Data conversion, extraction and record linkage using XML and RDF tools in Project SIMILE

Mark H. Butler (mark-h_butler@hp.com), John Gilbert, Andy Seaborne, Kevin Smathers
Hewlett Packard Laboratories, Bristol UK
16 August 2004

Abstract

SIMILE is a joint project between MIT Libraries, MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), HP Labs and the World Wide Web Consortium (W3C). It is investigating the application of Semantic Web tools, such as the Resource Description Framework (RDF), to the problem of dealing with heterogeneous metadata. This report describes how XML and RDF tools are used to perform data conversion, extraction and record linkage on some sample datasets featuring visual images (ARTstor) and learning objects (OpenCourseWare) in the first SIMILE proof of concept demo.

Keywords

SIMILE, RDF, Semantic Web, thesauri, data conversion, extraction, screen scraping, record linkage

1 Introduction

SIMILE[1] is a research project investigating how to extend DSpace[2], a digital asset management system developed by Hewlett-Packard Laboratories and MIT Libraries. SIMILE aims to produce an architecture for information search and retrieval across heterogeneous metadata that describes collections of resources from disparate domains. It is explicitly tasked with investigating the application of Semantic Web tools to the problem of dealing with heterogeneous metadata: such metadata is normally stored in separate systems, each of which has an associated cost for construction, maintenance and comprehension, and users cannot query these systems concurrently with the results presented in a consistent fashion.

An early goal of the project was to build a demonstration prototype using a collection of datasets that described different domains but where there was some overlap. Initially two such datasets were obtained by MacKenzie Smith (MIT Libraries): one from ARTstor[3] originally created as part of the UCAI project[4] that describes visual images, and one from MIT OpenCourseWare[5] (OCW) that describes learning objects[6], i.e. modular digital resources that can be used to support learning. The ARTstor dataset is particularly high quality as access to it is commercially licensed. Additional information to enrich these datasets was obtained from the Wikipedia open encyclopaedia[7] and the prototype OCLC Library of Congress Name Authority Service[8].

This report describes how the two main datasets were modelled in RDF and how they were converted from XML to RDF[9] using XSLT 2[10]. It also explains how information was extracted from the Wikipedia and OCLC sources to augment these datasets. Next

the report describes how the Levenshtein distance measure[11] was used to identify record linkage in the different data sets. Finally, it describes how the results were displayed in a faceted browser. It concludes that XSLT 2 is a useful tool for converting XML to RDF/XML, that a key decision for conversion is which values should be replaced by class instances and that string similarity measures are applicable to automating the task of identifying mappings between datasets but that those mappings must be checked by a human user.

2 An overview of the data conversion process

As the two main data sets are originally encoded in XML, it was possible to use XSLT to transform the XML into RDF/XML. The RDF/XML versions of the ARTstor and OCW datasets may be built automatically and do not require any human review or correction. Accessing web services such as the OCLC authority service and screen scraping[12] web pages such as the Wikipedia is not possible in XSLT, so Java was chosen for these tasks. Here a human reviewer is required in order to choose between multiple alternatives or to remove data that is semantically incorrect, so this program outputs results as N3[13] rather than RDF/XML as it is less verbose and easier for human reviewers to read.

3 Generating RDF encodings of ARTstor and OCW

3.1 Selection of appropriate schemas

The first step towards encoding the ARTstor data as RDF was to create an appropriate collection of classes and properties. When converting a dataset from XML to RDF, it is often possible to increase the number of types of classes used to model the data. This is because some XML data uses containership exclusively as the only way to describe relations between objects, resulting in a collection of objects of a single type, which may subcomponents that correspond to other objects. However when converting this data to RDF, it is desirable to use different classes to represent each different type of entity in the data, as separating entities make it easier to identify multiple instances of the same entity as will be discussed in a later section. Here "entity" is used in the same sense as in the Functional Requirements of Bibliographic Records[14]. For example the original ARTstor dataset has a single class to represent works, which contains several instances of a media object class that are particular instances of the work. When this is converted to RDF, additional classes are added to represent creators and institutions, as well as subject, topic and geographic index terms.

VRAcore[15] is a standard designed for describing visual images that is used as the basis for the ARTstor XML schema. There is no standard XML or RDF serialisation for VRAcore, only a textual description that defines the meaning of terms used in the data model, so this dictionary was reviewed by the SIMILE developers as it provides additional explanation of some of the terms used in ARTstor. ARTstor also uses some non-VRAcore terms.

When creating the RDF Schema for the VRAcore dataset, the convention was adopted that if a property or class came from VRAcore, it was placed in a VRAcore namespace whereas ARTstor specific terms were placed in an ARTstor namespace. In the examples used in this report, the former is represented using XML namespace

prefix `vra` whereas the latter is represented using `art`. This approach was taken to simplify the future reuse of the VRAcore portion of the schema.

A number of additional schemas were also used: where appropriate, the SKOS[16] thesauri standard was used for encoding controlled vocabularies of subjects, topics, geographic place names, type of work and measurement, the VCard schema was used for naming individuals, and a new class, Person, associated with the prefix `person` was created to indicate an individual person. By combining these different schemas to represent the ARTstor data, we have effectively created an application profile [17] for ARTstor. For more details of the classes and properties used see Appendix A.

The OCW data on the other hand was represented using the RDF IMS LOM binding[18] and the VCard schema. The IMS LOM schema is split into several parts and is based on the Dublin Core schema. The key classes in LOM are Learning Resource Type and Entity which has a similar role to Entity in ARTstor i.e. represents a person or an organization.

In order to use the LOM schema, it was necessary to make a few changes: the original schema lists several instances of the `LearningResourceType` class called `Exercise`, `Simulation`, `Questionnaire`, `Diagram`, `Figure`, `Graph`, `Index`, `Slide`, `Table`, `NarrativeText`, `Exam`, `Experiment`, `ProblemStatement` and `SelfAssessment`. There are a number of problems with this approach: `LearningResourceType` is not a particularly appropriate name for a class, as a type is normally a property, for example `rdf:type`. Although the name chosen was not ideal, it was decided to retain it for compatibility. More substantively, it is better to use subclass relations in a schema to indicate relations between classes rather than instance relations, whether using RDFS or OWL[19]. Therefore, the schema was altered so that there is a single class called `LearningResourceType` to represent a learning resource and then various subclasses of learning resource.

Schema designers often make the mistake of omitting `rdfs:label` from schemas as they assume this information can be extracted from the fragment of the URI. This is not a good way of solving the problem as we should not expect URIs to contain metadata[20] so it is far better to add human readable information explicitly using `rdfs:label`. Therefore the addition of an `rdfs:label` property to several property or class definitions where it had been omitted was also required.

It was also necessary to create some new subclasses of `LearningResourceType` to deal with learning objects present in OCW. These classes were `Laboratory`, `Bibliography`, `Calendar`, `LectureNotes`, `Syllabus` and `ProblemSet`. One more class was found in OCW, `SelectResource`, but it was thought that this is more likely to be an error in the data, perhaps a default value for a drop down box.

3.2 Transformation of the data

After the RDF Schemas for ARTstor and OCW had been created and selected, converting the XML data representations to RDF/XML was fairly straightforward. XSLT 2 was chosen for the ARTstor and OCW data transformation, because it has a number of features for string manipulation that are not available in earlier versions that make it suitable for this task. Specifically the transform developed uses

`substring-after`, `starts-with`, `matches` and `replace` functions and defines some custom functions. A recent version of Saxon[21], version 7.8, is used for this task as it implements XSLT 2.

Converting XML to RDF/XML in this way is a syntax to syntax translation, as it is necessary to consider both the syntax and the semantics of the data when writing the transform. One proposed advantage of RDF is that if data is in RDF form it is no longer necessary to worry about syntax, simplifying such translations. OWL is a candidate solution for these types of translations.

In practice though if RDF models use different approaches, for example if they do not represent the same entities as classes, i.e. in one model a creator is represented as a string literal whereas in another model the creator is represented as a class, mapping between those models is difficult in OWL. A rules language or a transformation language could be used to transform the first representation into the second, allowing OWL to be used to map between the two representations. Although a rules language is available in Jena, or in a standalone form such a Cwm[22], there is no standardised language for this purpose. XSLT is a transformation language rather than a rules language, it has played an important role in widening the adoption of XML, so it is anticipated that a standardised rule language for RDF could have a similar role.

In the following subsections we describe a number of issues that arose while creating these transformations: specifically how to create and encode URIs, how to normalize names, how to deal with hierarchical subject terms, dealing with mixed mode data in the ARTstor data set and data cleaning.

3.2.1 Creating URIs

One important design principle in using RDF for data modelling is that all properties and some subjects and objects are often given unique URIs so they can be referred to in an unambiguous fashion. URIs were created in the datasets by giving each dataset a base URI, then extending that URI with the type of object being described and the human readable label assigned to the object in order to ensure uniqueness for each URI. It was necessary to use URI encoding on the human readable labels, as they do occasionally contain characters that are not valid in URIs. Unfortunately XSLT 2 does not have a URI encoding function, so instead *replace* was used to replace certain illegal characters with underscores. The need for URL encoding in XSLT has been recognised before as it is implemented in EXSLT[23], a set of extensions created for XSLT.

Here are some examples:
"`Museo del Prado`"
becomes
`http://simile.mit.edu/metadata/artstor/site#museo_del_prado`

"`Goya, Francisco, 1746-1828`"
becomes
`http://simile.mit.edu/metadata/artstor/Creator#goya,_francisco,1746-1828`

Creating URIs in this way has the side effect that URIs are repeatable so that multiple references to the same object in a single dataset will be merged when the RDF/XML is deserialized. Note that here we have made a pragmatic assumption that the same

string in a single dataset refers to the same thing but we do not make this assumption across datasets.

3.2.2 Name normalization

Names are encoded in different ways in ARTstor and OCW. In the library community, it is common to include bibliographic data in names such as year of birth and year of death. However this makes machine processing more difficult as it may be necessary to do free text parsing to identify the different pieces of metadata and use them effectively. In order to highlight this issue, here are some examples of names from the original datasets:

```
Van Dyck, Anthony,Sir,1599-1641
Hamilton, Sir William,1751-1801
Herschel, John F. W.(John Frederick William),Sir,1792- 1871
Close, Chuck,1940-
Paschke, Ed
```

There are also occurrences of the same name in different forms e.g.

```
Miyagawa, Shigeru
Shigeru Miyagawa
Dower, John W.
John Dower
```

It would be much better when using structured formats such as XML or RDF to separate names from biographical information at data entry time, as well as separating family and given names to avoid accidental ambiguities. Unfortunately additional complexities arise when dealing with international names as they may not follow Western conventions such as family and given. In this case reauthoring the name data was simply too time consuming, so an XSLT function was created that attempts to separate the name up into unambiguous parts.

Here is some pseudo-code for the algorithm used for name normalization:

```
IF the name contains a comma THEN
  REM it must be of the form surname forename
  IF the name is surname, forename, year of birth, year of death THEN
    extract surname, forename, year of birth, year of death
  ELSE
    extract the surname and the forename
  END IF
ELSE
  extract forename and surname
END IF
IF the forename begins with a Sir or Prof THEN
  that is a prefix
END IF
```

Note this algorithm cannot cope with all the variations found in the data, but was designed to fail in a graceful way because it always preserves the original compound name. Also it cannot deal with some name variations so they have to be fixed manually.

3.2.3 Hierarchical terms

Some of the subject terms used in ARTstor are hierarchical e.g.

```
Prints: Reference
Architecture: Modern
Architecture: Site
Religion: Buddhism: History
Nebraska: United States
Maya: Dzibilchaltun
Japan: Jomon Culture
Aachen (Germany)--Town Hall
Adam(Biblical figure)--Expulsion from Paradise
Almourol (Portugal)—Castle
```

Where hierarchical subject terms were detected in Subject, Geographic or Topic they were separated at the double hyphen or the colon and then encoded using SKOS e.g.

```
@prefix vra:     <http://web.mit.edu/simile/www/2003/10/vraCore3#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos:    <http://www.w3.org/2004/02/skos/core#> .
@prefix topic:   <http://web.mit.edu/simile/www/metadata/artstor/Topic#> .
@prefix :        <http://web.mit.edu/simile/www/2003/10/artstor#> .

<http://web.mit.edu/simile/www/metadata/artstor/Id#UCSD_41822001194057>
      a       vra:Image ;
      :topic  topic:prints_reference ;
      vra:title "Intaglio: Aquatint Plate fr. Los Proverbios by Goya" .

topic:prints
      a       skos:Concept ;
      skos:inScheme :topic ;
      skos:prefLabel "Prints" .

topic:prints_reference
      a       skos:Concept ;
      skos:broader topic:prints ;
      skos:inScheme :topic ;
      skos:prefLabel "Prints: Reference" .
```

Here we have assumed that the first term is a broader term than the second term e.g. “Architecture: Site” is a refinement of “Architecture” and “Religion: Buddhism” is a refinement of “Religion” etc. Further hierarchical relations could be extracted from the free text by parsing the brackets in the last three examples but this is not done at the moment.

3.2.4 Mixed mode data

The ARTstor data set uses mixed mode data, which is when an XML element contains a mixture of text and other XML elements. Mixed mode data is often used in XML used to store documents rather than data, but it is sometimes found in XML data as well. Recent discussions[24] on how to make XML more RDF friendly have suggested it is a good idea to avoid its use. Here are some typical examples from the ARTstor data:

```
<Subject>couples portraits
  <Geographic>Mexico</Geographic>
  <Topic>Painting</Topic>
</Subject>

<Subject>Birds
  <Geographic>Mexico</Geographic>
  <Topic>Painting</Topic>
```

```
</Subject>
```

In the XSLT stylesheet mixed mode data is dealt with by checking if an element contains a text node before processing any child elements.

Note there is a potential ambiguity in the use of the term `Subject`, as it can either refer to the subject of the record e.g. a painting or the subject of the subject of the record e.g the subject of the painting itself. As the ARTstor collection was created from six different sources, there is some variation in the meaning of subject, presumably depending on the source of the record.

3.2.5 Data cleanup

It was also necessary to do some data clean up, to correct variations in capitalization and missing or extra spaces in the ARTstor dataset:

```
Architecture: site
Architecture:Site
Architecture:site
```

is converted to

```
Architecture: Site
```

The OCW data set contained errors such as the use of different forms of the same name as illustrated in section 3.2.2. In some cases it was not possible to use the name normalization algorithm to correct these errors, so it was necessary to correct these errors manually.

3.3 Identifying record linkage and enriching the data

The previous section explained how the two main datasets from ARTstor and OCW were converted into RDF. In order to produce the data for the demo, the next step was to try to link the ARTstor and OCW datasets. This involved linking the ontologies used in the datasets as well as identifying and linking all instances of the same entity e.g. the same work, the same creator, the same site, the same subject term etc. Then the data was enriched by querying the experimental OCLC Library of Congress Authorities Web Service and the Wikipedia encyclopaedia.

3.3.1 Mapping properties

The ARTstor and OCW ontologies were linked using `owl:equivalentProperty` to link `vra:creator` to `dc:contributor` and to link `vra:subject` to `dc:subject`. An inference engine, in this case the one provided by Jena[25], was then used to generate the additional triples inferred from these two relations and add them to the underlying model. For example, if there is a triple which has a property `vra:subject`, then the inference engine will add an additional triple with the same subject and object but with `dc:subject` as the property. In each of these pairs of properties, the two linked properties use different controlled vocabularies i.e. if the same creator is used in both datasets, it will be assigned a different URI.

Mapping the `dc:contributor` relation in OCW to the ARTStor `dc:creator` relation was not straightforward. In the OCW dataset, contributors are people who contributed to the learning object (LOM). In some cases whether the contributor is the author of

the LOM is made explicit, but many LOMs in OCW do not make a distinction between the author of the LOM and the creator of the work of art as shown in the example taken from OCW below:

```
<ResourceMetadata
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <General>
    <Title>Untitled (Multiform) 1946</Title>
    <SectionTitle>Lecture "3-1"</SectionTitle>
    <Description>Through Surrealism and the experience of Still's
paintings, Rothko achieved his "signature style." Some scholars
believe a covert figuration remains in these landscape-like
abstractions.
    </Description>
  </General>
  <LifeCycle>
    <Version>Fall 2002</Version>
    <Contribute>
      <Entity>Prof. Caroline Jones</Entity>
    </Contribute>
    <Contribute>
      <Entity>Rothko, Mark,1903-1970</Entity>
    </Contribute>
  </LifeCycle>
</ResourceMetadata>
```

This is because the domain of interest of this dataset is learning objects, so no distinction is made between the different roles of the individuals involved in the LOM, and it was envisaged that users would search via course creator rather than via the creator of an individual artwork. In the ARTStor dataset, a creator is typically the creator of a work of art, rather than the person or entity who rendered the painting into a digital image. Therefore `dc:contributor` and `vra:creator` do not have exactly the same meaning so perhaps an `rdfs:subPropertyOf` relation would be more appropriate than `owl:equivalentProperty`. We suspect this may be a common problem when integrating datasets i.e. even though datasets have used standard schemas such as Dublin Core, their interpretation of those schemas varies dependent on their domain of interest.

For the prototype, although the OCW data did not always draw a distinction between author and creator, this omission was not a big problem as it was found that simply mapping properties from one dataset to the other is not sufficient to unify the datasets, as they may talk about the same entity but represent it using different URIs. Therefore, it is necessary to identify when entities are equivalent and link them using the `owl:sameAs` property. Identifying equivalent entities is time consuming for users if performed by hand so it was decided to automate the process by analysing the `rdfs:label` property of the entities and matching equivalent labels using string matching methods.

The literature describes many string matching methods which have different strengths and weaknesses – for a good overview see [26]. Open source implementations of a number of these methods are available in the SecondString framework [27]. For initial experimentation, the Levenshtein distance was chosen because it was used to create the ARTstor dataset, which itself was derived from six different datasets [28]. As it was

sufficient for this purpose, it was hoped it would also be useful for the SIMILE use case.

The Levenshtein distance is a simple distance measure but it is not ideal for our purposes: it identifies good matches, such as "murders" and "murder" but also poor matches e.g. "mountains" and "fountains". As this is partly a stemming[29] problem, future work could explore the application of stemming algorithms in conjunction with string matching. All string matching approaches have to make assumptions about the process underlying the string variation. In the case of the Levenshtein distance it is assumed to be random variation due to typing errors or due to a noisy communication medium. There are some instances of errors like this in the dataset, but most of the variation we are interested in is due to plural and singular variations, and the variations in names described previously. It was therefore necessary to review results by hand, removing false positives that occur. It may be possible to improve matching performance by exploring the use of other similarity methods but no method is sufficiently accurate that human review will not be required.

Again, an inference engine was required to process the RDF statements involving an `owl:sameAs`. This either involves adding additional triples or simulating the existence of those triples when querying the store. Unfortunately, it was found that for the demonstration data set (170,244 triples) that using the standard Jena OWL inference engine required a very large amount of memory. Therefore, it was necessary to write a custom inference engine that added additional triples inferred from statements that used `owl:sameAs` statements. It is anticipated that later prototypes will take advantage of a more memory efficient approach that is forthcoming in Jena.

3.3.3 OCLC Authority Service

The OCLC LAF (Linked Authority File) service is a prototype web service interface to data from the Library of Congress Authorities File. It supports both REST[30] and SOAP[31] using WSDL[32] to provide a web interface to retrieve LOC authority records. This experimental web service allows users to look up canonical name records by full and by partial name matching.

In order to create the SIMILE demo dataset, the service was queried with the name of all the entities that are identified as people in the combined ARTstor / OCW dataset. In order to perform disambiguation when there are multiple matches, year of birth and death information was included in the query when available. When the service is queried, it returns an XML document that links to one or more XML documents, each of which is an authority record that matches the query. The OCW dataset does not contain year of birth and death information, so this meant it was often not possible to identify the correct individual using the OCLC so instead it performed word matching which produces a large number of records of moderately similar names. Because of this profusion of data we were faced with a choice: either we needed a richer construction of equivalence than that permitted by the `owl:sameas`, or we would have to go through the result sets and by hand verify each of the references returned for validity. Initially we created a prototype implementation written in Perl that used a custom ontology that distinguished between single and multiple matches calling the first `probableMatches` and the latter `possibleMatches`. Investigating how these relations could be used and represented in the browsing interface remains a topic for future research.

For the interim, when multiple matches were returned, they were human reviewed in order to determine the correct match. Human review was also necessary for single matches, but in this case at least there is a chance that they will be correct without reviewer intervention. In order to simplify this process, the Perl implementation was replaced with a Java implementation that passes information to the human reviewer via comments in the N3 rather than as an ontology. This approach was taken as it minimizes the amount of hand editing the user needs to do compared to the other approach i.e. converting `probableMatches` or `possibleMatches` properties to `owl:sameAs` properties.

When custom tools are available for reviewing and correcting the output of automated processes, it may be preferable to adopt the ontology approach as the tools can change property names automatically. To illustrate why human reviewers are required, here is some sample output for a person mentioned in the OpenCourseWare data set, “Graham, John, 1881-1961”. As the OCLC authority service cannot find an exact match for this entity with corresponding years of birth and death, it returns 11 partial but incorrect matches:

```
http://simile.mit.edu/metadata/ocw/Contributor#graham,_john,1881-1961>
  ## John Graham
  ## oclc matches
  ## match 1
  oclc:establishedForm "Graham, John" ;
  owl:sameAs <http://errol.oclc.org/laf/n80-9522> ;
  oclc:citation "[Co-author of Pardon me, Prime Minister]" ;
  ## match 2
  oclc:establishedForm "Maxtone-Graham, John" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-7023> ;
  oclc:citation "His The only way to cross, 1972." ;
  ## match 3
  oclc:establishedForm "McDonald, John Graham,--1922-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-7751> ;
  oclc:citation "His Cases and materials on income tax, 1957." ;
  ## match 4
  oclc:establishedForm "Graham, John,--1926-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-34875> ;
  oclc:citation "His A crowd of cows, 1968." ;
  ## match 5
  oclc:establishedForm "Graham, John Alexander,--1941-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-34876> ;
  oclc:citation "His Arthur, 1969." ;
  ## match 6
  oclc:establishedForm "Graham, Peter John,--1939-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-34881> ;
  oclc:citation "His A dictionary of the cinema, 1964." ;
  ## match 7
  oclc:establishedForm "Alexander, J. J. G.--(Jonathan James Graham)" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-35146> ;
  oclc:citation "Oxford. University. Bodleian Library. Illuminated
manuscripts in the Bodleian Library, 1966-" ;
  ## match 8
  oclc:establishedForm "Brooks, John Graham,--1846-1938." ;
  owl:sameAs <http://errol.oclc.org/laf/n50-41283> ;
  oclc:citation "His The social unrest, 1903." ;
  ## match 9
  oclc:establishedForm "Lord, Graham,--1943-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-42293> ;
  oclc:citation "His Marshmallow pie, 1970." ;
  ## match 10
  oclc:establishedForm "Anderson, J. G. C.--(John Graham Comrie)" ;
```

```
  owl:sameAs <http://errol.oclc.org/laf/n50-51890> ;
  oclc:citation "His Scottish sands and gravels, 1942?" ;
  ## match 11
  oclc:establishedForm "Smith, John Graham,--1936-" ;
  owl:sameAs <http://errol.oclc.org/laf/n50-53333> ;
  oclc:citation "His Ghana&apos;s economic objectives ... 1968." .
```

When multiple matches are returned, it is necessary for a human reviewer to edit those matches and either delete all but one match, or delete all matches in this case where there are no correct matches.

3.3.4 Wikipedia data

The Wikipedia encyclopaedia is another web resource that was queried as part of building the prototype for all the entities representing people in the dataset as it contains biographical information on some of these individuals. Unlike the OCLC service, it is designed to be a human readable web resource rather than a machine readable one. This means it was necessary to try to identify the portion of text that contains the biographical entry on the page and convert it from HTML to ASCII text. This was not always successful due to different formatting standards in use, but as this information was checked by a human reviewer who can remove extraneous HTML, the results are acceptable, although some unnecessary work was performed.

When biographical information is successfully obtained, it can be quite useful for the purpose of reviewing the information obtained from OCLC, as sometimes year of birth and death information is available from the Wikipedia entry even if it was omitted from the original datasets. This allowed the human reviewer to disambiguate the OCLC results. Just as with the OCLC data, human review is essential in order to remove erroneous matches that occur. For an example of the kind of problems that occur, here are the OCLC and Wikipedia results for the artist “Graham, John, 1881-1961”.

```
http://simile.mit.edu/metadata/ocw/Contributor#graham,_john,1881-1961>
  wikipedia:description ""Jack Gilbert Graham(Redirected from John Graham)
John "Jack" Gilbert Graham (January 23, 1932 - January 11, 1957) was a mass
murderer who killed 44 people by planting a dynamite bomb on board United
Airlines flight 629. Flight 629 was using a Douglas DC-6-B and was taking
off from Denver, Colorado's Stapleton Airport, bound for Portland, Oregon on
November 1, 1955. There were no survivors of the subsequent explosion.
Jack's mother was on board the plane - the motive for the bombing was to
claim $37,500 worth of life insurance money, from policies bought in the
airport just before take-off. Flight insurance was bought in vending
machines at airports back in the 1950's. Jack Gilbert Graham was executed by
gassing in the Colorado State Penitentiary gas chamber on January 11, 1957.
The Jack Gilbert Graham case may have been the inspiration for the 1970
movie Airport.
  """ .
```

Clearly this is a different individual from the one referred to in the OCW dataset as the year of birth and death differ – for the correct biography see [33]. Unfortunately there is no way of using year of birth and death information in a Wikipedia query. This should emphasise the importance of human review as we are using datasets with limited information on individuals, which have incomplete information, or are not intended to be machine readable.

One interesting side effect of querying the OCLC and Wikipedia services was that it created links between the ARTstor and OCW data by identifying the same individual

in both datasets. Effectively this is like using a string similarity measure with a maximal allowable edit distance of zero.

3.4 User Interface

A faceted browser was written using the Jena RDF Framework to allow the user to browse the resulting dataset. Faceted browsing is an interaction style where users filter a set of items by progressively selecting from only valid values from a set of different classification dimensions[34] [35]. In the prototype, the browser represents these dimensions using SKOS. The choice of faceted browser was deliberate because it demonstrates the added value of annotating a collection with structured metadata. If the prototype had only implemented a search interface, similar to Google's image search, then this could have been implemented using free text descriptors so the advantages due to the use of structured metadata would not have been clear. For some screenshots of the faceted browser see Figures 1 and 2.

The Haystack[36] universal information client developed at MIT uses several strategies to assist users when browsing through semistructured data. One of these strategies, collection refinement, involves grouping terms in the collections member elements in a similar way to faceted browsing[37], but as multiple strategies are used the Haystack functionality goes beyond simple facet browsing. It is anticipated that future SIMILE work may investigate the use of Haystack or browsing approaches taken in Haystack over SIMILE datasets.

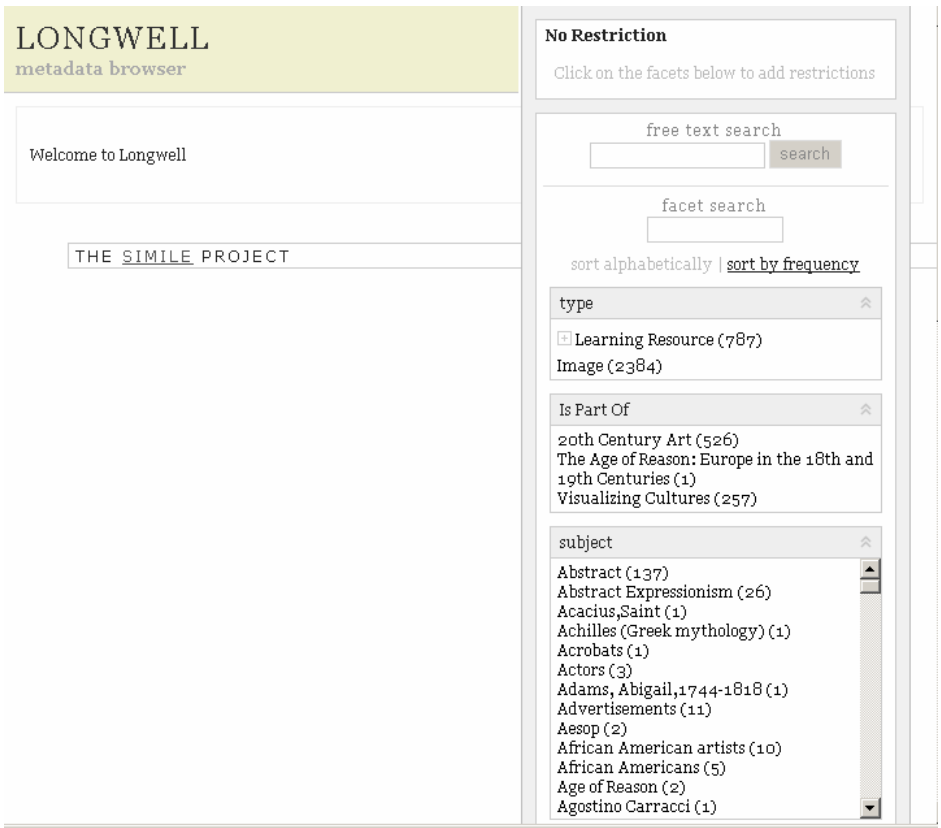

Figure 1 - First screen presented to user in SIMILE prototype

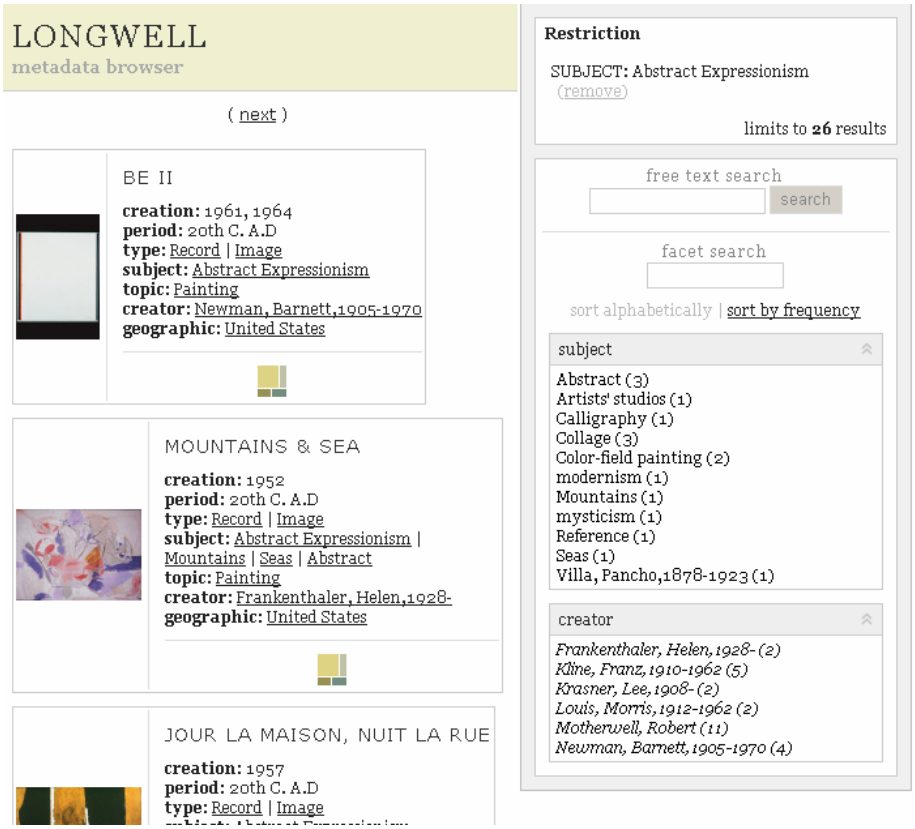

Figure 2 - Browser output after selecting a facet

4 Conclusions

In conclusion, this work has demonstrated that it is possible to convert XML datasets to RDF/XML using XSLT. This is a syntax to syntax translation, so the author of the transform has to consider both the syntax and the semantics of the data. Rather than seeing RDF and XML as opposing technologies, we think this highlights that XML technologies can play an important role in enabling the Semantic Web due to the large amount of legacy XML data available, so the ability to convert XML data to RDF, even though the conversion is syntactic, is nevertheless valuable.

Furthermore XSLT 2 is much better for this purpose than previous versions as it provides better pattern matching and substitution support. It would be very useful if future revisions of XSLT supported URI encoding rather than requiring developers to create ad-hoc URI encoding algorithms using the XPath `replace` function.

When converting existing datasets to RDF, one key issue is determining which string literals should be replaced by class instances. In the case of the ARTstor data, the literals representing people, sites, sources, collections and subject, topic and geographic concepts were replaced with class instances. Correctly identifying which literals to replace is essential to allow the use of OWL in order to perform record linkage between datasets or to use SKOS aware faceted browsers to navigate data sets using classification schemes.

This work also shows that just as for other approaches to data modelling, it is possible to automate identifying record linkage of RDF datasets using string similarity techniques. However this approach is error prone and the results must be checked by

human reviewers. One particular problem here is the amount of information available, because if the only information available to determine wether a match is correct is the name of a creator it is very hard to ensure that the match is correct with any certainty.

The work also demonstrates that it is possible to query machine and human readable web sources in order to enrich datasets with authority or biographical information, although full consideration must be given to copyright and ownership issues. Again it is necessary for human reviewers to check the validity of data created in this way.

Appendix A

Prefixes

Prefix	Name	URI
art	ARTstor	http://web.mit.edu/simile/www/2003/10/artstor#
vra	VRAcore	http://web.mit.edu/simile/www/2003/10/vraCore3#
person	Person	http://web.mit.edu/simile/www/2003/10/person#
vcard	VCard	http://www.w3.org/2001/vcard-rdf/3.0#
skos	SKOS	http://www.w3.org/2004/02/skos/core#
dc	Dublin Core	http://purl.org/dc/elements/1.1/
dcterms	Dublin Core	http://purl.org/dc/terms/
lom	LOM	http://www.imsproject.org/rdf/imsmd_rootv1p2#
lom-life	LOM	http://www.imsproject.org/rdf/imsmd_lifecyclev1p2#
lom-gen	LOM	http://www.imsproject.org/rdf/imsmd_generalv1p2#

ARTstor schema

Class	Property	Property description
vra:Image	art:metadataCreationDate	The date the metadata was created
	art:metadataUpdateDate	The date the metadata was updated
	art:imageId	An identifier for the image
	art:objectId	An identifier for this object
	art:inCollection	The collection the image was sourced from
	vra:type	Identifies the record as being either a Work record or an image record.
	vra:creator	The creator of the image
	vra:id	The ID of this record
	vra:idCurrentAccession	The accession number of this record
	vra:measurementsFormat	Information about the format of the image, for example whether it is color
	vra:title	The title of the image
	vra:period	The time period during which the image was created e.g. 17th C. A.D.
	vra:topic	The topic of the image e.g. a painting
	vra:geographic	The geography associated with the image e.g. Flanders (Belgium)
	art:thumbnail	A pointer to an image thumbnail.

	vra:typeAAT	Subject terms encoded using the Getty Art and Architecture thesaurus.
	vra:locationCurrentRepository	The current repository
	vra:locationCreationSite	The creation site
vra:Entity person:Person	rdfs:label	The name of the person
	vc:prefix	Any prefixes associated with a name
	vc:Family	Family name (surname)
	vc:Given	Given name (forename)
	vc:FN	Full name
	person:birth	Year of birth
	person:death	Year of death
art:Site	rdfs:label	The site name
art:Source	rdfs:label	The source name
art:Collection	rdfs:label	The collection name
skos:Concept	skos:prefLabel	A preferred label
	skos:inScheme	The scheme that the concept is part of.

OCW Schema

Class	Property	Property Description
lom:learningResourceType	dc:title	The title of the resource
	dc:description	A text description of a resource
	lom-life:version	The version number of a resource
	lom-life:instructionalDesigner	The instructional designer
	dc:contributor	Someone who contributed to the resource e.g. the painter of the painting which the resource is discussing
	dcterms:isPartOf	A resource that this resource is part of, e.g. a course
	lom-gen:aggregationlevel	The LOM aggregation level
lom:Entity	rdfs:label	A human readable label for the entity
	vc:prefix	Any prefixes associated with the name
	vc:FN	Full name
	vc:Given	Given name (forename)
	vc:Family	Family name (surname)

[1] SIMILE, Semantic Interoperability of Metadata and Information in unLike Environments
http://simile.mit.edu/

[2] DSpace™, MIT Libraries and Hewlett Packard,
http://www.dspace.org

[3] ARTstor,
http://www.artstor.org/

[4] Union Catalogue of Art Images Project, University of California, San Diego
http://gort.ucsd.edu/ucai/

[5] MIT OpenCourseware
http://ocw.mit.edu/

[6] What are "learning objects"?, University of Wisconsin, Milwaukee
http://www.uwm.edu/Dept/CIE/AOP/LO_what.html

[7] Wikipedia Open Encyclopedia
http://en.wikipedia.org/wiki/

[8] Prototype OCLC Library of Congress Name Authority Service
http://alcme.oclc.org/eprintsUK/index.html

[9] RDF, World Wide Web Consortium
http://www.w3.org/RDF/

[10] XSLT 2, World Wide Web Consortium, Editor: Michael Kay,
http://www.w3.org/TR/xslt20/

[11] The Levenshtein distance, Michael Gilleland,
http://www.merriampark.com/ld.htm

[12] Screen scraping, Chris Ball,
http://www.perl.com/pub/a/2003/01/22/mechanize.html

[13] Getting into the Semantic Web through N3, Tim Berners-Lee,
http://www.w3.org/2000/10/swap/Primer

[14] Functional Requirements of Bibliographic Records, IFLA Section on Cataloguing, 1998
http://www.ifla.org/VII/s13/frbr/frbr.pdf

[15] VRAcore, Visual Resources Association Data Standards Committee,
http://www.vraweb.org/vracore3.htm

[16] SKOS, Alistair J. Miles, Nikki Rogers, Dave Beckett,
http://www.w3.org/2001/sw/Europe/reports/thes/1.0/guide/

[17] Application profiles: mixing and matching metadata schemas, Rachel Heery and Manjula Patel
http://www.ariadne.ac.uk/issue25/app-profiles/

[18] IMS RDF LOM Binding, IMS Global Learning Consortium,
http://www.imsproject.org/rdf/

[19] Subclasses versus Instances, OWL Web Ontology Language Guide, Editors: Mark K. Smith, Chris Welty, Deborah L. McGuinness, W3C Recommendation 10th February 2004,
http://www.w3.org/TR/2004/REC-owl-guide-20040210/#DesignForUse

[20] Architecture of the World Wide Web, First Edition, W3C Working Draft 9 December 2003, Editor: Ian Jacobs, W3C
http://www.w3.org/TR/webarch/#uri-opacity

[21] Saxon XSLT Processor, Michael Kay,
http://saxon.sourceforge.net/

[22] Closed World Machine (CWM), Tim Berners-Lee, W3C
http://www.w3.org/2000/10/swap/doc/cwm.html

[23] EXSLT encode-uri function
http://www.exslt.org/str/functions/encode-uri/index.html

[24] Make your XML RDF friendly, Bob DuCharme, Jack Cowan
http://www.xml.com/pub/a/2002/10/30/rdf-friendly.html

[25] Jena, A Semantic Web Framework for Java, HP Labs,
http://jena.sourceforge.net/

[26] A Guided Tour to Approximate String Matching (1999), Gonzalo Navarro, ACM Computing Surveys
http://citeseer.ist.psu.edu/409171.html

[27] Secondstring Sourceforge project,
http://secondstring.sourceforge.net/

[28] UCAI: Challenges and Choices, Esme Cowles,
http://www.diglib.org/forums/fall2003/cowles/cowles-dlf03.html

[29] Stemming Algorithms A Case Study for Detailed Evaluation, David A. Hull, 1996 Journal of the American Society of Information Science
http://citeseer.ist.psu.edu/hull96stemming.html

[30] Building Web Services the REST Way, Roger L. Costello
http://www.xfront.com/REST-Web-Services.html

[31] A Gentle Introduction to SOAP, Sam Ruby,
http://www.intertwingly.net/stories/2002/03/16/aGentleIntroductionToSoap.html

[32] Using WSDL in SOAP applications , Uche Ogbuji,
http://www-106.ibm.com/developerworks/webservices/library/ws-soap/?dwzone=ws

[33] Visual Thinking: Sketchbooks from the Archives of American Art
http://archivesofamericanart.si.edu/exhibits/sketchbk/jgraham.htm

[34] Faceted Browsing Blog, Keith Instone
http://user-experience.org/uefiles/facetedbrowse/

[35] Flamenco Search Interface Project
http://bailando.sims.berkeley.edu/flamenco.html

[36] Haystack, MIT Computer Science and Artificial Intelligence Lab
http://haystack.lcs.mit.edu

[37] Assisted Browsing for Semistructured Data, Vineet Sinha, David Karger, and David F. Huynh, Poster in the Twelfth International World Wide Web Conference, May 2003
http://www.ai.mit.edu/people/vineet/1
http://www.ai.mit.edu/people/vineet/www2003-nav-extabs.pdf