Interoperability between Multimedia Collections for Content and Metadata-Based Searching

P. Allen[1], M. Boniface[1], P. Lewis[2], K. Martinez[2]

[1] [pja,mjb]@it-innovation.soton.ac.uk. IT Innovation, University of Southampton, SO16 7NP, UK.

[2] [km,phl]@ecs.soton.ac.uk. Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK.

Abstract

Artiste is a European project developing a cross-collection search system for art galleries and museums. It combines image content retrieval with text based retrieval and uses RDF mappings in order to integrate diverse databases. The test sites of the Louvre, Victoria and Albert Museum, Uffizi Gallery and National Gallery London provide their own database schema for existing metadata, avoiding the need for migration to a common schema. The system will accept a query based on one museum's fields and convert them, through an RDF mapping into a form suitable for querying the other collections. The nature of some of the image processing algorithms means that the system can be slow for some computations, so the system is session-based to allow the user to return to the results later. The system has been built within a J2EE/EJB framework, using the Jboss Enterprise Application Server.

Keywords: RDF, Interoperability, Multimedia, Content-based Retrieval

Word Count: 2672

Introduction

The Artiste project is a European Commission funded collaboration, investigating the use of integrated content and metadata-based image retrieval across disparate databases in several major art galleries across Europe. Collaborating galleries include the Louvre in Paris, the Victoria and Albert Museum in London, the Uffizi Gallery in Florence and the National Gallery in London. A key aim is to make a unified retrieval system which is targeted to users' real requirements and which is usable with integrated cross-collection searching. Museums and Galleries often have several digital collections ranging from public access images to specialised scientific images used for conservation purposes. Access from one gallery to another was not common in terms of textual data and not done at all in terms of image-based queries. However the value of cross-collection access is recognised as important for example in comparing treatments and conditions of paintings. While ARTISTE is primarily designed for inter-museum searching it could equally be applied to museum intranets. Within a Museum's intranet there may be systems which are not interlinked due to local management issues.

Previous European research projects on Art such as Van Eyck [1] and Aquarelle [2] used a standard metadata format or Z39.50 [3] interface to integrate Art collections. In Artiste we maintain the individual database schemas of the galleries but search across them by using an RDF mapping [4]. This mapping refers to common metadata schemas such as, but not limited to, Dublin Core. A key innovation in ARTISTE is using this approach to provide versatile content and metadata based retrieval and navigation facilities both within and between gallery collections. The distributed architecture gives galleries control over their own collections but the unified approach to retrieval and navigation simplifies and accelerates the processes.

Content-based image retrieval is really in its infancy and most researchers recognise its general use primarily in conjunction with metadata based retrieval [5] Various systems have been described in the literature from the early and pioneering IBM QBIC system [6] to the web based Webseek system from Columbia University[7]. Commercial systems offering content-based techniques are now available such as those from Virage Inc [8], but it is notable that the majority of web search engines offering image retrieval facilities still base them on text-based queries.

In our system, image content descriptors (feature vectors) are integrated with the text metadata for each object. Some of the descriptors have been developed to enable the system to answer specific types of query which were identified as of particular importance by the museum partners themselves.

System Definition

The ARTISTE architecture is illustrated in Figure 1. Each site has a complete instance of the ARTISTE system with its own images and metadata.

Images of the art objects from a collection are held in an object relational database from NCR (TOR: Teradata Object Relational) as Binary Large Objects (BLOBS). A number of user-defined modules (UDMs) define the image processing functions that can be applied to these representations. The UDMs are executed in the database, which has several benefits. The system uses parallelism automatically, which is important in this application where large volumes of data have to be processed by sometimes complex algorithms. Keeping the processing close to the data minimises data transfers, which is essential again due to the size of the data collections. The image processing functions can be called directly from SQL, allowing complex queries involving image processing and text metadata. The maintainability of the system is increased due to the DBMS functionality.Metadata is also stored in the database. This may be loaded from legacy systems or generated directly by the Artiste system, according to the incumbent systems at the museum sites. For example the field "Ecole" in the Louvre database will still be called "Ecole" in the ARTISTE database.

The ARTISTE architecture follows a standard n-tier approach.The server has been implemented within a J2EE/EJB framework [9], using the JBoss [10] Enterprise Application Server. This provides a robust, portable and scalable framework which emphasises modularity resulting in a more maintainable system. The framework provides system services such as database connection pooling which reduces the development time.

A user accesses the system from a desktop machine using a standard web browser. The web pages are generated in the presentation layer using JSP and JavaServlet technologies with a standard model-view-controller approach. The distributed query and metadata layer provides a single interface to the art and its metadata and facilities to enable queries to be directed towards multiple distributed databases. This enables the end user to seamlessly search the combined art collections. The distributed query layer is implemented as a web service using XML messaging and SOAP as the transport protocol.

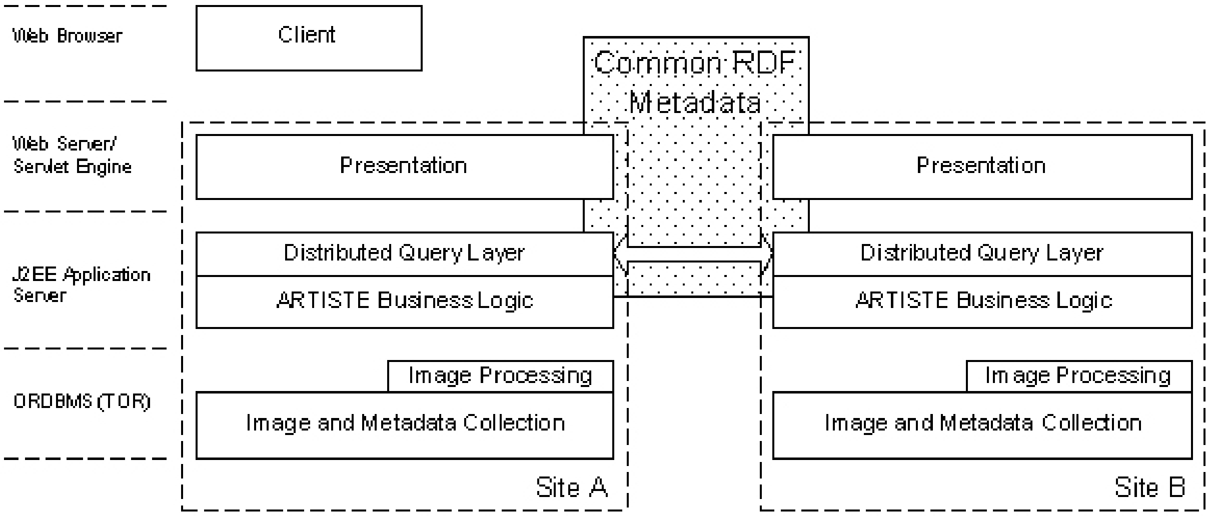


Figure 1 Artiste Architecture

The ARTISTE Server accesses the image processing functions within the ORDMBS using the ARTISTE Image Processing API. The API is a SQL interface for user defined image processing functions that enables interoperability with ORDBMS other than TOR. The requirements for interoperability are that the ORDBMS supports user defined functions that operate on complex objects such as images that can be invoked directly within SQL statements.

Interoperability

The ARTISTE system enables queries to be executed across multiple, distributed collections without requiring each collection to conform to a standard schema. This is achieved by using resource description format (RDF) [11][12] to define the syntax and semantics for standard metadata terms. Each collection provides a mapping that relates these standard metadata terms to individual database table and column values. Queries are composed using RDF, and subsequently translated to SQL at each site (Figure 2).

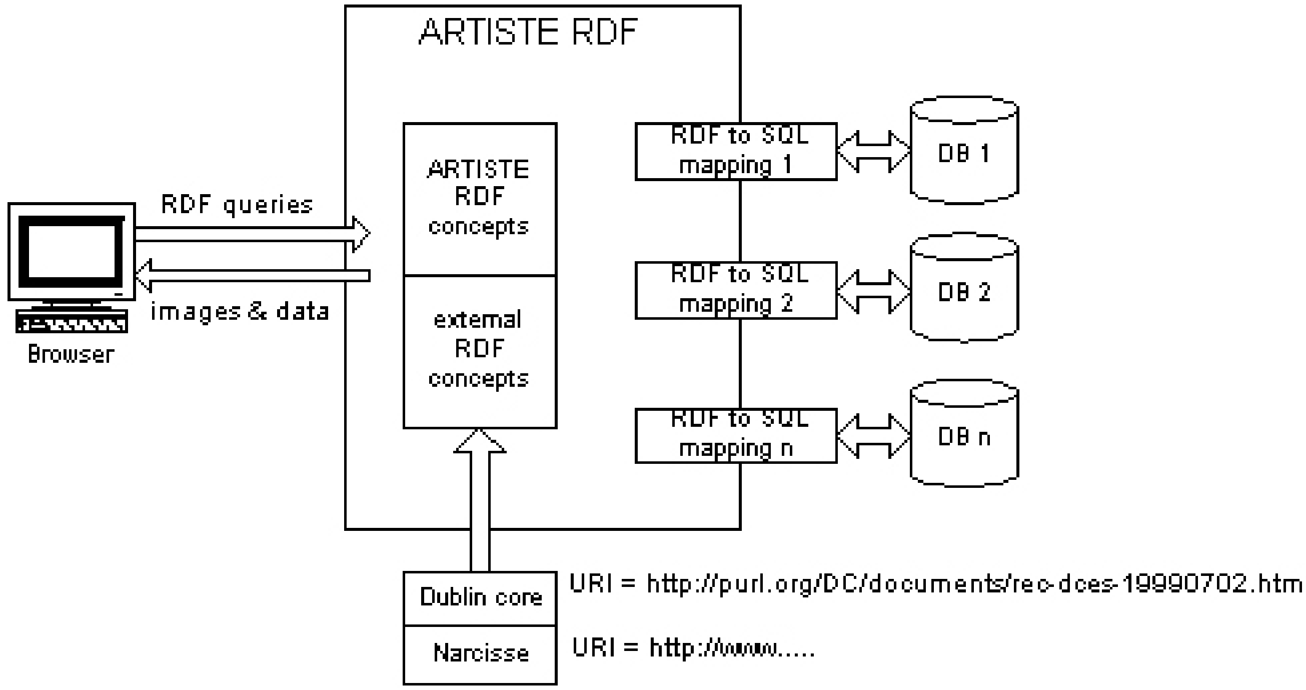


Figure 2 Interoperable RDF approach

A example set of metadata terms used for describing documents is defined by the Dublin Core (DC) RDF schema [13]. Figure 3 shows how DC is used within the ARTISTE system to provide query interoperability between collections when querying using the DC Title standard metadata term. Each collection maps the DC Title metadata term to columns in metadata tables. In this case Collection A defines image title within the Caption column and Collection B defines image title within the TitleM column. A user can then execute a single query across both collections and at each collection the RDF URI for DC Title is translated to the SQL for selecting Caption or TitleM.

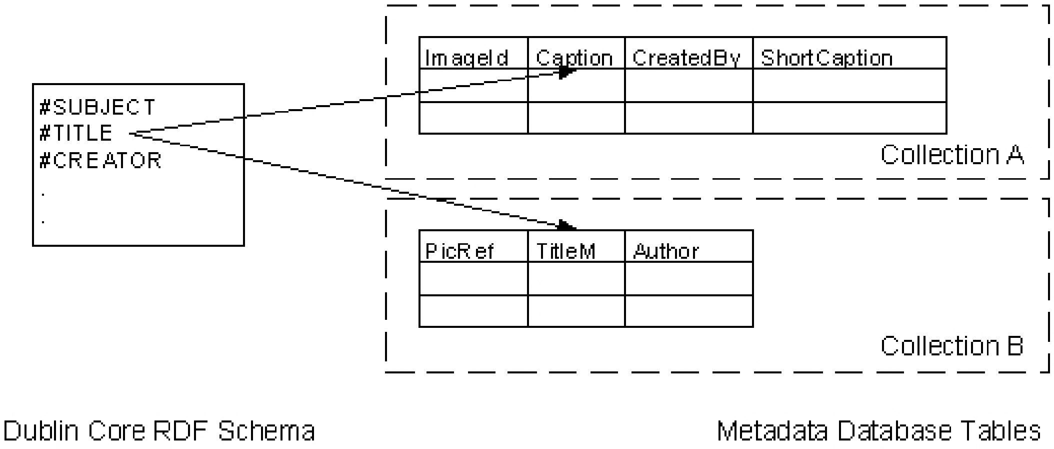


Figure 3 Dublin Core Schema Interoperabilty

As it is the responsibility of each site to define which metadata terms are supported, it is unlikely that collections will provide the same querying capabilities. For example, each collection may only support a subset of the Dublin Core and/or additional terms defined in other standard metadata schemas. To ensure interoperability the ARTISTE system generates a Query Context when a user builds a new query based upon the collections they wish to search. The Query Context contains a union of query capabilties supported by the selected collections.

Issues

Current digital library query representations and protocols, such as Z39.50, deal entirely with textual metadata. This is not sufficient for multimedia digital libraries such as ARTISTE, where searches can be made on image content as well as textual metadata. In particular, current protocols have the following restrictions:

- There are no methods for specifying image content as a metadata item.
- There are no operators defined relating to image content.
- There are no methods for carrying out searches that result in the execution of image processing algorithms.

These issues are addressed in the following sections.

Specifying image content

The ARTISTE approach provides a seamless way to query image collections. Firstly, we define a base term for all aspects that can be queried: a QueryItem. This can be the image itself, properties of the image (such as colour or shape), or attributes associated with the image (conventional metadata such as textual and numeric items). Figure 4 shows the current query item structure.

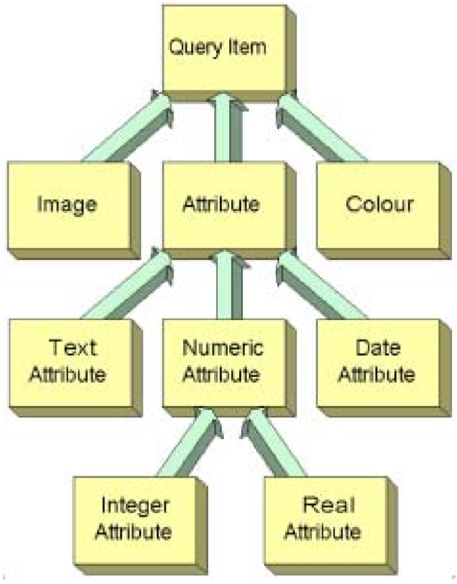


Figure 4 ARTISTE Query Item structure

The RDF schema for the ARTISTE query items is defined in Appendix A1.

Query Operators

We define a Query Operator as an abstract operation that can be performed on query items. Figure 5 shows the current query operators. These include exact operators (such as equals, less than etc.) and fuzzy operators (such as similar to).

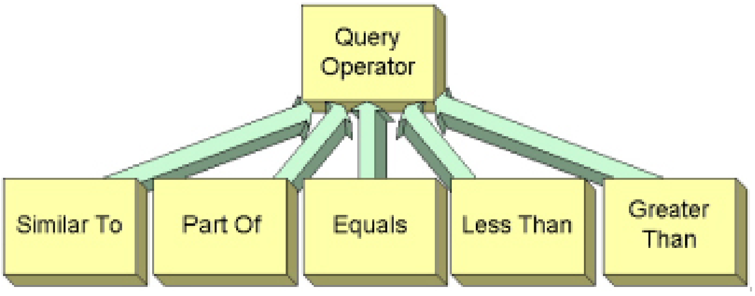


Figure 5 ARTISTE Query Operator structure

The RDF schema for the ARTISTE query operators is defined in Appendix A2.

Query Expression Rules

All operators must be instantiated to form rules – each rule describes a legal ARTISTE query expression and specifies which query items can be used with each query operator. Each query operator is assumed to take two objects: a subject and an object.

The following properties are defined:

- **AllowedSubject –** this property is used by the rules to define the query items that a particular operator can take. The property can only be applied to a query expression rule, and must refer to a query item.
- **AllowedObject –** this property defines the allowed object of a query expression rule, in a similar way to the allowed subject property.

Figure 6 shows a diagrammatic representation of the "Text Equals" query expression rule. This rule defines "Text Equals" as an operation involving two text attributes and the equality operator.

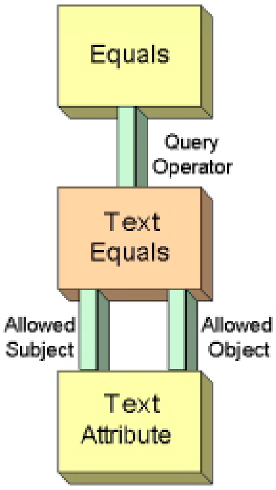


Figure 6 Simple Query Expression Rule

Other rules can be defined as appropriate (DateEquals, TextContains etc.) Rules that contain fuzzy operators must further be qualified by relating an analyser, as described in the following section. The RDF schema for the ARTISTE query expression rules is defined in Appendix A3.

Image Processing

We define an algorithm as being a software module that operates on an image to generate or compare image feature vectors. An analyser is an algorithm together with the relevant metadata which describes how the algorithm interacts with the rest of the system. Analysers may be used in conjunction with fuzzy operators (e.g. similar to) or may be used to generate conventional metadata terms.

In ARTISTE our approach has been to develop a range of algorithms producing feature vectors, which are typically pre-computed from the images and stored in the database, augmenting their metadata. Some general content-based techniques based on texture, colour and spatial colour distributions have been implemented but the main thrust has been on additional novel retrieval algorithms addressing some specific needs of the galleries. There are two ways that the image processing algorithms are used. One is to carry out a comparison of images in order to find for example images which have similar colour layout. The second is to run algorithms once in order to populate metadata, for example with a count of wooden supports or a shape classification.

Some of the images in the collection are very large (up to 800 Mbytes) and also very high resolution (20 pels/mm) demanding special purpose algorithms for effective handling. We use a Java applet viewer [14] to browse large images in the user interface. One new algorithm takes a query image which may be a sub-image of an image within the database and which may be recorded at a significantly different resolution from its parent. The requirement is to identify from which parent image the query is derived and to locate its position in the parent image. Since the query may have been captured at a different time, possibly in a different state of restoration, or simply under different lighting conditions, robust techniques are required. A multi-scale technique based on colour coherence vectors has been developed for the task, which is giving useful results [15].

Another algorithm has been designed to detect and classify image boundaries, which enables frame shape classification and another technique is targeted at low quality queries, in particular faxes, which often provide the means by which queries are submitted to galleries. A further algorithm detects features in the restoration framework as some images in the collection are of the backs of paintings showing how the framework has been restored. These techniques are typically combined with a metadata based search which limits the content based search to a sub-class of the total image collection. Development of these specialised algorithms is ongoing and a user group is able to access the emerging prototypes and provide feedback to the research teams.

The analysers which calculate the feature vectors must be applied to query expression rules for them to be used by the system. This is achieved by using the property **AnalyserAppliesTo** to map the analysers to the query expression rules to which they apply (e.g. it may be used to say that the Colour Histogram analyser is used to find similar images). Figure 7 shows the query expression rule using the fuzzy operator "Similar To" and qualified with the Colour Histogram analyser.

The ability to author hypertext links between database objects and other parts of the information space is also being incorporated [16]. These links are generic, in the sense that their source anchors and destinations are recorded in the database, and can be followed from any instance of the source anchor.

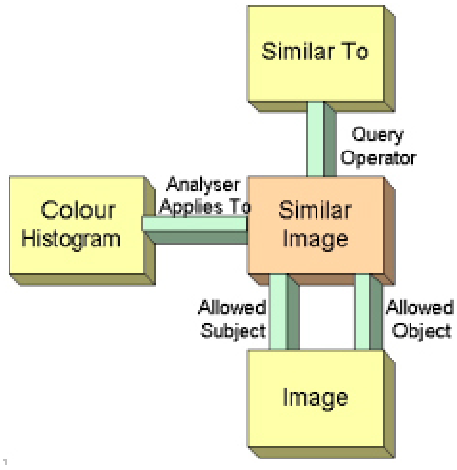
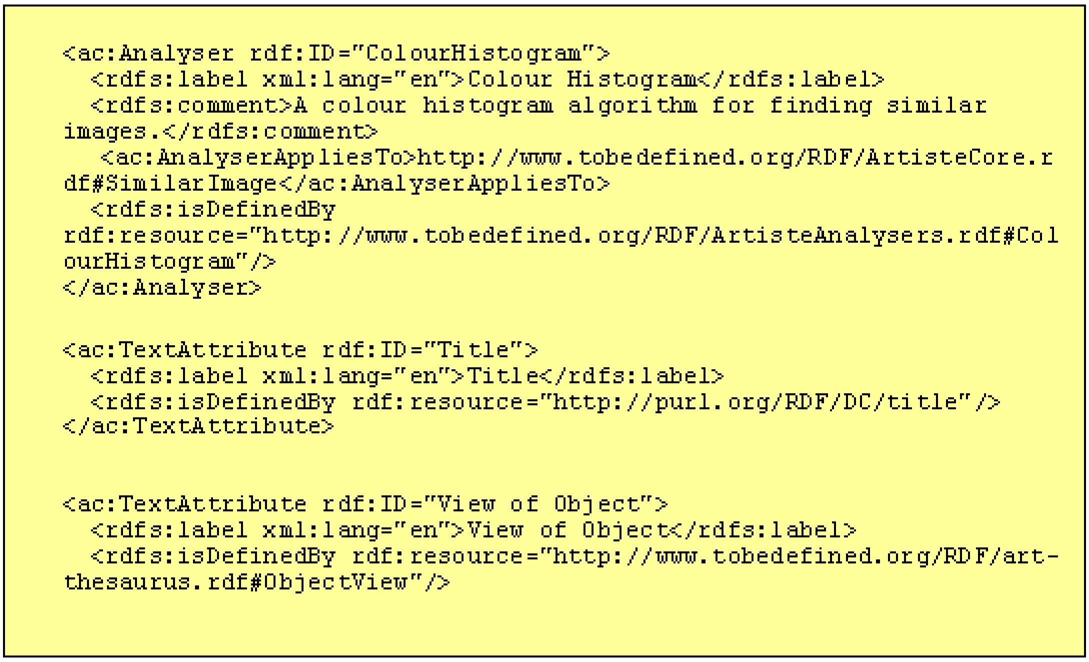


Figure 7 Fuzzy Query Expression Rule

The RDF schema for the ARTISTE analysers is defined in Appendix A4.

Examples

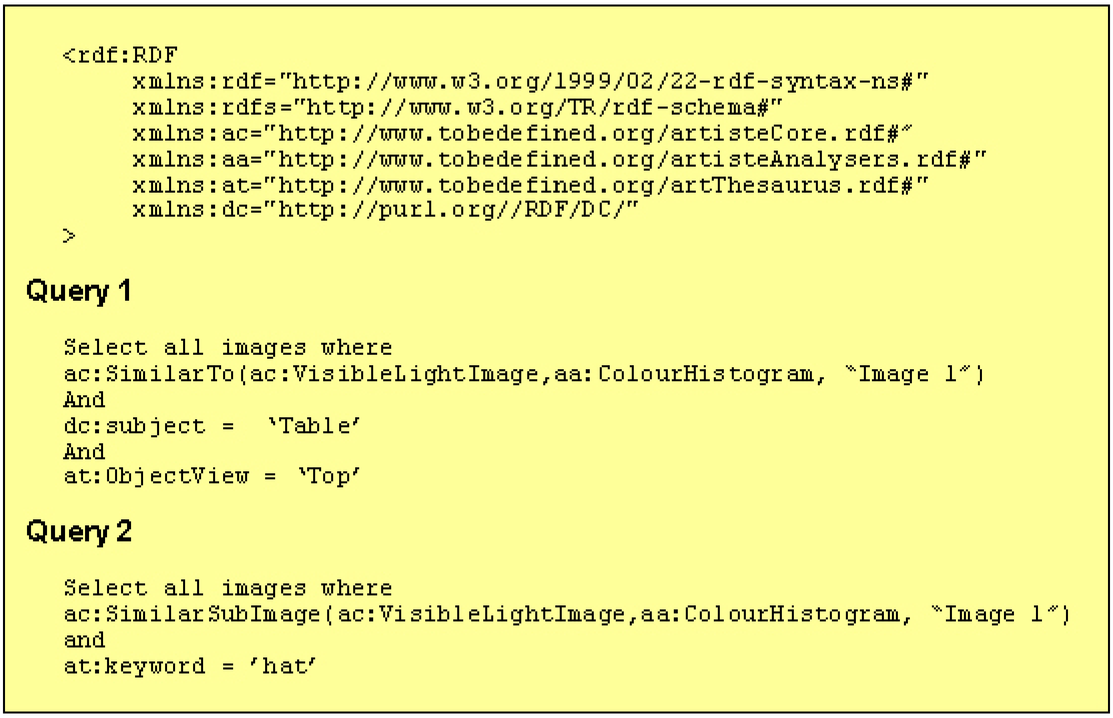
We now present an example set-up and query to show how the RDF schema used in ARTISTE can provide an interoperable framework across distributed sites. Figure 8 shows an extract from the RDF an application profile [17] which described the site. The site declares that it supports the ‘Colour Histogram’ algorithm (as defined by the published ArtisteAnalysers RDF file), the title (as defined by Dublin Core) and the view of the object (as defined by a hypothetical art thesaurus).



```
<ac:Analyser rdf:ID="ColourHistogram">
  <rdfs:label xml:lang="en">Colour Histogram</rdfs:label>
  <rdfs:comment>A colour histogram algorithm for finding similar
images.</rdfs:comment>
   <ac:AnalyserAppliesTo>http://www.tobedefined.org/RDF/ArtisteCore.r
df#SimilarImage</ac:AnalyserAppliesTo>
  <rdfs:isDefinedBy
rdf:resource="http://www.tobedefined.org/RDF/ArtisteAnalysers.rdf#Col
ourHistogram"/>
</ac:Analyser>

<ac:TextAttribute rdf:ID="Title">
  <rdfs:label xml:lang="en">Title</rdfs:label>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/RDF/DC/title"/>
</ac:TextAttribute>


<ac:TextAttribute rdf:ID="View of Object">
  <rdfs:label xml:lang="en">View of Object</rdfs:label>
  <rdfs:isDefinedBy rdf:resource="http://www.tobedefined.org/RDF/art-
thesaurus.rdf#ObjectView"/>
```

Figure 8 Example: extract from site application profile

The pseudo code for example queries is shown in Figure 9. Query 1 searches for images of tabletops that are similar to "Image 1". Query 2 searches for sub-images where the keyword is hat.



```
   <rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/TR/rdf-schema#"
        xmlns:ac="http://www.tobedefined.org/artisteCore.rdf#"
        xmlns:aa="http://www.tobedefined.org/artisteAnalysers.rdf#"
        xmlns:at="http://www.tobedefined.org/artThesaurus.rdf#"
        xmlns:dc="http://purl.org//RDF/DC/"
   >
```

Query 1

```
   Select all images where
   ac:SimilarTo(ac:VisibleLightImage,aa:ColourHistogram, “Image 1”)
   And
   dc:subject =  ‘Table’
   And
   at:ObjectView = ‘Top’
```

Query 2

```
   Select all images where
   ac:SimilarSubImage(ac:VisibleLightImage,aa:ColourHistogram, “Image 1”)
   and
   at:keyword = ’hat’
```

Figure 9 Example query pseudo code

Figure 10 shows the query image for ‘Query 2’, as described above. Note that the query image is a sub-image of the required result and that some of the images in the test collection were as large as 13 MBytes. This query was executed against a database of about 50000 images, of which 500 had the keyword ‘hat’. The execution time for this query was about one minute on a 4x450MHz Windows NT platform. No feature vector indexing has been implemented yet, which will significantly improve the speed of retrieval. The results can be seen in Figure 11. It can be seen that the system has correctly retrieved the corresponding image, and has found an extremely good match for a second image.

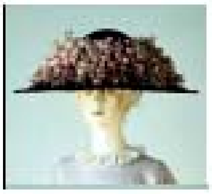
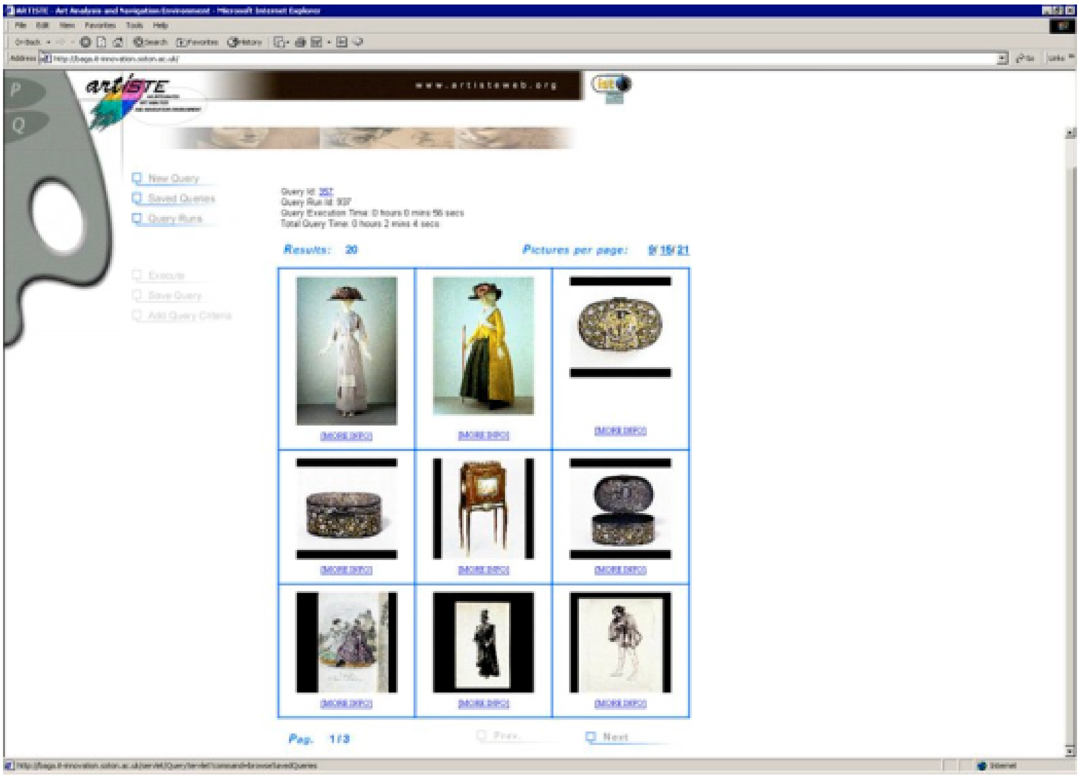


Figure 10 Query image



Query Id: 357
Query Run Id: 937
Query Execution Time: 0 hours 0 mins 56 secs
Total Query Time: 0 hours 2 mins 4 secs

Results: 20

Pictures per page: 9/ 15/ 21

Pag. 1/3

Figure 11 Results of Query

Conclusions

The use of RDF mapping has provided a more flexible solution to cross-collection searching. Its flexibility has allowed us to design a system which also handles content based retrieval. The query language for this type of system is not yet standardised but we hope that an emerging standard will provide the session-based connectivity this application seems to require due to the possibility of long query times.

The content-based retrieval is fast enough to be usable even without multidimensional indexing, although that will speed up the system considerably when it is implemented. It was found that the targeted content-based algorithms provide more successful results than applying generic content-based algorithms in isolation.

The ARTISTE system provides a very flexible framework to provide content and metadata-based queries across multiple sites.

Future Work

In the near future, the project will be introducing controlled vocabulary support for some of the metadata fields. This will not only make retrieval more robust but will also facilitate query expansion. The Louvre’s multilingual thesaurus will be used in order to ensure greater interoperability.

The system is easily extensible to other multimedia types such as audio and video (eg by adding additional query items such as "dialog" and "video sequence" with appropriate analysers). A follow-up project is scheduled to explore this further.

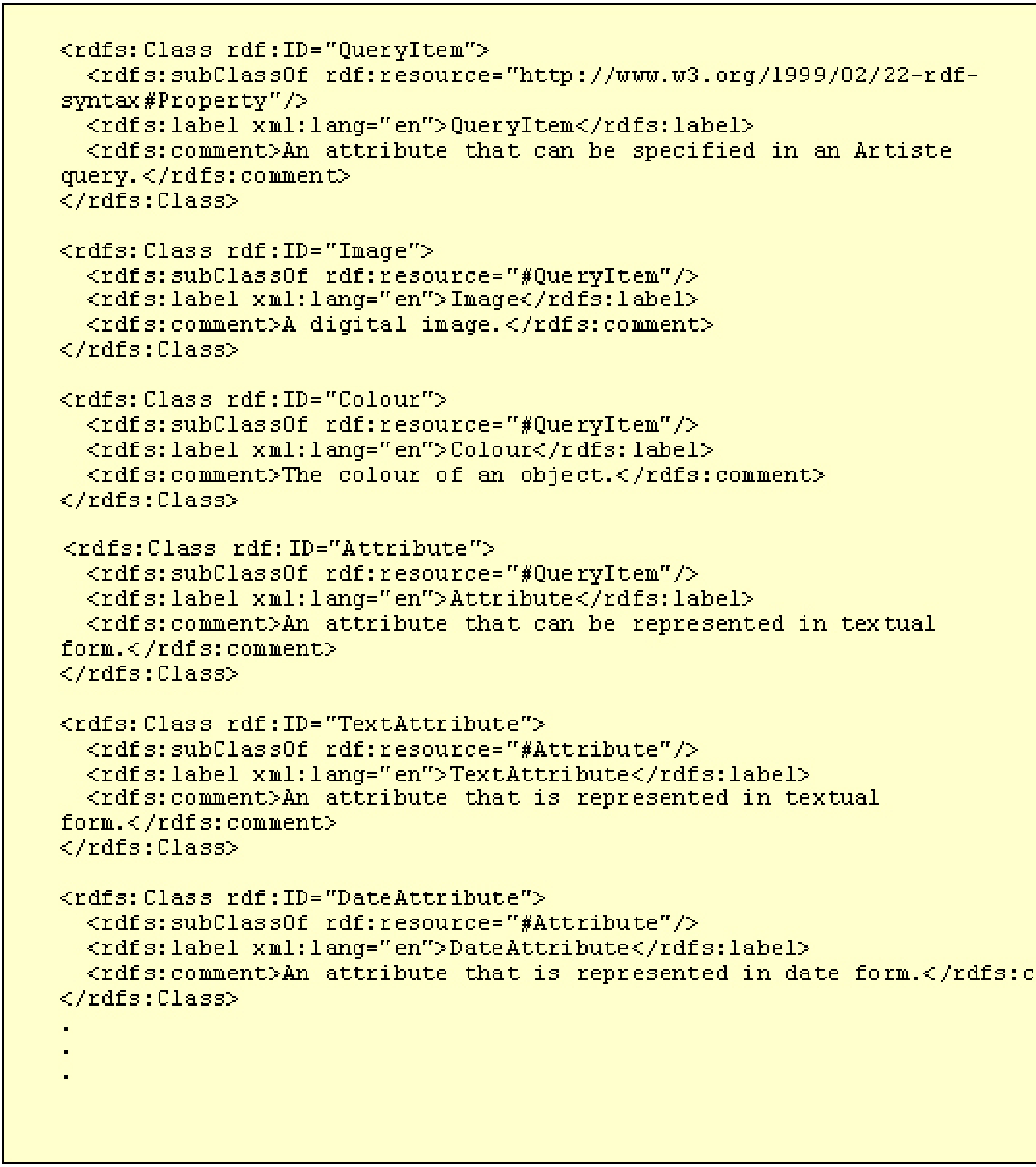
There is some scope for relating our RDF query format to the emerging query standards such as XQuery [18] and we also plan to feed our experience into standards such as the ZNG initiative.

Acknowledgements

Artiste is funded by the European Community's Framework 5 programme. The partners are: NCR, The University of Southampton, IT Innovation, Giunti Multimedia, The Victoria and Albert Museum, The National Gallery, The research laboratory of the museums of France (C2RMF) and the Uffizi Gallery. We would particularly like to thank our collaborators Christian Lahanier, James Stevenson, Marco Cappellini, John Cupitt, Raphaela Rimabosci, Gert Presutti, Warren Stirling, Fabrizio Giorgini and Roberto Vacaro.

Appendix A RDF

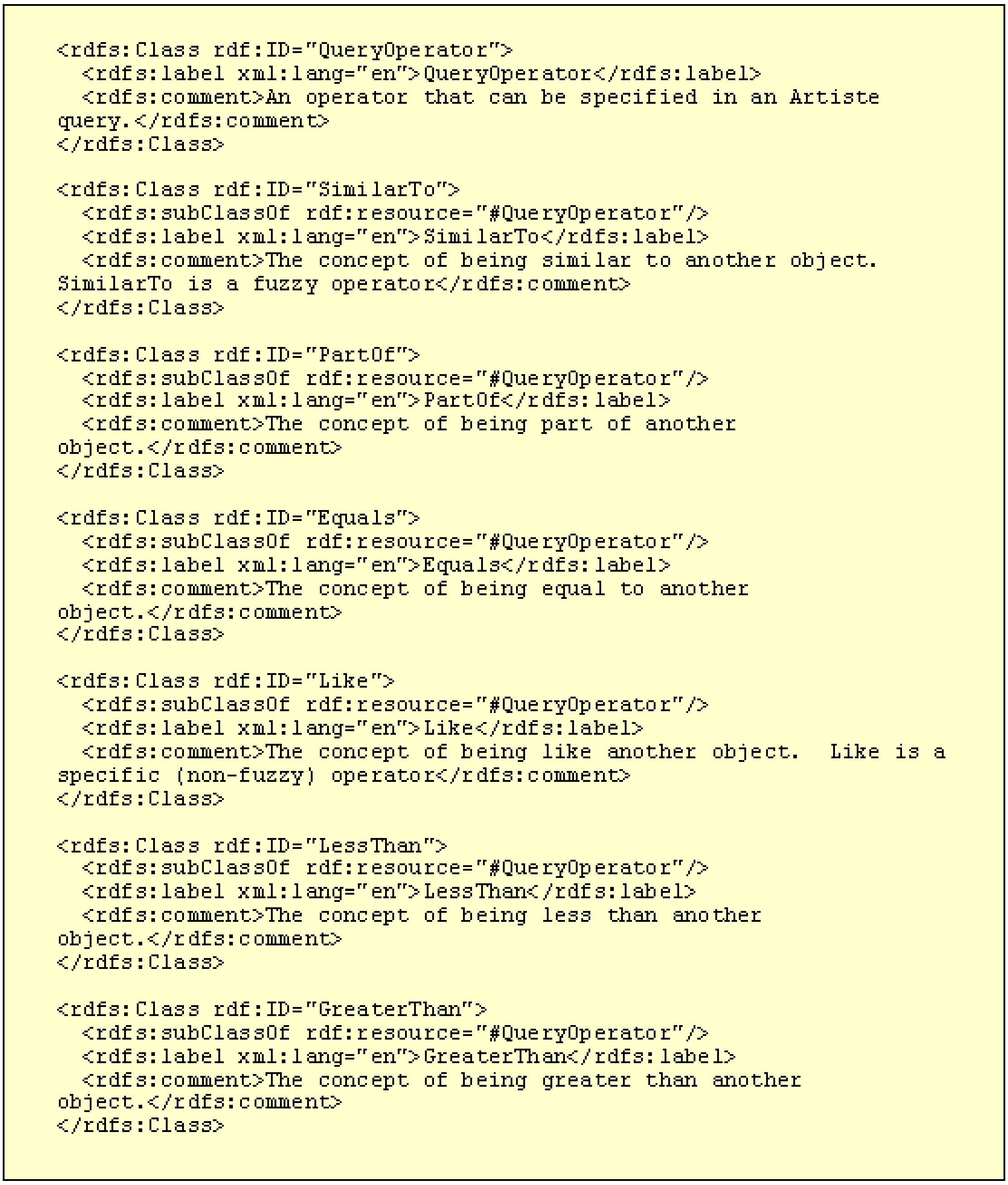
Appendix A1 Query Item RDF



```
<rdfs:Class rdf:ID="QueryItem">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax#Property"/>
  <rdfs:label xml:lang="en">QueryItem</rdfs:label>
  <rdfs:comment>An attribute that can be specified in an Artiste
query.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Image">
  <rdfs:subClassOf rdf:resource="#QueryItem"/>
  <rdfs:label xml:lang="en">Image</rdfs:label>
  <rdfs:comment>A digital image.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Colour">
  <rdfs:subClassOf rdf:resource="#QueryItem"/>
  <rdfs:label xml:lang="en">Colour</rdfs:label>
  <rdfs:comment>The colour of an object.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Attribute">
  <rdfs:subClassOf rdf:resource="#QueryItem"/>
  <rdfs:label xml:lang="en">Attribute</rdfs:label>
  <rdfs:comment>An attribute that can be represented in textual
form.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="TextAttribute">
  <rdfs:subClassOf rdf:resource="#Attribute"/>
  <rdfs:label xml:lang="en">TextAttribute</rdfs:label>
  <rdfs:comment>An attribute that is represented in textual
form.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="DateAttribute">
  <rdfs:subClassOf rdf:resource="#Attribute"/>
  <rdfs:label xml:lang="en">DateAttribute</rdfs:label>
  <rdfs:comment>An attribute that is represented in date form.</rdfs:c
</rdfs:Class>
.
.
.
```

Figure 12 ARTISTE Query Item RDF

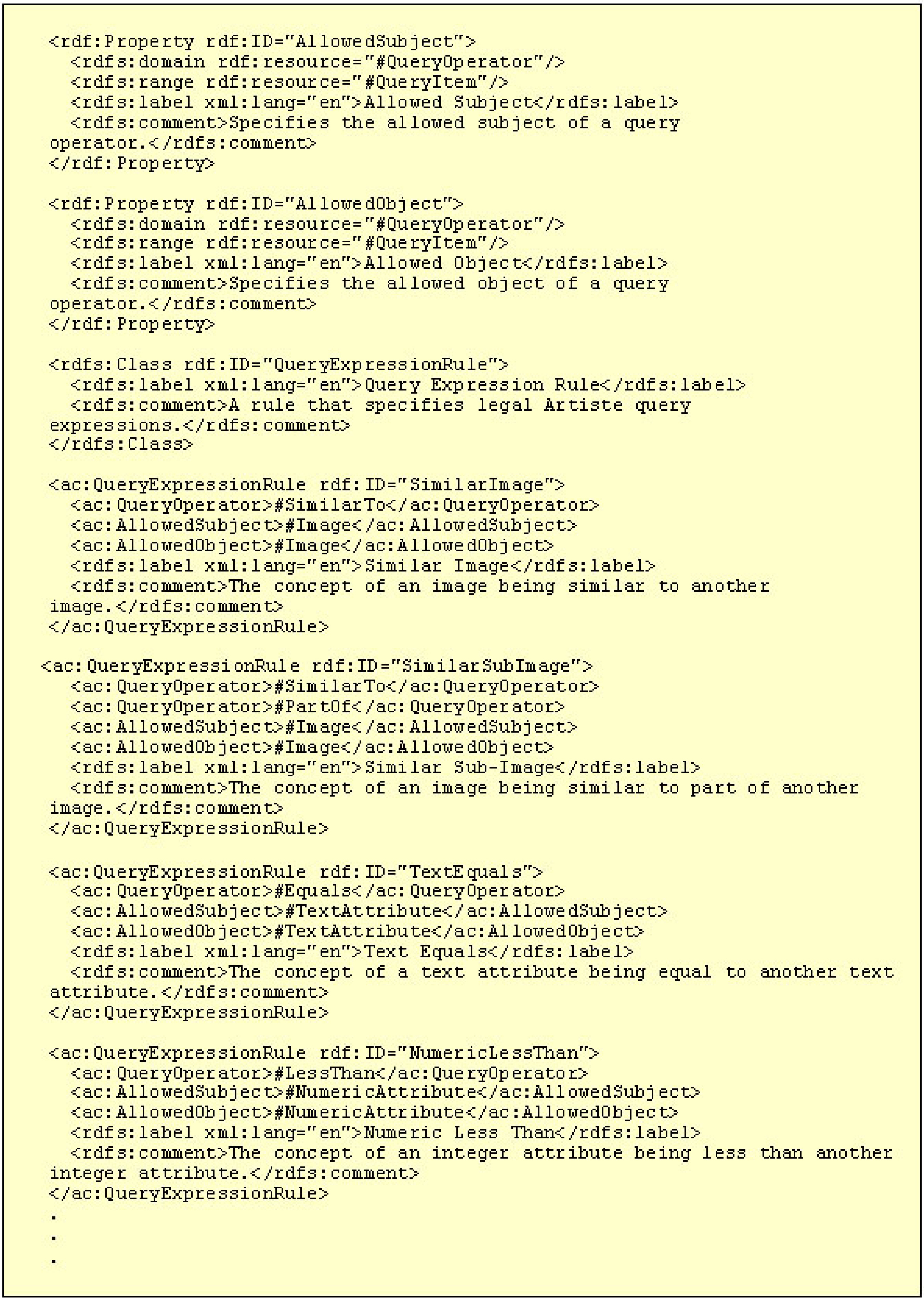
Appendix A2 Query Operator RDF



```
<rdfs:Class rdf:ID="QueryOperator">
  <rdfs:label xml:lang="en">QueryOperator</rdfs:label>
  <rdfs:comment>An operator that can be specified in an Artiste
query.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="SimilarTo">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">SimilarTo</rdfs:label>
  <rdfs:comment>The concept of being similar to another object.
SimilarTo is a fuzzy operator</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="PartOf">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">PartOf</rdfs:label>
  <rdfs:comment>The concept of being part of another
object.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Equals">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">Equals</rdfs:label>
  <rdfs:comment>The concept of being equal to another
object.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Like">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">Like</rdfs:label>
  <rdfs:comment>The concept of being like another object.  Like is a
specific (non-fuzzy) operator</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="LessThan">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">LessThan</rdfs:label>
  <rdfs:comment>The concept of being less than another
object.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="GreaterThan">
  <rdfs:subClassOf rdf:resource="#QueryOperator"/>
  <rdfs:label xml:lang="en">GreaterThan</rdfs:label>
  <rdfs:comment>The concept of being greater than another
object.</rdfs:comment>
</rdfs:Class>
```

Figure 13 ARTISTE Query Operator RDF

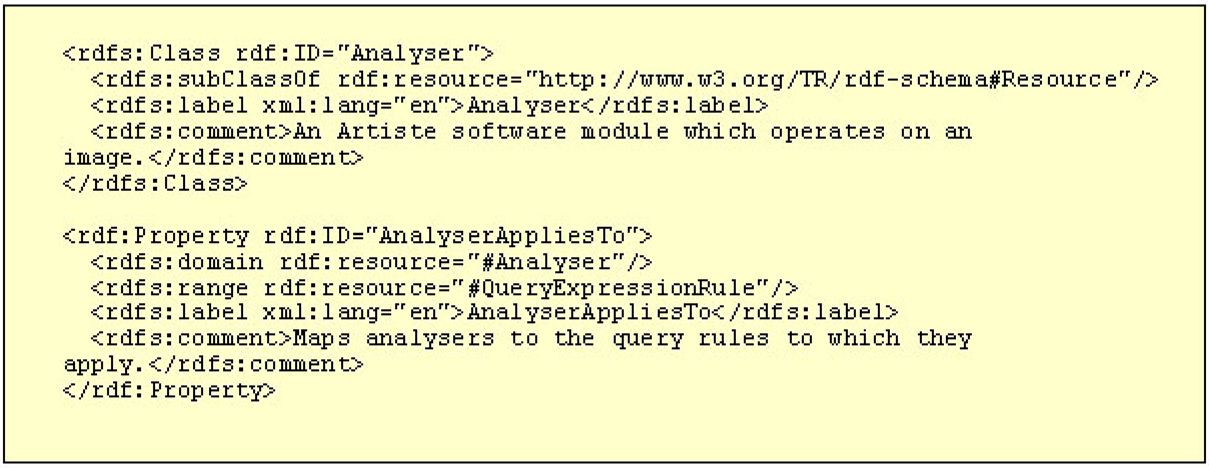
Appendix A3 Query Expression Rules RDF



```
<rdf:Property rdf:ID="AllowedSubject">
  <rdfs:domain rdf:resource="#QueryOperator"/>
  <rdfs:range rdf:resource="#QueryItem"/>
  <rdfs:label xml:lang="en">Allowed Subject</rdfs:label>
  <rdfs:comment>Specifies the allowed subject of a query
operator.</rdfs:comment>
</rdf:Property>

<rdf:Property rdf:ID="AllowedObject">
  <rdfs:domain rdf:resource="#QueryOperator"/>
  <rdfs:range rdf:resource="#QueryItem"/>
  <rdfs:label xml:lang="en">Allowed Object</rdfs:label>
  <rdfs:comment>Specifies the allowed object of a query
operator.</rdfs:comment>
</rdf:Property>

<rdfs:Class rdf:ID="QueryExpressionRule">
  <rdfs:label xml:lang="en">Query Expression Rule</rdfs:label>
  <rdfs:comment>A rule that specifies legal Artiste query
expressions.</rdfs:comment>
</rdfs:Class>

<ac:QueryExpressionRule rdf:ID="SimilarImage">
  <ac:QueryOperator>#SimilarTo</ac:QueryOperator>
  <ac:AllowedSubject>#Image</ac:AllowedSubject>
  <ac:AllowedObject>#Image</ac:AllowedObject>
  <rdfs:label xml:lang="en">Similar Image</rdfs:label>
  <rdfs:comment>The concept of an image being similar to another
image.</rdfs:comment>
</ac:QueryExpressionRule>

<ac:QueryExpressionRule rdf:ID="SimilarSubImage">
  <ac:QueryOperator>#SimilarTo</ac:QueryOperator>
  <ac:QueryOperator>#PartOf</ac:QueryOperator>
  <ac:AllowedSubject>#Image</ac:AllowedSubject>
  <ac:AllowedObject>#Image</ac:AllowedObject>
  <rdfs:label xml:lang="en">Similar Sub-Image</rdfs:label>
  <rdfs:comment>The concept of an image being similar to part of another
image.</rdfs:comment>
</ac:QueryExpressionRule>

<ac:QueryExpressionRule rdf:ID="TextEquals">
  <ac:QueryOperator>#Equals</ac:QueryOperator>
  <ac:AllowedSubject>#TextAttribute</ac:AllowedSubject>
  <ac:AllowedObject>#TextAttribute</ac:AllowedObject>
  <rdfs:label xml:lang="en">Text Equals</rdfs:label>
  <rdfs:comment>The concept of a text attribute being equal to another text
attribute.</rdfs:comment>
</ac:QueryExpressionRule>

<ac:QueryExpressionRule rdf:ID="NumericLessThan">
  <ac:QueryOperator>#LessThan</ac:QueryOperator>
  <ac:AllowedSubject>#NumericAttribute</ac:AllowedSubject>
  <ac:AllowedObject>#NumericAttribute</ac:AllowedObject>
  <rdfs:label xml:lang="en">Numeric Less Than</rdfs:label>
  <rdfs:comment>The concept of an integer attribute being less than another
integer attribute.</rdfs:comment>
</ac:QueryExpressionRule>
.
.
.
```

Figure 14 ARTISTE Query Expression Rules RDF

Appendix A4 Analyser RDF



```
<rdfs:Class rdf:ID="Analyser">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/rdf-schema#Resource"/>
  <rdfs:label xml:lang="en">Analyser</rdfs:label>
  <rdfs:comment>An Artiste software module which operates on an
image.</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="AnalyserAppliesTo">
  <rdfs:domain rdf:resource="#Analyser"/>
  <rdfs:range rdf:resource="#QueryExpressionRule"/>
  <rdfs:label xml:lang="en">AnalyserAppliesTo</rdfs:label>
  <rdfs:comment>Maps analysers to the query rules to which they
apply.</rdfs:comment>
</rdf:Property>
```

Figure 15 Analyser RDF

References

[1] J. H. E. van der Starre, "Visual Arts Network for the exchange of cultural knowledge: Van Eyck Project", Information Services & Use, Vol.13, pp 347-355, 1993.
[2] A. Michard, V. Christophides, M. Scholl, M. Stapleton, D. Sutcliffe, A.M. Vercoustre, "The Aquarelle resource discovery system", Computer Networks and ISDN Systems, Vol. 30, 1185-1200, 1998.
[3] National Information Standards Organisation, Z39.50 Information Retrieval Protocol http://www.niso.org/z3950.html, 1998.
[4] J. Hunter, C. Lagoze, "Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles", Computer Networks, Proceedings of the tenth Web conference, 2000.
[5] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, "Content-Based Image Retrieval at the end of the Early Years", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, pp.1349-1380, 2000
[6] M. Flickner, H. Sawhney, W. Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P., "Query by image and video content: The QBIC system", IEEE Computer Magazine, vol. 28, pp. 23-32, September 1995
[7] S. F. Chang, J. R. Smith, M. Beigi, and A. Benitez, "Visual Information Retrieval from Large Distributed On-Line Repositories", Communications of ACM, Special Issue on Visual Information Management, Vol. 40 No. 12, pp. 63-71, Dec. 1997.
[8] http://www.virage.com
[9] N. Kassem, "Designing Enterprise Applications with the Java(TM) 2 Platform, Enterprise Edition", Addison Wesley, 2000.
[10] www.jboss.org
[11] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/
[12] Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000, http://www.w3.org/TR/rdf-schema/
[13] Dublin Core Metadata Initiative http://www.dublincore.org/ [14] K. Martinez, J. Cupitt, and S. Perry. "High resolution Colorimetric Image Browsing on the Web", Computer Networks and ISDN Systems. Vol. 30, 1998, pp. 399 - 405.
[15] S. Chan, K. Martinez, P. Lewis, C. Lahanier and J. Stevenson, "Handling Sub-Image Queries in Content-Based Retrieval of High Resolution Art Images", International Cultural Heritage Informatics Meeting pp.157-163, 2001.
[16] P. H. Lewis, R. J. Wilkins, S. R. Griffiths, H. C. Davis and W. Hall, "Content Based Navigation in an Open Hypermedia Environment". Image and Vision Computing, Vol.16 pp.921-929, 1998.

[17] R. Heery, M. Patel, "Application Profiles: mixing and matching metadata schemas", Ariadne, Issue 25, September 2000.

[18] XQuery 1.0: An XML Query Language, W3C Working Draft 07 June 2001. http://www.w3.org/TR/xquery/