Seme4

LINKED DATA SOLUTIONS

ResearchSpace: The Process of Creating Linked Data

prepared for

The British Museum

December 2010

Seme4 Ltd.

18 Soho Square, London, W1D 3QL
Tel: +44 (0)20 7060 1590
www.seme4.com

Table of Contents

1 Introduction

This document outlines the process recommended for converting data from a Collections Management System in the Cultural Heritage domain into Open Linked Data, compatible with the CIDOC Conceptual Reference Model (CRM) [].

Although there are brief introductions to appropriate topics, it is not intended for this document to be a comprehensive technical introduction to either Linked Data or the CIDOC-CRM, as there are many tutorials available [,]. It is presumed that the reader has detailed knowledge of the Cultural Heritage domain and Collections Management Systems in general, and has also familiarised themselves with the basics of Linked Data, RDF, and the CIDOC-CRM. In addition it is assumed that persons undertaking the transformation of a particular Collections Management System have an expert understanding of the both the collection itself and the model currently in use within that system.

We describe the main steps required to convert data from a Collections Management System into Linked Data, highlighting key considerations and decisions which must be made. This process has been used to demonstrate its feasibility for the ResearchSpace Initiative in converting the entire Collections data from the British Museum into Linked Data.

The feasibility study has produced Linked Data available at http://bm2.rkbexplorer.com/; however note that Seme4 is not contractually obliged to host this data. As the information has not yet been released into the public domain authentication is required, and details are available on request.

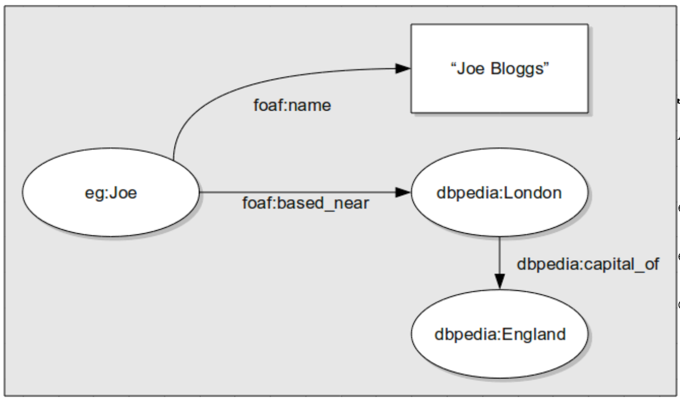
2 Linked Data

Linked Data, commonly Open Linked Data or Linked Open Data, comprises both a set of technologies and a publishing paradigm which actively encourages free and easy access to structured information on the World Wide Web [].

In essence Linked Data uses a machine readable data format called RDF [] to permit the representation of facts about virtually any kind of resource(s) to be described, including both real-world entities such as people, objects, places and materials, as well as more abstract notions such as manufacturing techniques, design styles, ethnicity and time periods.

Each and every resource, object or concept described using Linked Data is identified by a URI, (ie a web address), and crucially a description of that resource, object or concept is made available at that address, often both in machine and human readable forms. As a result, one can easily discover the information known about a specific resource, simply by "resolving" or visiting the address given by its identifier using either semantically aware software tools or an ordinary web browser.

RDF statements are comprised of a "*subject – predicate – object"* representation, forming directed graphs. The *subject* can be a real-world or conceptual entity, often referred to as a resource, and is identified by a URI. The *predicate* is a relationship or link, also represented by a URI, and the *object* is either another resource or a literal value. The following example represents three triples –



Joe Bloggs lives in London, the capital of England.

```
eg:Joe foaf:name "Joe Bloggs" .
eg:Joe foaf:based_near dbpedia:London .
dbpedia:London capital_of dbpedia:England .
```

The power of RDF and Linked Data comes through the ability to refer to resources by URI and the ease in which the resultant graphs can be created and combined to form powerful models. One can simply make statements about URIs, and when loaded into a semantically aware system these join up to provide a coherent graph. By utilising well defined URIs to identify concepts and resources, a wide range of data synthesis tasks can be easily achieved, including the combination of data taken from across multiple systems or organisations.

Semantic databases or "triplestores" offer a range of facilities to import RDF data and enable queries to be made over the various graph structures they contain. In a similar fashion to the ease in which RDF graphs can be created and combined, it is easy to add new predicates (relationships) or to enhance the ontological model by defining additional predicate URIs. This often leads to an analogy being drawn comparing a triplestore to a schema-less database system; the flexibility of RDF based approaches allows the structure or modelling schema of the data to be readily changed and enhanced in ways which would cause significant difficulties and require changes to be made in a traditional relational database system.

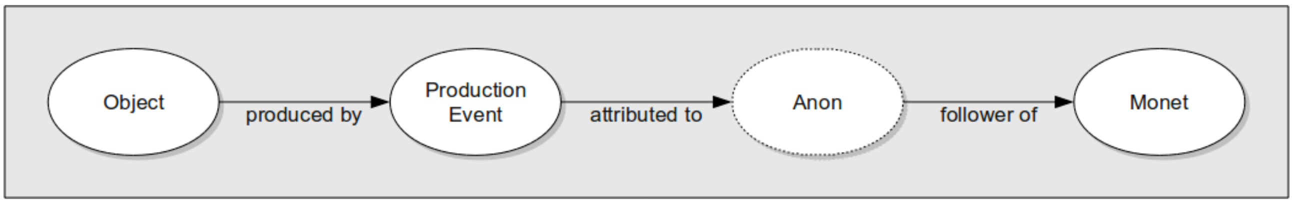
In the early years hosting Linked Data involved a number of non-trivial technical requirements, however open source software libraries and commercial services are now available which can be used in conjunction with a semantic database or RDF triplestore to provide the necessary functionalities.

3 CIDOC-CRM

The CIDOC-CRM [] provides a formal structure for describing the concepts and relationships used to represent the objects and processes inherent within the Cultural Heritage domain. The model is a result of more than 10 years development by leading members of the domain, and has resulted in ISO standardisation. In recent years, a schema has been produced [] which facilitates the description of Cultural Heritage artefacts in the RDF language in accordance with the CIDOC-CRM model. It is this semantic model, or Ontology, which forms the basis for the Linked Data representation.

The CIDOC-CRM can be seen as largely an event or activity based model, which may take some getting used to at first. However, once you become more familiar with the model, you will begin to appreciate the thoroughness and flexibility of the approach. Some relationships can at first seem unnecessarily long-winded, however "place holders" are important as additional information may come to light at a future time, and having the correct places to which this extra data can be attached is a useful asset.

For example, in attributing the style of an object to the influence of a particular sculptor or artist, one may envisage a direct relationship from the object asserting that it is "in the style of" the given individual or group. In the CRM however, the object is first associated with a production event, which is attributed to the creator of the object (irrespective of whether the creator is known, or unknown in which case an anonymous actor is described). It is then this creator which is described as being a pupil or a follower of the style, group or artist in question.



This may seem rather over-complex, however it is easy to imagine that further information is already known that fits within this structure. The place, date, or materials used in the production of the object may be known (these are attached to the production event) or details of the original master (their name, birth, death, floruit, etc). Or, for example, after further analysis at a later date the name of the previously unknown imitator may be determined, hence having the provision within the model to record this is important. Using the well-reasoned structures within CIDOC-CRM is an excellent way of future-proofing the model for later data attribution or extension.

The CIDOC-CRM provides comprehensive coverage over the basic concepts and processes which are typically found within Cultural Heritage organisations. In addition, the Simple Knowledge Organization System (SKOS) [] ontology can be used to represent thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary, facilitating the depiction of object types, materials, styles, cultures, places and so on, along with the hierarchical relationships between them. However there may be cases in which a given institution may wish to record more detailed properties of an object, or to describe relationships not explicitly defined within the CIDOC-CRM model. In such cases the flexibility of RDF permits additional custom predicates to be defined, either as extensions or specialisations of existing CRM properties, or as stand-alone relationships applicable to a specific domain only.

4 Conversion process

There are various stages required in taking data from a Collections Management System and producing RDF suitable for hosting as Linked Data. These consist of:

- making some initial decisions concerning the URI scheme to be used for the Linked Data;
- exporting information from the Collections Management System;
- defining a mapping to translate that information into a CIDOC-CRM compatible structure;
- defining any required ontology extensions;
- converting the data into RDF;
- finally importing it into a Linked Data publishing platform.

4.1 URI / domain format

Due to the URI resolution capability which lies at the heart of Linked Data, the URIs used are intrinsically linked to the hosting solution. Linked Data can only be published using URIs from a domain over which the publisher has administrative control. If you wish to publish data with URIs in the form http://www.example.com/object/1234 then you must have control of the example.com domain in order to configure suitable software and libraries to ensure that the description of object 1234 is made available at that address in the correct formats.

As outlined in the How to publish Linked Data tutorial [] there are a number of schemes in which Linked data can be published. There are two main points to consider, which are likely to be dictated by the publishing solution employed.

Firstly, there is consideration to be given as to how to identify different (potentially related) datasets published by the same organisation or hosting platform. The two key options are either to use a separate sub-domain distinguishing each dataset (1), or to include a dataset name or identifier in the URI path (2).

(1)	Sub-domain	http://{*dataset*}.domain.org/...
(2)	URI path	http://www.domain.org/{*dataset*}/...

A sub-domain solution (1) potentially requires more advanced configuration, but offers additional flexibility as individual datasets may be easily moved to a different machine or hosting provider at a later date if required. Segregating datasets based on a URI path prefix may be easier to set up initially, or in the case that only one dataset is to be hosted a prefix may not be required at all.

The second consideration is how to distinguish between the URI which is the identifier for a resource, and the description of that resource itself. This has traditionally been the most confusing aspect of Linked Data, and caused the most problems as developers hand-rolled early hosting implementations before libraries became readily available. In essence an object is identified by a *non-information resource* (NIR) which when accessed in a browser or resolved by a client returns a redirection to a separate *information resource* (IR) containing a description of the item in an appropriate format (usually either RDF for machine interpretation, or HTML for human consumption). Again, there are two main schemes for discriminating between the NIR and IR, namely a URI prefix (3) or file extension (4).

(3)	Prefix-based discriminator	http://data.example.com/{*discriminator*}/object/1234
(4)	Extension-based discriminator	http://data.example.com/object/1234{*.discriminator*}

Early linked data systems typically used a prefix-based discriminator, often using /id/object/1234 for the NIR and then redirecting to something like /data/object/1234 for RDF representation or /page/object/1234 for human readable HTML representation. Later systems more commonly use an extension-less URI for the NIR, eg /object/1234, which redirects to URI including a file extension appropriate to the format returned, eg /object/1234.rdf or /object/1234.html.

The UK Government guidelines for Linked Data [] uses a hybrid of both URI prefix and extension based discriminators, using /id/{type}/{number} for the NIR which redirects you to an IR in the form /data/{type}/{number}.{extension} largely to emphasise the difference between the identifier and the associated information resources.

However for most applications we would recommend simply using an extension-less URI for the non-information resource, and the same URI with an extension for the information resource(s), as it is felt that making the NIR to IR transition as transparent as possible is of benefit to the user; most do not even need to know or realise that they are being redirected from one to the other and appending an extension is the easiest and least obtrusive way in which this can be achieved.

Having discussed these various alternatives for URI conventions, those choices made for a particular deployment are likely to be strongly influenced by the capabilities and style of the hosting platform employed – it is not uncommon that a platform or library will only support one type of scheme, rather than having the flexibility to support multiple configurations.

As a general note however, whilst URIs are "opaque" and carry no meaning, it is useful for them to be assigned in a well structured manner which is understandable by a human user. That is to say it would be perfectly valid for Linked Data to be generated by assigning a random yet unique number or hash string to form the URI for each resource that is to be published; however it is more intuitive for a human user to navigate if a number of basic conventions are followed such as prefixing all resources of a particular type with a meaningful word or abbreviation.

The Linked Data translation of the British Museum collection created in this feasibility study is hosted on the RKB Explorer platform, and as a result all NIR URIs are in the basic format

http://bm2.rkbexplorer.com/id/{*resource_type*}/{*identifier*}

The resource types are typically object, thesauri, or person-institution. Where additional events occur related to a given resource, these are identified by appending additional path components prescribing that event to the URI of the resource, eg

http://bm2.rkbexplorer.com/id/object/PPA180466 has an associated production event with ID http://bm2.rkbexplorer.com/id/object/PPA180466/production

In a full production system, it is recommended that the URI scheme for the British Museum Collections catalogue is

NIR	http://collection.britishmuseum.org/{resource_type}/{identifier}
IR	http://collection.britishmuseum.org/{resource_type}/{identifier}.{extension}

4.2 Data Export

The first stage of the conversion process is to export data from the existing Collections Management System. In the feasibility study we have conducted, data has been exported in XML format from the British Museum's Merlin system, producing approximately 14 gigabytes of data in the following files. It should be noted that this is not an insignificant quantity of data, indeed due to limitations with the Merlin system, exporting this data has taken several weeks.

```
    5MB bibliography.xml
  212MB biography.xml
  781MB objects_A.xml
4,439MB objects_C.xml
1,008MB objects_E.xml
  626MB objects_G.xml
2,154MB objects_H.xml
2,965MB objects_P.xml
1,784MB objects_W.xml
  548MB objects_Y.xml
   51MB thesaurus.xml
```

XML has been selected as the most appropriate export format for a number of reasons. Firstly, it is a common standard data exchange format, with a wide variety of tools available for processing and manipulation. It is also capable of representing complex data structures which can be difficult to represent in flat file formats. In addition, the Merlin system used by the British Museum is interesting in that it employs an object-oriented database system, rather than a traditional relational database model, so producing tabular data such as a typical SQL dump would have been difficult. Conversely, relational databases and management tools such as phpMyAdmin can often export naive dumps of tabular data in XML format, hence these systems are not precluded. Systems which output data in CSV or similar files can quite easily be imported into a database, or via simple scripts be translated into XML.

In producing XML dumps, one should preserve as much of the original data structure as possible. The British Museum output includes quite complex representations of each object, including all the properties and features associated with it in a single record. An example is available at []. As a result, the configuration required to process these complex descriptions is similarly quite lengthy.

Conversely a system which employs a relational database model is likely to contain a fragmented structure of the overall data, for example containing tables representing each of the key concepts, and others tables providing the mappings, associations or linkages between them. Because of the graph representation of RDF and the ease of which one can simply write disjoint statements which later become coherently combined, a naive dump of each table can be processed separately in turn. As a result, a configuration for mapping an relational based system is likely to contain many small chunks corresponding to the various tables within the database. Indeed, a well structured database schema is likely to help inform and guide the conversion process.

A common pitfall encountered by those less familiar with XML is to ignore the important difference between an empty element, and one which “looks” empty but contains whitespace characters (eg spaces, tabs, or new-lines). Fields which have no genuine values should either be truly empty, eg `<foo></foo>`, or simply not included in the output. Finally all data exports should be valid UTF-8.

4.3 Conversion tools

There are a number of ways in which data can be processed into RDF. Typically RDF is serialised as an XML document, namely RDF/XML [], however RDF can also be encoded using N-Triples [] or Turtle [] formats. RDF/XML is by far the most common format, though it can be overly verbose and difficult for humans to read. N-Triples is extremely simple and consists literally of a line-by-line listing of triple statements, ideal for importing into a triplestore, but it is hard for users to understand any structure in the data in this format. Turtle is a good half-way house, providing basic structures in a terse format that can be quite easily understood by human readers.

During this project, Seme4 has converted the British Museum's catalogue data from XML to RDF/XML utilising an in-house tool, *makeRDF*. This tool is a generic component which utilises a configuration or mapping file to determine how an XML document should be converted into RDF/XML. At a very high level, the mapping configuration contains items in the XML input that the tool should look for, and tells it which type(s) of RDF element(s) to produce when those patterns are encountered.

Other approaches include XSLT [] for performing XML to RDF/XML conversion, however this requires a very specialist knowledge and is likely to produce a system which is impenetrable to those not familiar with XSLT and more importantly is unlikely to be easily re-purposed or adapted. Alternatively, a script could be written in whatever the favourite scripting language is that parses the XML and specifically outputs certain snippets of RDF/XML when certain elements are encountered, however this again does not lend itself to a solution which can be readily adapted and re-deployed for other institutions or input formats.

4.4 Producing a configuration for makeRDF

Seme4 has created a mapping configuration for makeRDF which converts the supplied XML exported from Merlin and produces RDF/XML output in accordance with the CIDOC-CRM. The complexities of the data being processed meant this was not a trivial task, even for staff with considerable experience in converting data, RDF, and ontology definition. It was necessary to acquire considerable nderstanding of the Cultural Heritage domain and initial unfamiliarity with the CIDOC-CRM was certainly a barrier. In practice, a majority of the problems encountered were concerned with understanding and interpreting the BM model, with clarification often required from the domain experts. In addition advice was available where necessary from Martin Doerr, one of the authors of the CIDOC-CRM. It is likely that other projects of this kind will meet the same challenges, although it is unlikely that there are other systems that are any more complex than the BM model.

The configuration produced can in some ways be seen as a bespoke solution for the British Museum. Certainly it is tied to the XML structure of the data exported from Merlin; other systems used by other institutions are unlikely to produce identical output. However, the BM model is based around the SPECTRUM model [], and represents a broad range of the typical concepts and relationships in the Cultural Heritage domain, so parallels are likely to be drawn between similar systems.

There are two main approaches for adapting the work carried out in this investigatory project to be suitable for use with data exported from other Collections Management Systems. One is to expend effort in manipulating whatever output is available from an alternative system such that it is in the same structural format as that of the Merlin export. If this can be achieved then the makeRDF tool can be run using the existing configuration with only the most minor modifications to change domain names and institution names. However translating XML from one arbitrary format to another requires significant programming abilities, which, depending on the similarity or difference between the input XML and desired Merlin structure, may well be on a par with creating a custom translator to convert the input XML directly to RDF. The alternative approach is to modify the configuration file to suit the input XML, and in most cases this is much more likely to be achievable by a non-expert programmer.

In future prototype work for ResearchSpace the existing configuration could be modified to suit the standard SPECTRUM serialisation, which is more likely to be similar to that which can be exported from a range of other Collections Management Systems.

Sadly there is no simple answer or magic technique in creating these mappings, largely because they involve moving from one imperfect conceptualisation to another, neither of which is necessarily an exact representation of reality. Issues debated often include personal opinions and perspectives, hence design by committee is to be avoided. These activities are best carried out by a small team of two to five people, within which there must be expert knowledge of the existing or input data model, and preferably as much experience as possible with RDF and the CIDOC-CRM.

An iterative approach is recommended, starting with a small subset containing only the core concepts and most important relationships, and then expanding to cover all of the existing domain model. A rigorous and methodical approach is essential, having a database schema diagram and/or print-out of example XML records upon which each object, table, field, relationship and element can be ticked off is of great benefit.

The vast majority of concepts and relationships required within the Cultural heritage domain are already present in the CIDOC-CRM. Looking at the graphical representations [,] of the CRM and the full index [] may assist. In addition, the CRM and extensions created for the BM are listed and browsable at http://crm.rkbexplorer.com/.

Where a property or relationship does not exist in the CRM, or the ones that do are not specific enough to sufficiently represent the original data, then additional properties should be created. Ontology definition is beyond the scope of this document, other than to say there are many guides and tutorials online. You can simply use a new URI to represent a new property, preferably on a domain name which you control such that the metadata describing that property can be properly represented and made accessible.

In converting the BM model, there are several instances where the CRM contains a generic predicate which was further specialised with sub-properties to capture more specific relationships. For example, the predicate `P11F.had_participant` can be used to associate a person or

organisation as having taken some role or participation in the production event of an object. However the BM model often has more specific details of the nature of the involvement. As a result, predicates have been defined to reflect these more specialised relationships, such as `PX.written_by`, `PX.photographed_by`, `PX.engraved_by` and `PX.painted_by`. Each of these specialisations are defined as a sub-property of `P11F`, so a triplestore with basic reasoning capabilities will infer `P11F` whenever one of these sub-properties are encountered. As a result, a tool which has no knowledge of the new extensions can still operate and access the data using the more generic property.
The full list of additional properties created for the British Museum is available at http://crm.rkbexplorer.com/bm-extensions/

Configuration mappings for makeRDF take the form of an XML document, and use XPath [] queries to identify parts of the input XML. The available directives for configuring makeRDF are described at http://crm.rkbexplorer.com/makeRDF/.

As a guide, an example of creating a configuration mapping for a simple book store is presented at http://crm.rkbexplorer.com/makeRDF/books-example/.

Throughout the task of mapping the British Museum model to CIDOC-CRM, a commentary outlining the process and the decisions made has been recorded. This is available at http://crm.rkbexplorer.com/mapping/. This document includes links to the full mapping configuration which has been successfully used to translate the BM Collections Catalogue data.

4.5 Linked Data best practices

We have outlined the requirements of Linked Data earlier in this document. To recap, the most important points are listed as follows –

- All entities, concepts and relationships are to be represented by HTTP URIs
- Descriptions of these resources will be made available at the location specified by each URI, see "How to publish Linked Data" [16]
- Identifiers must be persistent, and never re-issued
- Linked data should be published with a clear licence statement, preferably using a standard data publishing licence, such as PDDL [] or CC0 []. Licences requiring attribution are to be avoided since, when consuming RDF data, it is often not practicable or even possible to identify and credit all sources used. As a result, some systems may avoid using such data as they cannot properly comply with the attribution requirements of the licence.

4.6 Common pitfalls

There are two related areas in which novices can make serious mistakes when creating Linked Data.

RDF works under an "open world" assumption, which implies two things. Firstly, there may be more data out there, and thus your representation is not necessarily complete. Secondly, the presence of a triple implies that the specified relationship or fact holds true, the absence of a statement does not imply that it is not true, but that it is currently unknown. As a result, it is perfectly acceptable to have "missing" data in an RDF model, where information is not known. Conversely in database systems, a special value is often used to represent "unknown". When converting such data to RDF, it is important that any concepts representing unknowns are removed. If, for example, in a collection of ancient manuscripts a database records the author of many of the artefacts as unknown, it is important that the resulting RDF does not contain an author named Unknown. If it were to do so, then a large proportion of the manuscripts would be incorrectly associated together as having the same author, namely Mr Unknown.

The second common pitfall is closely related to this notion of false or unintended association, but occurs through the naive construction of URIs for resources which do not already have unique identifiers. For example, let us consider again the way in which people are represented in a Collections Management System, such as the author of a manuscript. The system may model people as first class entities, internally storing them each with their own identifier, or alternatively in a description of an object there may simply just be a field for "author". In this latter case, an identifier must be created such that the author can be properly described in Linked Data form. Typically values from the object to be identified are used as the basis for generating an identifier, such as computing the md5 hash of the name of the author. However, there is a danger here in that the author names are almost certainly not unique, Mr Smith who authored one manuscript is not necessarily the same Mr Smith who has authored another, and they may even have lived centuries apart. If the proposed scheme of hashing the name was imposed, then anyone named Mr Smith would be unwittingly collapsed into a single entity. In general, it is usually not sufficient to generate URIs based on a single field: rather in this case it would be sufficient to combine the author's name and the artifact ID before hashing to generate a unique identifier (again, unknowns are to be avoided).

Finally, all RDF data produced should be checked for validity, either using the online service from the W3C [] or local utilities such as the *rapper* tool provided as part of the Raptor RDF library [].

4.7 RDF conversion and makeRDF best practices

A few general notes on creating a mapping configuration and converting RDF, in addition to the Linked Data best practices.

- Try to keep the configuration as simple as possible. If you're creating horribly complex XPath expressions, deeply nested configurations, or lots of conditionals, think again to see if there is an easier approach. Perhaps several smaller more targeted configurations may be possible?
- Take special care in the definition of `<identifier>` URIs with relation to the pitfalls above.
- Try not to repeatedly define resources. For example if your museum has three different galleries, as you process each object, you don't want to define a new resource representing the complete location for each object, as part of this would be repeating the same description for each gallery many times. Instead, wherever possible, extract such features and process separately. You can then just create a single triple for each object associating it with the gallery resource, rather than creating the gallery resource in-line.
- Don't take short-cuts with the CRM model, always ensure you do not preclude future data

addition or ontology enhancements.

- Avoid `<bnodes>` with the exception of dimensions and time intervals, i.e. resources which you would not ordinarily give identifiers for in their own right.

4.8 Updates

The quantities of data involved with this feasibility study are considerable by most standards, and are only anticipated to increase as additional institutions become involved with future activities. Exporting XML data, translating to RDF and asserting into a triplestore are all time consuming tasks. It has taken several weeks to export from Merlin, approximately 12 hours to decompress and convert, and between 12 and 24 hours to assert depending on the triplestore employed, so clearly repeating the entire process for minor updates is infeasible.

An interesting constraint which must be taken into consideration is the varying ability of different triplestores to update data. Some permit a triple-level update, akin to changing a single field value in a database. More common however is the ability to only perform updates on an entire model, or graph, reflecting the unit of data that was originally imported. This is more like updating a row, or an entire table in a traditional database. As a result, it is important to consider the size or unit of data imported.

In this feasibility work, we have taken the step of breaking down the extremely large XML input files to small manageable chunks. This is in itself quite an expensive task, however a large number of small files are easier to manipulate and process compared to a few very large ones. In addition, it is this granularity of smaller files that are imported into the triplestore. Each of these files or chunks typically contains the full XML data for up to 10 objects, with objects being allocated to chunks based on a known hashing algorithm.

The proposed update mechanism is that an XML file is generated from Merlin containing full details of each object that has been modified since the last update, in addition to a separate list of any objects that have been removed. On receipt of this update, the relevant “chunks” can easily be identified via the hashing algorithm. The new records can be inserted or records removed as appropriate, and then only those chunks that have been changed need converting to RDF and then re-importing into the triplestore.

5 Summary

In summary, the task of producing the mapping configuration to convert the British Museum model into CIDOC-CRM compatible RDF has been challenging and time consuming, not least because of the need to become familiar with the Cultural Heritage domain. However, having produced the mapping, others more familiar with the domain are likely to be able to recognise similar patterns within their own datasets and will be more able to adjust to fit or create their own mappings. Overall, the CIDOC-CRM model has proved to be very suitable for representing the concepts and relationships required, with relatively few additions made to capture additional specific British Museum data.

The makeRDF tool has been used to successfully convert the British Museum collection data to RDF suitable for hosting as Linked Data, and subsequently this has been loaded into the RKB Explorer Linked Data Platform. This feasibility work has demonstrated that the process is indeed possible, and has given a useful model and configuration mapping for future work.

6 References and Links

1 http://collectionstrust.orangeleaf.org/schema

2 http://creativecommons.org/publicdomain/zero/1.0/

3 http://crm.rkbexplorer.com/mapping/sample.xml

4 http://librdf.org/raptor/

5 http://linkeddata.org/guides-and-tutorials

6 http://www.cabinetoffice.gov.uk/media/301253/puiblic_sector_uri.pdf

7 http://www.cidoc-crm.org/

8 http://www.cidoc-crm.org/cidoc_core_graphical_representation/graphical_representation.html

9 http://www.cidoc-crm.org/cidoc_graphical_representation_v_5_1/graphical_representaion_5_0_1.html

10 http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.2.pdf

11 http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs

12 http://www.cidoc-crm.org/tutorials.html

13 http://www.opendatacommons.org/licenses/pddl/1-0/

14 http://www.w3.org/DesignIssues/LinkedData.html

15 http://www.w3.org/RDF/Validator/

16 http://www.w3.org/TeamSubmission/turtle/

17 http://www.w3.org/TR/rdf-primer/

18 http://www.w3.org/TR/rdf-syntax-grammar/

19 http://www.w3.org/TR/rdf-testcases/#ntriples

20 http://www.w3.org/TR/skos-reference/

21 http://www.w3.org/TR/xpath/

22 http://www.w3.org/TR/xslt

23 http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/