCultureRace	optional	5 6	InternationalString Defined in CIMI Profile
materialMedium	optional	5 7	InternationalString Defined in CIMI Profile
typeClassification	optional	5 8	InternationalString Defined in CIMI Profile
creditLine	optional	5 9	InternationalString Defined in CIMI Profile
creatorDateOfBirth	optional	5 10	InternationalString Defined in CIMI Profile
creatorDateOfDeath	optional	5 11	InternationalString Defined in CIMI Profile
creatorRole	optional	5 12	InternationalString Defined in CIMI Profile
actionAssociatedWithDate	optional	5 13	InternationalString Defined in CIMI Profile
displayObject	optional yes	2 9	Image Images of object

Notes:

1. Either server, db and recordID all occur, or alternativeIdentifier occurs, but not both.
2. Either title or objectTitle occurs, but not both.

In addition to the elements mentioned in this record structure, records may contain any other tagSet-M and/or tagSet-G elements: the particular, the use of Dublin Core Elements [12], expressed using the proposed extensions to tagSet-G [13], is encouraged.

The use of tagSet-G's element 9 for images is based upon the assumption that the current proposal to replace the current meaning (bodyOfDisplay) with displayObject is accepted.

The structure for the Image sub-record is defined as follows:

Element	Occurrence Repeatable?	Tag type Tag value	Datatype Semantics
rendition	mandatory yes	5 14	Rendition Renditions of image

Dublin Core (*Note 1*)	optional *As appropriate*	*As appropriate* *As appropriate*	*As appropriate* Standard semantics

Notes:

1. Any of the Dublin Core elements, with the exception of Identifier, may be included in Image sub-records.

The structure for the Rendition sub-record is defined as follows:

Element	Occurrence Repeatable?	Tag type Tag value	Datatype Semantics
identifier	mandatory no	2 28	InternationalString URL of actual image
Dublin Core (*Note 1*)	optional *As appropriate*	*As appropriate* *As appropriate*	*As appropriate* Standard semantics

Notes:

1. Any of the Dublin Core elements, including Identifier, may be included in Rendition sub-records. Only Identifier is mandatory.

The use of tag 28 to represent the element in which URLs are specified is taken from the proposed extensions to tagSet-G [13], which have not yet been formally accepted by the Z39.50 Implementors Group.

A record may contain zero or more Images, each of which must contain one or more Renditions and which may also have associated with it any relevant metadata (eg. copyright restrictions). Each rendition may contain additional Dublin Core elements, containing more specific information than that in the containing Image record, which it overrides when the element is specified in both places. Each rendition must include the Identifier element, containing the location of the actual image file as a URL.

For encoding the location of image files, http URLs are preferred to other forms such as ftp and z3950r .

The preferred image-file formats are currently JPEG and GIF, as they are most widely recognised by web browsers. However, the PNG format is increasingly also recognised: as this format is technically superior to GIF in all respects, as well as being free from the LZW patent, this will perhaps become preferable after version 0.2.

Servers must arrange that the renditions of each image are provided in ascending order in the Image record. They may in addition attach appliedVariant information specifying the size of the image-files, in bytes, pixels or both.

The intent of this structure for Rendition sub-records is to allow each image to be made available in a number of renditions, which will typically vary in size: this allows client software to choose which to fetch – thumbnails, screen resolution, etc.

The structure of Image and Rendition sub-records may also be used for shipping multiple renditions of media objects other than images – for example, audio clips, video clips and VRML constructs. In this case, appliedVariants should be used to specify the types of the media objects.

4.3.4.3 Element Set Names

4.3.4.3.1 Supported Element Set Names

Clients and servers must support three element set names: “f” (full), “b” (brief), and “i” (unique identifier). They are described in the following sections.

4.3.4.3.2 “f” (Full)

When a client requests a full record, the record returned by the server should include all data that is to be published via the Aquarelle project.

Databases may contain additional information which is available only through bespoke interfaces, perhaps for reasons pertaining to security or IPR: this profile does not require such information to be included in full records.

4.3.4.3.3 “b” (Brief)

Aquarelle servers are recommended to include the following information when asked for a brief record:

Element	Occurrence
typeOfDescriptiveRecord	Always included
typeOfObject	Always included
categoryOfObject	Always included
server	Included if applicable
db	Included if applicable
recordID	Included if applicable
alternativeIdentifier	Included if and only if server, db and recordID are absent
creator	Always included
title	Included if and only if objectTitle is absent
objectTitle	Included if and only if Title is absent
date	optional
currentLocation	optional
description	optional
displayObject	optional

Servers may exercise some judgement in the choice of elements to be returned, omitting even elements described here as mandatory, and including tagSet-M and tagSet-G elements not explicitly mentioned in the record structure.

Clients may ignore such additional elements, or display them as best they can.

4.3.4.3.4 "i" (Unique Identifier)

This element-set name may be used by a client to request an opaque unique identifier for a record. The server must return a GRS-1 record with a single element containing the unique identifier with tag type 2 (tagSet-G), tag value 5 (documentId). The identifier must be a string, not intended for display, which contains all the information necessary to locate the record in the database on a subsequent occasion.

Unlike the "f" and "b", which are required by the Z39.50 standard [1] to be supported by all servers, the "i" element set name is unique to Aquarelle.

Subsequent location of the record is performed by a search consisting of a single search-term, namely the previously-returned unique identifier, to which the attribute use=local number (1=12) is attached. Servers must support this search.

It may be that the identifier is intended to include a specification of the server address and database-name (ie. it identifies the document from anywhere in the world). In this case, retrieval lies outside the domain of Z39.50 and is presumably an issue for Folder Servers.

4.3.4.4 Use of SGML

Aquarelle servers are divided into two categories: Archive Server and Folder Server (see the Aquarelle Architecture document [10] for details.)

Folder servers will generally return all their information in a single SGML Document. In this case, the document must be packaged in a GRS-1 record as described below.

Archive servers may also elect to return a single SGML document, in which case they must use the same GRS-1 wrapping as Folder Servers. However, in the interests of interoperability with other Z39.50 clients, particularly those influenced by CIMI, Archive Servers are encouraged to return fielded data.

The Aquarelle Access Server includes a portion called the Z39.50 Client, which has among its functions the transformation of fielded GRS-1 records into SGML documents encoded with the "simple" DTD. So there is no need for Data Servers to compromise their interoperability with other software by duplicating this process.

Servers which return full records as SGML documents may use any DTD; it is anticipated that the CI and CIMI DTDs will be used most often. When brief records are returned as SGML documents, however, the "simple" DTD *must* be used.

4.3.4.5 GRS-1

4.3.4.5.1 GRS-1 Overview

Servers must support the record syntax GRS-1, described in Appendices REC.5 (Generic Record Syntax 1), and RET (Z39.50 Retrieval) of the Z39.50 standard [1].

In the interests of interoperability with non-Aquarelle Z39.50 clients, servers may elect also to support other record syntaxes such as SUTRS and USMARC.

4.3.4.5.2 Transmitting SGML Data

Aquarelle Data Servers should send a SGML documents as GRS-1 records with a single element having tag type 2 (tagSet-G) and tag value 19 (DocumentContent).

The element's content should contain the text of the document in the `octets` or `string` branch of the ElementData CHOICE.

The type of the data must be specified by an appliedVariant using the standard variant-set variant-1, having class 2 (BodyPartType), type 2 (Z39.50Type), and value "sgml/*xxx*", where *xxx* is the DTD name.

The element's tagOccurrence and metaData, if supplied, are currently ignored.

This may change in the future: in particular, GRS-1 metaData may be used to support features such as hits information.

The complete GRS-1 structure looks like this in ASN.1:

```
GenericRecord ::= SEQUENCE OF -- just one SEQUENCE
    SEQUENCE {
    tagType          [1] IMPLICIT INTEGER {2} -- tagSet-G
    tagValue         [2] INTEGER {19} -- Document content
    content          [4] OCTET STRING, -- SGML file goes here
    appliedVariant   [6] IMPLICIT SEQUENCE { -- just one in the SEQUENCE
                triples    [2] IMPLICIT SEQUENCE OF -- just one SEQUENCE
                    SEQUENCE{
                        variantSetId  [0] IMPLICIT OBJECT IDENTIFIER,
                                                -- 1.2.840.10003.12.1
                        class         [1] IMPLICIT INTEGER {2},
                        type          [2] IMPLICIT INTEGER {2},
                        value         [3] InternationalString,
        -- value is the string "sgml/<xxx>" where <xxx> is the DTD name.
```

For more details, see Appendices REC.5 (Generic Record Syntax 1), and VAR (Variant Sets) of the Z39.50 standard [1].

4.3.4.5.3 Transmitting Images

Aquarelle servers should not encode images directly in GRS-1 records and send them via Z39.50; instead, URLs should be used, wrapped in the structure described in section 4.3.4.2, and an external protocol such as HTTP or FTP used to fetch the image files themselves.

Later versions of this profile may allow images to be transmitted in line as part of GRS-1 records. In this case, servers will be required also to transmit the image-file format by attaching an appliedVariant using the standard variant-set variant-1, having class 2 (BodyPartType), type 1 (ianaType/subType), and the appropriate MIME type as its value.

4.3.4.5.4 Search-Term Highlighting

Servers may provide search-term highlighting (hits) information, but are not obliged to do so. Highlighting information for a given element is specified by filling in the hits member of the TaggedElement's ElementMetaData structure. This member is described in Appendix RET.3.2.3.1 of the Z39.50 standard [1].

Clients may use the hits information, if provided, to highlight the specified portions of the element (eg. by using a different colour, or underlining); or they may ignore this information.

4.3.4.5.5 Relevance Feedback

Servers may provide relevance feedback by including the tagSet-M elements rank (10) and/or score (18) in the retrieval record, but are not obliged to do so. These tags are described in Appendix TAG.2.1 of the Z39.50 standard [1].

Clients may use these elements, if provided, to determine the order in which they display records; or they may display them to the user; or they may ignore them completely.

4.4 Diagnostic Messages

Clients and servers must support the BIB-1 diagnostic set.

If servers generate errors which are not currently represented in the BIB-1 diagnostic set, new diagnostics may be proposed. These may be absorbed into BIB-1, or placed in a new Aquarelle-specific diagnostic set.

Appendix A. The Aquarelle Attribute Set

A.1 Attribute Set Overview

This appendix specifies a Z39.50 attribute set for use in searching databases provided by Aquarelle Data Servers. Since the attributes in this set are typically generated by qualifiers in AQL (the Aquarelle Query Language), the AQL qualifiers are listed alongside the attributes which they generate.

In order to facilitate interoperability, this attribute set shares the object identifier used by the CIMI-1 attribute-set. The Aquarelle attribute-set provides extensions to CIMI-1 and provides guidelines in its use. The OID is {Z39-50 attributeSet 8}, that is, 1.2.840.10003.3.8

This attribute set imports the following types from the BIB-1 attribute set defined in the Z39.50 standard:

Type	Name	Restrictions, Constraints	Notes
1	Use	All values defined in Z39.50 BIB-1	see below
2	Relation	All values defined in Z39.50 BIB-1	
3	Position	All values defined in Z39.50 BIB-1	may be ignored
4	Structure	All values defined in Z39.50 BIB-1	
5	Truncation	All values defined in Z39.50 BIB-1	
6	Completeness	All values defined in Z39.50 BIB-1	may be ignored

For more details of the semantics of attributes inherited from BIB-1, including the meanings of inherited access points, see the BIB-1 semantics document [4].

One additional type is defined by this attribute set: 101, Authority: a value identifying the authoritative source from which a term is taken.

A.2 USE Attributes (Access Points)

It is recommended that queries restrict their use of imported BIB-1 USE attributes to the following:

Value	Name	AQL Qualifier
1	personal name	access$personal_name
2	corporate name	access$corporate_name
4	title	access$title
12	local number	access$local_number
20	local classification	access$local_classification
21	subject	access$subject
30	date	access$date
31	date of publication	access$date_of_publication
32	date of acquisition	access$date_of_acquisition

54	Code--language	access$language
58	name geographic	access$name_geographic
62	abstract	access$description
1003	author	access$author
1016	any	access$any
1018	publisher	access$publisher
1031	material type	access$resource_type

Local number (12) may be used to search for a record by the unique identifier earlier retrieved by a fetch with element set name "i". See the note in section 4.3.4.3.4 for details.

The Aquarelle attribute-set may extend the subset of supported BIB-1 access-points in the future: for example, Date/time added to database (1011) may be useful.

The following additional access points are from the CIMI-1 attribute set:

Value	Name	AQL Qualifier
2000	award	access$award
2001	bibliography	access$bibliography
2002	collection	access$collection
2003	concept	access$concept
2004	copyright restrictions	access$copyright_restrictions
2005	credit line	access$credit_line
2006	event	access$event
2007	inscription/mark	access$inscription_or_mark
2008	material	access$material
2009	nationality/culture/race	access$nationality_or_culture_or_race
2010	object	access$object
2011	occupation	access$occupation
2012	process/technique	access$process_or_technique
2013	quote	access$quote
2014	role	access$role
2015	subject description	access$subject description
2016	subject identification	access$subject identification
2017	styles/movements	access$styles_or_movements
2018	technique	access$technique
2019	type/classification	access$type_or_classification

Role (2014) may be used in conjunction with author (1003) to find records describing objects in which a particular person had a particular creative role. See section 4.3.3.4 for details.

For descriptions of the semantics of access points from CIMI-1, see Appendix A of The CIMI Profile [7]. For a more detailed description of the CIMI access points, see the CIMI DTD Tagging Guide [9].

As well as inherited BIB-1 and CIMI-1 attributes, the Aquarelle attribute set defines seven additional values for USE attributes:

Value	Name	AQL Qualifier
3000	protection status	access$protection_status
3001	protection date	access$protection_date
3002	physical condition	access$physical_condition
3003	spatial referencing system	access$ref_system
3004	X-coordinate in referencing system	access$x_coordinate
3005	Y-coordinate in referencing system	access$y_coordinate
3006	historical context	access$historical_context

The Aquarelle-specific access points arise from the need to search for sites and buildings in architectural heritage databases. Semantics follow: they are derived from Council of Europe Recommendation No. R (95) 3 [11].

Protection status

Indicates whether a building is protected, and if so the type of protection.

NOTE *Until "type of protection" has been codified across different national practices, this can only be treated as a text field.*

Protection date

The date at which the protection status was granted.

Physical condition

Records the integrity of the building (demolished, ruined, remodelled, restored, etc.) and its state (good, fair, poor, bad, etc.). A controlled vocabulary may be proposed at some stage.

Spatial referencing system

A string indicating the spatial referencing system in which search terms for X-coordinate and Y-coordinate are expressed: for example, "UTM", "Lambert", "Gauss-Boaga", etc. A strict controlled vocabulary may be proposed at some stage.

X-coordinate in referencing system

Y-coordinate in referencing system

A pair of numbers indicating a point in the nominated spatial referencing system; or a pair of ranges indicating an area.

Historical context

Summary of the historical development of a building or other object (construction phases, etc.) Political, social, economic or religious events or circumstances associated with the object.

There may be further Aquarelle-specific access points in the future. Any or all of them may in the future be adopted by CIMI.

If no USE attribute is specified in a query, the default behaviour is to search across all access-points; this is equivalent to explicitly specifying "any" (1016).

A.3 RELATION Attributes

All of the RELATION attributes defined in the BIB-1 attribute set may be used in queries; servers are only required to implement the equality attribute (3). Less-than (1), less-than-or-equal (2), greater-than-or-equal (4), greater-than (5) and not-equal (6) are all considered desirable.

Value	Name	AQL Qualifier
1	less than	rel$lt
2	less than or equal	rel$le
3	equal	rel$eq
4	greater than or equal	rel$ge
5	greater than	rel$gt
6	not equal	rel$ne
100	phonetic	rel$phonetic
101	stem	rel$stem
102	relevance	rel$relevance
103	always matches	rel$always

Note that equal (3) is currently also used for range searching. See section 4.3.3.6 for details.

If no RELATION attribute is specified in a query, the default behaviour is to search for terms equal to the specified data; this is equivalent to explicitly specifying equality (3).

A.4 POSITION Attributes

All of the POSITION attributes defined by BIB-1 may be used in queries, and must be recognised by servers, but may be ignored.

Value	Name	AQL Qualifier
1	first in field	pos$first_in_field
2	first in subfield	pos$first_in_subfield
3	any position in field	pos$any

A.5 STRUCTURE Attributes

All of the STRUCTURE attributes defined by BIB-1 may be used in queries, and must be recognised by servers, but all except data (normalised) (5) may be ignored.

Value	Name	AQL Qualifier
1	phrase	struct$phrase
2	word	struct$word
3	key	struct$key
4	year	struct$year
5	date (normalised)	struct$date
6	word list	struct$word_list
100	date (un-normalised)	struct$raw_date
101	name (normalised)	struct$name
102	name (un-normalised)	struct$raw_name
103	structure	struct$structure
104	urx	struct$urx
105	free-form-text	struct$free_form_text
106	document-text	struct$document_text
107	local number	struct$local_number
108	string	struct$string
109	numeric string	struct$numeric_string

Date (normalised) (5) should be attached to all search-terms which are dates encoded as ASN.1 GeneralizedTime. See section 4.3.3.5 for details.

A.6 TRUNCATION Attributes

All of the RELATION attributes defined in the BIB-1 attribute set may be used in queries; servers are only required to implement the no-truncation attribute (100). Right-truncation (1) left-truncation (2), left-and-right-truncation (3) and process-#-in-search-term (101) are all considered desirable.

Value	Name	AQL Qualifier
1	right truncation	trunc$right
2	left truncation	trunc$left
3	left and right	trunc$both
100	do not truncate	trunc$none
101	process # in search term	trunc$hash
102	regExpr-1	trunc$regexp
103	regExpr-2	trunc$regexp2

If there is sufficient demand, servers may be required also to implement POSIX-like regular expressions (102), or ISO 8777-like masking patterns in which # stands for any single character and ? stands for any number of characters. (No attribute value is assigned for this yet.)

If no TRUNCATION attribute is specified in a query, the default behaviour is to search with no truncation; this is equivalent to explicitly specifying "none" (100).

Servers may right-truncate even when explicitly requested to perform no truncation. See section 4.3.3.7 for details.

A.7 COMPLETION Attributes

All of the COMPLETION attributes defined by BIB-1 may be used in queries, and must be recognised by servers, but may be ignored.

Value	Name	AQL Qualifier
1	incomplete subfield	comp$incomplete
2	complete subfield	comp$subfield
3	complete field	comp$field

A.8 AUTHORITY Attributes

These attributes identify the authoritative source from which a search-term is taken. Servers may modify their interpretation of search-terms with these attributes attached in any way they please: If no AUTHORITY attribute is specified in a query, the server is to interpret this as the client not saying anything about the term's source.

The authoritative sources listed here are from the CIMI-1 attribute set, and are described in ### [14].

Value	Name	AQL Qualifier
1	Non-Authoritative	auth$Non_Authoritative
2	Local-to-server	auth$Local_to_server
3	USMARC	auth$USMARC
4	LCSH	auth$LCSH
5	AAT	auth$AAT
6	AAT_Date	auth$AAT_Date
7	ACRL/RBMS_Binding	auth$ACRL_or_RBMS_Binding
8	ACRL/RBMS_Genre	auth$ACRL_or_RBMS_Genre
9	ACRL/RBMS_Paper	auth$ACRL_or_RBMS_Paper
10	ACRL/RBMS_Printing	auth$ACRL_or_RBMS_Printing
11	ACRL/RBMS_Type	auth$ACRL_or_RBMS_Type
12	Base_Merimee	auth$Base_Merimee
13	BGN	auth$BGN
14	British_Archaeological	auth$British_Archaeological
15	Canadiana	auth$Canadiana
16	Dictionarium_Museologicum	auth$Dictionarium_Museologicum
17	Garnier	auth$Garnier

18	Geosaurus	auth$Geosaurus
19	Glass	auth$Glass
20	ICOM_Costume	auth$ICOM_Costume
21	ICONCLASS	auth$ICONCLASS
22	Jewish_Art	auth$Jewish_Art
23	ISO_Language	auth$ISO_Language
24	ISO_Documentation	auth$ISO_Documentation
25	ISO_Iconic	auth$ISO_Iconic
26	ISO_AV	auth$ISO_AV
27	ISO_Date/Time	auth$ISO_Date_or_Time
28	LC_Descriptive_Graphic	auth$LC_Descriptive_Graphic
29	LC_Name	auth$LC_Name
30	LC_Thesaurus_Graphic	auth$LC_Thesaurus_Graphic
31	Moving_Image_Materials	auth$Moving_Image_Materials
32	Nomenclature	auth$Nomenclature
33	Reynies	auth$Reynies
34	TGN	auth$TGN
35	Tozzer	auth$Tozzer
36	ULAN	auth$ULAN
37	Villard	auth$Villard
38	Yale_British_Artists	auth$Yale_British_Artists

The Aquarelle attribute set defines one additional value of the AUTHORITY attribute:

Value	Name	AQL Qualifier
1000	RCHME	auth$RCHME

There may be further Aquarelle-specific authoritative sources in the future. Any or all of them may in the future be adopted by CIMI.

For details of the semantics of AUTHORITY attributes from CIMI-1, see Appendix A of the CIMI Profile [7].

Appendix B. Dublin Core Mapping

The following table is provided for the use of server providers attempting to map their databases onto the CIMI attribute set and GRS-1 tags. The USE Attribute suggests a way of specifying how to search for each DC element, and the Tag set and Tag value suggest how each DC element can be tagged in a GRS-1 record.

CAVEAT The suggested mapping is approximate, and is supplied for guidance only. Individual databases may map more accurately to other attributes and tags than the suggested ones.

Dublin Core	USE Attribute	Tag set	Tag value
Title	title (4)	2 (tagSet-G)	1 (title)
Creator	author (1003)	2 (tagSet-G)	2 (author)
Subject	subject (21)	2 (*Note 1*)	21 (subject)
		5 (CIMI)	5 (subjectDescription)
Description	abstract (62)	2 (tagSet-G)	17 (description)
Publisher	publisher (1018)	2 (*Note 1*)	31 (publisher)
		2 (tagSet-G)	10 (organisation)
Contributors	*Note 2*	2 (*Note 1*)	32 (contributors)
		5 (CIMI)	1 (creator)
Date	date of publication (31)	2 (tagSet-G)	8 (date)
Type	material-type (1031)	2 (*Note 1*)	22 (resourceType)
Format	–	2 (*Note 1*)	27 (format)
		5 (CIMI)	7 (materialMedium)
Identifier	Identifier-document (1032)	2 (*Note 1*)	28 (Identifier)
		2 (tagSet-G)	5 (documentId)
Source	–	2 (*Note 1*)	33 (source)
Language	Code--language (54)	2 (*Note 1*)	20 (language)
Relation	–	2 (*Note 1*)	30 (relation)
Coverage	–	2 (*Note 1*)	34 (coverage)
Rights	copyright restrictions (2004)	2 (*Note 1*)	29 (rights)
		5 (CIMI)	9 (creditLine)

Notes:

1. These elements are from the proposed extensions to tagSet-G [13], which have not yet been formally accepted by the Z39.50 Implementors Group. Where equivalents exist in the CIMI tag set, or less exact matches exist in the standardised part of tagSet-G, they may be preferred for use in the Aquarelle project.

2. It may be acceptable to treat OtherAgent as another instance of Creator for searching purposes, using the BIB-1 access point Personal name (1) or Author (1003).

Appendix C. Server-Specific Interpretations

Servers are allowed some leeway in how they interpret searches.

In particular:

- Servers may default to performing single-level monolingual thesaurus expansion, so that, for example, a search for `church` actually searches for `church or chapel or cathedral`.
- Even if the attribute truncation=none (5=100) is explicitly specified, servers may perform right-truncation, so that, for example, a search for `bus` will also find `bush`, `busy` and `business`.

  Because of its potentially surprising consequences, this dispensation may be withdrawn in the future.

- If servers are not able to perform range searches to the precision specified in a query, they may widen or narrow the range search to one which can be supported: for example, a server asked to find records with dates between 1730 and 1780 might widen this to the range 1700 to 1800 if century date-ranges are all that is supported.

  Widening ranges is probably to be preferred to narrowing, since it is better to find all the records that should be found plus some "false hits" than to find only some of the desired records.

  This kind of adjustment is frowned upon in the Z39.50 community. Server implementors should avoid it if at all possible.

  Later versions of this profile may require that servers which alter range searches in this way comment on the fact using the subqueryRecommendation element of a SearchInfoReport returned in the Additional-search-information parameter of the searchResponse APDU. (See section 3.2.2.1.12 and Appendix USR.1 of the Z39.50 standard [1] for details.)

Unusual interpretations such as these must be documented as part of the archive-specific documentation. This information may be codified and held as meta-data in the Access Server in a future release.

Appendix D. References

The following reference documents provide useful background material to this profile:

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

[2] Z39.50 Implementors Agreement. (1996). *Returning Diagnostics in an Init Response*. Z39.50 Implementors Agreements are available from the Z39.50 Maintenance Agency at
`http://lcweb.loc.gov/z3950/agency/agree.html`

[3] Lynch, Clifford A. (1994). RFC 1729, *Using the Z39.50 Information Retrieval Protocol in the Internet Environment*. Available at
`http://ds.internic.net/rfc/rfc1729.txt`

[4] *Attribute Set BIB-1 (Z39.50-1995): Semantics*. Available at
`ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt`

[5] International Standards Organization. *ISO 8824. Specification of Abstract Syntax Notation 1 (ASN.1)* Available from ANSI, 1430 Broadway, New York, NY 10018, Tel. 212 642-4932

[6] Library of Congress. (1996). *Z39.50 Profile for Access to Digital Collections*. Available as HTML or Postscript files at
`http://lcweb.loc.gov/z3950/agency/profiles/collections.html`
`ftp://ftp.loc.gov/pub/z3950/profiles/collections.ps`

[7] CIMI Profile Development Working Group. (1996). *The CIMI Profile: Z39.50 Application Profile Specifications for Use in Project CHIO*. Available at
`http://lcweb.loc.gov/z3950/agency/profiles/cimi2.html`
`ftp://ftp.cimi.org/pub/cimi/CIMI_Profile/`

[8] CIMI Profile Development Working Group. (1997). *CIMI Z39.50 Interoperability Testbed Implementors Agreement* (preliminary draft of 28th May 1997) Available at
`http://people.unt.edu/~wem0002/CIMI/cimiz3950.html`

[9] *The CIMI DTD Tagging Guide*. Available at
`ftp://ftp.cimi.org/pub/cimi/CIMI_SGML/TG-HTM.ZIP`

[10] Duce, Sutcliffe, Watson and Mac Randal. (1996). *Aquarelle Technical Specifications – Version 1*, Deliverable D3.1. Available to Aquarelle partners from the Aquarelle web site document repository,
`http://aquarelle.inria.fr/Aquarelle/Member/technical-doc.html`

A good overview of the Aquarelle system architecture can be obtained from reading chapters 2 and 3, which are only 8 pages long in total.

[11] Council of Europe, Committee of Ministers. (1995). *On Co-ordinating Documentation Methods and Systems Related to Historic Buildings and Monuments of the Architectural Heritage*, Recommendation No. R (95) 3

[12] *Dublin Core Metadata Element Set: Reference Description*, revision of January 15, 1997. Available at
`http://purl.org/metadata/dublin_core_elements`

[13] Library of Congress. (1997). *TagSet Proposal*, revision of March 27, 1997. Available at
`http://lcweb.loc.gov/z3950/agency/aprilzig/tags.html`

[14] *### the authority sources document*.

[15] Dale Sutcliffe. (1997). *Aquarelle Query Language (AQL)*. Available to Aquarelle partners from the Aquarelle web site document repository,
`http://aquarelle.inria.fr/Aquarelle/Member/technical-doc.html`
This document may not be available until July 1997.

Title: **Aquarelle Z39.50 Profile**

Author: System Simulation Ltd.

Date: June 1997

Revision: 1.14

Location: /tmp_mnt/nfs/appl/aquarelle/doc/SCCS/s.profile.m

Status: THIS DOCUMENT IS A WORKING DRAFT. COMMENTS AND REQUESTS FOR EXTENSIONS ARE WELCOME: THEY SHOULD BE SUBMITTED BY EMAIL TO `aqwp@ssl.co.uk`
