

The Australian National University  
COMP 6701 eScience Project

# Semantic Web for Museums Final Report

Version Date: November 15th 2006



# Table of Contents

<b>TABLE OF CONTENT.....</b>	<b>3</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>6</b>
<b>ABSTRACT.....</b>	<b>7</b>
<b>GLOSSARY OF TERMS.....</b>	<b>8</b>
<b>1. INTRODUCTION.....</b>	<b>10</b>
1.1 MOTIVATION.....	10
1.2 SEMANTIC WEB TECHNOLOGIES.....	11
1.3 SEMANTIC WEB PROTOTYPE SYSTEM.....	14
1.4 INTEGRATING THE WHOLE SYSTEM.....	15
<b>2. REQUIREMENTS.....</b>	<b>17</b>
2.1 CHAPTER OVERVIEW.....	17
2.2 CLIENT REQUIREMENTS.....	17
2.3 USER REQUIREMENTS.....	17
2.4 TOPICS.....	18
<b>3. PROJECT PLAN.....</b>	<b>19</b>
3.1 DELIVERABLES.....	19
3.2 MILESTONES.....	19
3.3 LIMITS AND CONSTRAINTS.....	19
3.4 RESOURCE PLAN.....	19
3.4.1 <i>Technical Resources</i> .....	19
3.4.2 <i>Hardware Resources</i> .....	20
3.4.3 <i>Human Resource</i> .....	20
3.5 PROJECT SCHEDULE.....	20
3.6 RISK ANALYSIS.....	20

3.6.1	<i>Risk Assessment Form</i> .....	20
3.6.2	<i>Risk Response Matrix</i> .....	20
<b>4.</b>	<b>DESIGN ANALYSIS AND MODELLING</b> .....	<b>22</b>
4.1	CHAPTER OVERVIEW.....	22
4.2	TRANSFORMATION ANALYSIS.....	22
4.3	INTEROPERABLE MUSEUM COLLECTIONS.....	22
4.4	WHY TRANSFER DOCUMENTS TWICE.....	24
<b>5.</b>	<b>TRANSFORMATION SPECIFICATION</b> .....	<b>27</b>
5.1	CHAPTER OVEREIVEW.....	27
5.2	XSLT TECHNOLOGY.....	27
5.3	MAPPING FROM ARBITRIARY XML TO STANDARD XML.....	27
5.3.1	<i>Mapping to standard XML for Museum Victoria</i> .....	27
5.3.2	<i>Mapping to standard XML for Berndt Museum</i> .....	28
5.4	MAPPING FROM STANDARD XML TO STANDARD RDF/XML.....	29
<b>6.</b>	<b>TESTING</b> .....	<b>31</b>
6.1	TEST ENVIRONMENT.....	31
6.2	TEST DESCRIPTION.....	31
6.3	CHALLENGES.....	31
6.4	RECOMMEND IMROVEMENTS.....	32
6.4.1	<i>The Prototype System</i> .....	32
6.4.2	<i>The Current Museum Database</i> .....	32
<b>7.</b>	<b>USER GUIDE</b> .....	<b>33</b>
7.1	CONFIGURATION THE SERVER SIDE.....	33
7.2	ADDITIONAL CONFIGURATION.....	33
7.3	IMPLEMENTATION EXAMPLE.....	33
<b>8.</b>	<b>CONCLUSIONS</b> .....	<b>36</b>
<b>9.</b>	<b>FUTURE WORK</b> .....	<b>37</b>

**APPENDIX A.....38**

**REFERENCE.....40**

Figure index

Figure 1-1: The working process for the Semantic Web prototype System.....15

Figure 1-2: The Integrated Semantic Web Search.....16

Figure 4-1: The data transfer from the client side to the server side.....23

Figure 4-2: XML document transfer process.....24

Figure 4-3: The benefit of doing two step transformation (1).....25

Figure 4-4: The benefit of doing two step transformation (2).....26

Figure 7-1: Search result before the modification.....34

Figure 7-2: Search result after the modification.....35

Table index:

Table 1 - Initial Plan.....19

Table 2 - Actual Procedure.....19

Table 3 – Project Schedule.....20

Table 4 – Risk Assessment.....20

Table 5 – Risk Response Matrix.....21

Table 6 – Mapping Table: from MV Schema to standard SWM Schema.....28

Table 7 – Mapping Table: from standard SWM schemas to standard RDF/XML.....29

Table 8 – Comparison of the files for MV and BM.....33

## Acknowledgements

I would like to thank my client and technical supervisor, Mr Tom Worthington, for leading, helping and supporting my study in Semantic Web technology and researching, implementing, and enhancing the Semantic Web for Semantic Web museum prototype system. Also, I would like to thank my academic supervisor, Dr. Alistair Rendell, for giving me many suggestions and helps in this field from academic perspective. Moreover, I would like to thank Katie Hayne, who contacted the museums for getting the real museum database records for this project, and also thank her for spending a lot of time to read my report and giving me lots of comments to correct my English grammar.

Finally, thank my husband Jie, who gives me the endless support and help during the project progressing period.

## Abstract

Currently, museums around the world are located at various geographic locations, using proprietary database management systems, with own customized database schemas. As a result, the cultural objects collected by different museums have the totally different data structures and schemas in each database system. It means that the Semantic Web for Museum System is hard to integrate and interoperate cultural heritage records information without a widely agreed XML vocabulary standard among museums. The Semantic Web for museum prototype system was developed by Junren Lei at first semester of year 2006. The standard schemas defined in this prototype system have been used as a standard for transferring the two sets of records in XML format from two museums to the standard RDF document. The testing of the implementation of the new generated RDF files shows the Semantic Web Prototype System can do semantic searching across different museums, and the semantic web technology has large potential benefits to offer.

**Keyword:** Semantic Web, Metadata, RDF, Museum, Collection, Standard

## Glossary of Terms

CIDOC CRM	The International Committee for Documentation of the International Council of Museums Conceptual Reference Model
Metadata	Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use or manage an information resource. Metadata is often called data about data or information about information. (Hodge 2001:P3)
OWL	Web Ontology Language. Where earlier languages have been used to develop tools and ontologies for specific user communities (particularly in the sciences and in company-specific e-commerce applications), they were not defined to be compatible with the architecture of the World Wide Web in general, and the Semantic Web in particular. [10]
RDF	Resource Description Framework integrates a variety of applications from library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos, and events using XML as interchange syntax. [8]
RDF Schema	RDF Schema is a language for describing vocabularies in RDF. RDF Schema is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources. RDF Schema vocabulary descriptions are written in RDF using the terms described in this document. These resources are used to determine characteristics of other resources, such as the domains and ranges of properties. [9]
URI	Uniform Resource Identifier is a formatted string that serves as an identifier for a resource.
W3C	World Wide Web Consortium
WWW	World Wide Web

XML            Extensible Markup Language

XSLT           Extensible Stylesheet Language Transformations.

# 1. Introduction

## 1.1 Motivation

Nowadays, with the rapid development of web technology, Internet use has become an important part of life style. To make accessing the information easy and convenient, organizations built their websites which include organization's introduction, product's information, with other information. They then published the web page to the Internet for public access. Some organizations like museums, which hold large number of pictures, sounds, and film collections, have built their own websites for their museum and collections' information. However, because of the special characteristics of the cultural collections within museums - the rich associations, museums are seeking better web technologies to represent their collections' information more efficiently and accurately. The external users are able to identify the searching contents clearly and get more efficient and accurate replies from the Semantic Web server.

The contents of cultural collections in museums are more rich and complex than other entities. Collections have rich associations with many features, such as historical, environmental, and societal data. For instance, a hat designed by person A, and produced by company B during a certain period, was popular used in certain cities, collected by person C, then the hat is preserved in museum D now. This widely associated information between entities is an important aspect of the cultural collections themselves.

The numerous associations' among collections may be preserved in different museums. For example, all the related information about that hat may not only available in the museum which the actual hat located at; the information of that hat may be spread around the other museums. One museum may only holds the information of person A, and another museum may store the details of the company B, due to documentation of another hat item. In this case, the original web technologies are unable to present the wide association between museum collections. Therefore, museums need new technologies to gather the information together for cultural heritage items, from various museums around world, and retrieve them as a whole.

Currently, physical museums are located in various geographic locations, using proprietary database management systems, with own customized database schema. The cultural object data collected by different museums are difficult for the Semantic Web System to integrate and interoperate. For example, museum A defines a schema, “name” of the collection as “Caption”, museum B defines it as “Title”, and museum C defines it as “CollectionName”. As a result, without a widely agreed vocabulary among museums, the Semantic Web System and the museums are not able to understand the varied vocabularies from different museums which actually indicate the same item.

At the first semester of year 2006, the prototype system of the Semantic web for Museum [3] has been developed by Junren Lei. By using this system, the information of the cultural collections in each museum can be integrate and interoperate with other museum collections, and the wide associative information of the collections can be revealed to the user. The standard vocabularies defined in the prototype system will be used as a standard for transferring the XML file with varied vocabularies from different museums to the server side standard RDF (Resource Description Framework) files. In the following section, the basic concepts of the technologies used in semantic web will be introduced.

## 1.2 Semantic Web Technologies

Before we introduce the semantic web, we need to understand how the original World Wide Web works. The original web pages (eg. written in HTML language) were designed for human to human communication. The machine only displays the information on the web pages by using fixed tags. The information on the web page is represented one paragraph followed by another without any meaningful description. That means, the machine does not have any sense to indicate the difference among person name, address, phone number, age, gender, etc. Hence, in other words, the machine is unable to process the information on the web page.

The Semantic Web is an emerging and promising web technology invented by Tim Berners-Lee in 1998[13]. It promotes information integration and interoperability on the web. The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries [17]. The W3school [14] concludes that the semantic web has the following features:

1. Web information that has exact meaning
2. Web information that can be understood and processed by computers
3. Computers to integrate information from multiple web

Berners-Lee states that, the Semantic Web is the next generation of the World Wide Web. However, he points out that the Semantic Web is not artificial intelligence such as it can comprehend like humans; it can only mechanically manipulate the well-defined pieces of information to the end user [13].

Metadata, RDF (Resource Description Framework) and Ontology [21] are the three basic ingredients in the current semantic web technology. Metadata is data about data, which provides a basic description of the raw data. The RDF can be used to describe the metadata, works on top of the metadata, and classifying the vocabularies. Ontology is complementary to the RDF, it not only describes the relationship in the domain, but also defines the domain vocabulary, and even across different domain if mapping between ontologies is constructed and applied.

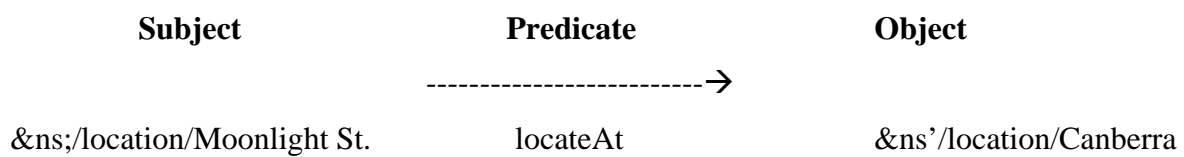
At the metadata level, all data are described by predefined vocabulary set. The vocabulary is defined with a common reference schema [21] (eg. string, date, integer, etc.). The meaning of data can be perceived by both humans and the machine according to the schema. For example, the following information in XML file indicates that “John Smith” is the name of a person, the “Moonlight St.” is the street name, and the “Canberra” is the city name.

```
<person>  
  <name>John Smith</name>  
  <street>Moonlight St.</street>  
  <city>Canberra</city>  
</person>
```

A simple description of the raw data is not enough to represent the real world information. The machine is unable to know the association between the Moonlight Street and the Canberra City. Exposing the global semantic associations between collection items is one of the main goals of our Semantic Web for Museum project. The association between entities is more important than the metadata of the raw data. RDF which standards for Resource

Description Framework is W3C recommend for enhancing the expression of the metadata. The main purpose of RDF in the semantic web is to classify the objects defined in the XML document. The object's association can be expressed in the RDF level.

The resource described in RDF is identified by URI (Uniform Resource Identifiers). The RDF statement includes three components, which are subject, predicate and object, called triple. For example, the statements "The Moonlight St. is located at Canberra"; the Subject "Moonlight St." is identified by URI: &ns;/location/Moonlight St, the Object "Canberra" is identified by URI: &ns;/location/Canberra, and the predicate of the object is "locateAt".



The previous metadata example can be written in RDF as following:

```
<person rdf:about="http://www.w3.org/Peole/ John Smith">
  <name> John Smith </name>
  <street> Moonlight St. </street>
</person>
```

```
<location rdf:ablout="http://www.w3.org/location/Moonlight St.">
  <type>street</type>
  <identifier>Moonlight St.</identifier>
  <locateAt rdf:resource="http://www.w3.org/location/Canberra"/>
</location>
```

Using the RDF, the association and properties of the resource are explicitly expressed. The objects are associated with each other by the RDF expression.

Ontology Web Language (OWL) has the similar structure as RDF, which makes richer relationships and a semantic network of semantic web application. Ontology is not only for describing the rules and relationships in the domain, it also defining the domain vocabulary. The information could be shared across different institutes using domain vocabulary and even across different domain if the mapping between ontologies is constructed and applied. [3] However, the Semantic Web prototype system is only developed to the RDF level, so we are only able to transfer the museum records to the RDF level in this project.

The basic semantic web technology: metadata, RDF, and Ontology have already been introduced; the museums can use those technologies to benefit their semantic web application. The framework of RDF satisfies the relational management of museum collections. The objects defined by the semantic web are both understandable by humans and machine. The machine can process the collection information, and present the results to the end users automatically. The semantic application can infer all the associated information of the objects from other related objects based on the RDF URI. The collections from different museums are not isolated to others any more.

### 1.3 Semantic Web Prototype System

Early 2006, the Semantic Web for Museum prototype system has built by a master student, Junren Lei [3]. The system consists of the following components: Entity, Physical Object,

Representation, Activity, Location, condition assessment, and Collection. Based on the conditions of the two sets of records from the two museums – Museum Victoria and Berndt Museum, we will only use the first five components in this project.

- The **entity** component is the super-class of all the other components; its properties can be adapted to all the sub-classes. The properties include Identifier, Title, Description, Creator, date of creation, and subject.
- **Physical Object** is the sub-class of entity. It inherits all the properties from entity. This component is used to provide the information of the physical museum items, such as the type, the extent, the medium, and belongs to whom, etc.
- **Representation** is the component providing the information of the representation of the museum items, such as the documents used to document, the images used to depict, the videos and audio used to record the physical objects, activities, locations, assessment or collection. The entity properties in object properties are also adoptable for representation.
- The **activity** components representing all the activities concerning museum and object history, such as creation, production, acquisition, transformation, destruction, dissolution, loan, exhibition and movement. Activity is the sub-class of entity.
- **Location** is the sub-class of the entity, used to express the location level, for example, the institute located in a city, and the city being part of a state.

In the Semantic Web prototype system, the data can be imported by two methods. One is the staff at the server side, entering the records from a specific museum manually one by one with RDF format in a text input area on the web page and saving the records to the server repository. Another method is to import a generated RDF document into the server repository with multiple records at once. That means, each museum only needs to provide one RDF document to the server. And, before the data saved into repository, the validation of the RDF data would be checked first by some validate functions in the Semantic Web system.

A straightforward search can be done after the RDF records have been imported to the server repository. In addition, the system provides another function to draw the relationship diagram between the resulting entity and its related entity.

### 1.4 Integrate the whole system

Figure 1-1 shows how the Semantic Web prototype system works, which focuses on the semantic web searching only. The standard RDF file (batch.rdf) or individual item in RDF format needs to be imported to the repository first (step 1 in the figure).

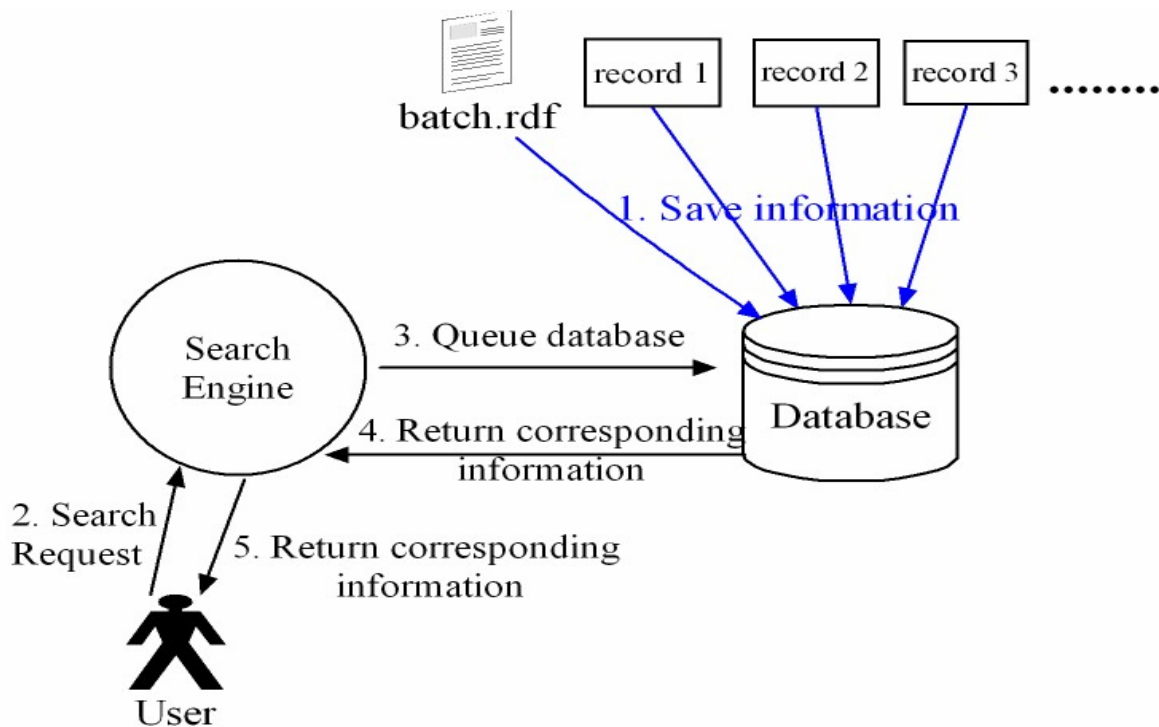


Figure 1-1: The working process for the Semantic Web prototype System:

The RDF data is saved into the database before the user can do the searching. The search engine acts as an intermediate between the user and the database to forward the query and pass the results back to the user.

The figure 1-2 at the bottom shows how the whole semantic web search system works, integrates with the data transfer part and server side searching part. First, various database records from different museums will be transferred to a standard RDF format, and then imported them into server repository.

The first part, data transfer will be discussed later in this project. The XSLT technology will be used for transfer the original museum document to the final RDF document. More details about XSLT technology will be discussed in the later chapters.

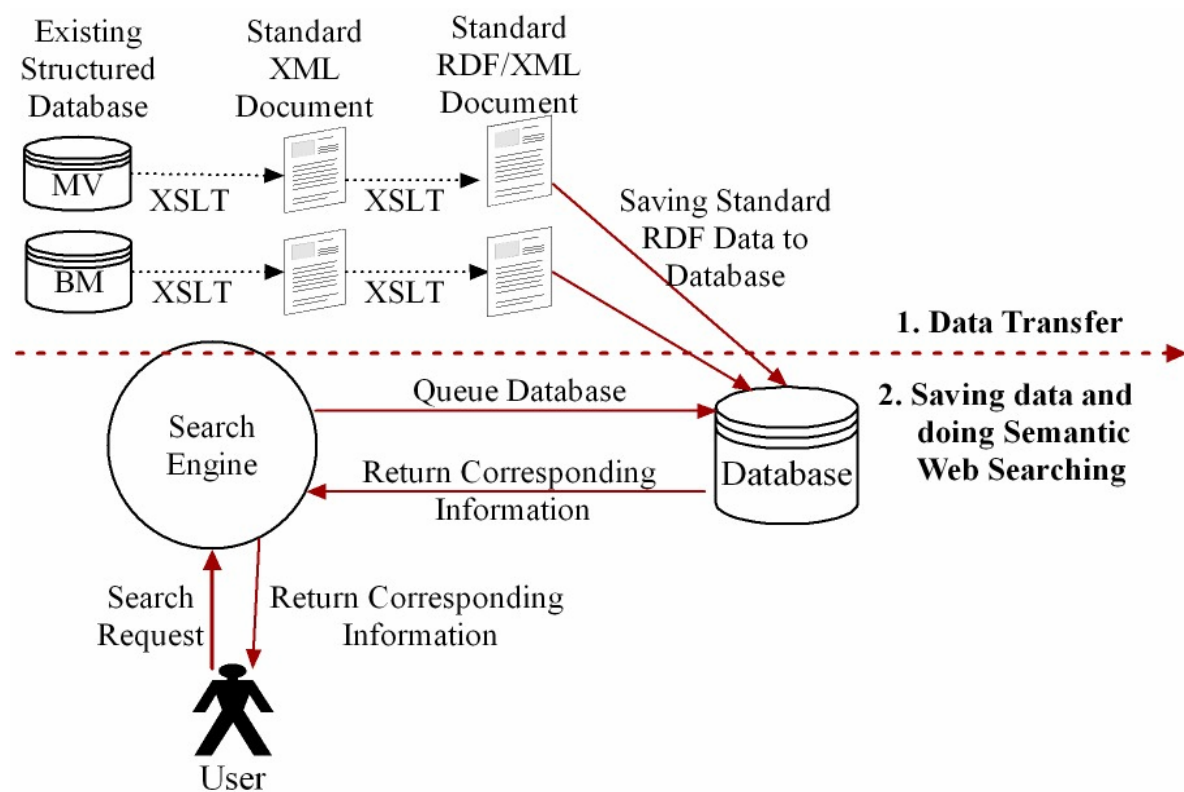


Figure 1-2: The integrated Semantic Web Search

Two steps are involved in the Semantic Web Searching:  
 first step is the data transfer from the arbitrary museum records to the standard RDF/XML document;  
 second step is to save the transferred RDF data to the database, and do the Semantic Web searching.

## 2. Requirements

### 2.1 Chapter Overview

The client requirements come from one of the project requirements from an Australia Research Council project. The title of the project is "Developing a prototype multi-institutional search engine for Australian Indigenous collections (iDig)"[2]. The aim of the project is to develop a prototype search engine to link selected materials from the collections of several museums and demonstrate the potential for more comprehensive linking of indigenous collections in the future. A subset of the requirements from iDig is selected to develop in this project.

### 2.2 Client Requirements

1. Develop a prototype search engine linking two Australian museum collections, which are Museum Victoria [6] (<http://melbourne.museum.vic.gov.au/index.asp>) and Berndt Museum [7] (<http://www.berndt.uwa.edu.au/>).
2. Manage and access materials in photographic format.
3. Transfer the data in the databases into the RDF/XML format.
4. Link the transferred museum RDF to the existing semantic web part (done by Junren Lei in first semester 2006[3])

Junren Lei did the web searching part, and in this project, we will transfer database records from decentralized structured museum databases to a server side standard RDF/XML format document. Different methods maybe discuss according to different situations.

### 2.3 User Requirements

A search engine similar to "Google" will be provided to the user. The semantic web search engine will do more powerful searching than the current search engine. It will provide more efficient searching results and draw the relationship diagrams between data which cross different museums.

## 2.4 Topics

According to the client requirements and user requirements discussed above, the following topics are deduced:

1. Analysis of the Semantic Web prototype system, find the basic methods been used,
2. Develop a method to transfer the database records with arbitrary XML standards from different museums to the server side standard RDF/XML documents,
3. Import the transferred RDF/XML files into the Semantic Web prototype system,
5. Semantic Web searching across multiple records from various museums,

## 3. Project Plan

### 3.1 Deliverables

Project Plan

Project Requirements

Project Analysis and design

XSLT transformation documents

Final Report

### 3.2 Milestones

Table 1 - Initial Plan

Milestone	Date
Initial Presentation	31 July – 6 August 2006
Analysis Completed	27 August 2006
Draft-Project Due	4-10 September 2006
Implementation Due	27 October 2006
Final Presentation	6-12 November 2006
Final Report Due	Wednesday, 15 November 2006

Table 2 - Actual Procedure

Milestone	Date
Initial Presentation	Tuesday, 2 <sup>nd</sup> August 2006
Analysis Completed	3 September 2006
Draft-Project Due	10 September 2006
Implementation Due	27 October 2006
Final Presentation	12 November 2006
Final Report Due	Wednesday, 15 November 2006

### 3.3 Limits and Constraints

Implementation would be limited to a centralized situation, and run under IE 6.0 web browser.

### 3.4 Resource Plan

#### 3.4.1 Technical Resources

XML transformation processor: Saxon XSLT processor [16]

RDF/Ontology tool for support the exist system: Jena.

Tools for supporting the exist system: Tomcat, Java.

Database management system for support the existing system: MYSQL

### 3.4.2 Hardware Resources

Laptop: CPU: Intel Duo Core 1.6GHz

Memory: 1 G SDRAM

Operation System: Window XP with service Pack 2

### 3.4.3 Human Resource

170 hours workload would be cost in this project.

## 3.5 Project Schedule

Table 3 – Project Schedule

1.0	Project Plan & Report
2.0	Requirement
2.1	Requirement Analysis
3.0	Knowledge Learning and Research
3.1	Tools and Technology used in Previous project
3.2	How the existed system works
3.3	XSLT
	(knowledge Learning and Researching throughout all project)
4.0	Building the environment for semantic web system
5.0	Design the mapping schema
6.0	Mapping and transformation
7.0	Test
8.0	Project Closeout

## 3.6 Risk Analysis

### 3.6.1 Risk Assessment Form

Table 4 – Risk Assessment

Risk Event	Likelihood(1-5)	Impact(1-5)
Limited time to study the existed system, and the technology it been used.	5	4
No agreed XML/RDF standard among individual museum	2	3
The time to wait reply from museum	4	1
Records update several times	3	2
Schema applied in the existing system are not very compactable with the schema in the two new museum	1	5

### 3.6.2 Risk Response Matrix

Table 5 – Risk Response Matrix

Risk Event	Contingency plan
Limited time to study the existing system, and the technology it been used.	Concentrate on the related information we need to do this project
No agreed XML/RDF standard among individual museum	Use the standard defined in the existing semantic web system
The time to wait for a reply from the museum	Manually change the part if needed
Records updated several times	Write the XSLT document according to the previous XML document; make change if new version XML document received.
Schema applied in the existing system are not very compactable with the schema in the two new museum	Mapping the museum’s schema to the schema in the existing system as much as possible, leave the unmapped schema for later consideration.

## 4. Design Analysis and Modelling

### 4.1 Chapter Overview

This chapter analyses the situation of current museum systems around the world. Detailed analysis will focus on the data structure of Museum Victoria and Berndt Museum. Mapping table shows the detailed transformation from the arbitrary museum vocabulary to a standard semantic web of museum vocabulary.

### 4.2 Transformation Analysis

In reality, the client museums may locate at different geographic locations, and use different database management systems with different database schemas. It is hard to merge the heterogeneous distributed collections from different museums that participate in the same Semantic Web system in an interoperable way.

The vocabularies applied in Junren's [3] Semantic Web prototype system are Dublin Core [19] mixed with CIDOC CRM [18] model. The RDF vocabularies are based on the metadata as well. To making the whole Semantic Web search system possible, the individual museum system will be asked to translate their database records into RDF format documents, and then stored into the database system on the server.

However, it is impossible to force or persuade every museum to translate the database records into some XML/RDF format. Since, there is no world agreed standard for museum data.

### 4.3 Interoperable Museum Collections

Two museum systems (Museum Victoria [6] and Berndt Museum [7]) are involved in this project, this is a simple case compare to interoperable all of the existing globe museum systems. For now, we can treat each museum system as a single customer, and translate their database record to XML/RDF format individually.

However, some museums may not prefer to disclose the database system to external users. In this case, the individual museum system could write an application or use some XML translation software to translate the database records into some XML documents. The XML documents from different museum may looks totally different with the prototype system

required (Dublin Core mixed with CRM model), because there is no agreed schema standards between museums and the server side Semantic Web system.

As show in Figure 4-1, the server side collects the arbitrary XML document from multiple museums, and then uses vary XSLT documents to transfer the corresponding museum XML files to the standard server side XML documents. After that, the server side transfers all those standard XML documents to the standard XML/RDF documents by using another XSLT document.

10 records from each museum are provided for testing purpose. The two museums are Museum Victoria (MV) and Berndt Museum (BM). Museum Victoria also provided 10 images for the corresponding records. After we went through all the records information for

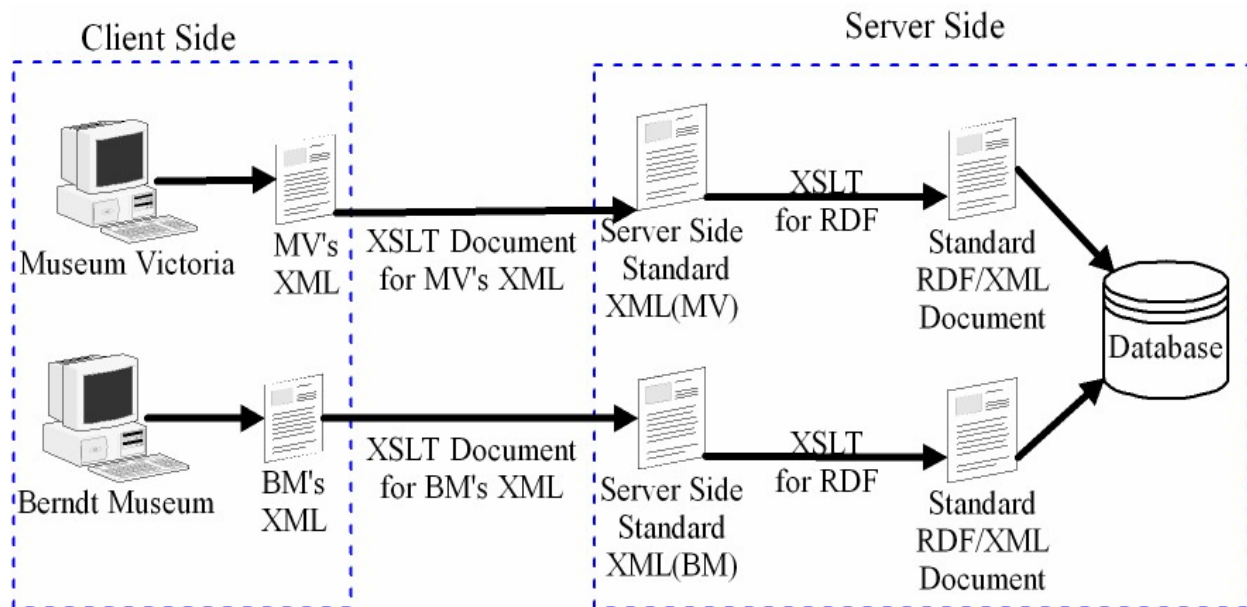


Figure 4-1 - The data transfer from the client side to the server side

The database records already transferred to XML document with the museums' own formats by using special software at the client side. The XSLT technology is used in this project to transfer the documents from the arbitrary museums XML documents to the standard RDF/XML documents.

both museums, we found that the information of the collection records provided by the two museums is totally unrelated. As a result, to achieve the main goal of this project, related semantic search across different museums, we have to modify some data in one of the final transferred standard RDF file. In section 7, User Guide will show you how and where you can modify and get the related information from both museums.

## 4.4 Why Transfer the Documents Twice

We discussed some details about the document transfer from the client side to the server side above. However, it seems over complicated, as it needs transfer twice from the original museum XML document to the final standard RDF document we needed, as show in figure 4-1.

Someone may ask as the documents are all translated by XSLT technology, why not just translate once from the arbitrary XML document to the standard RDF document by using one XSLT file. Well, you can, if you really want to save the step. But imagine, if the system become bigger and the situation become more complicate. For example, the sever side like to generate another RDF file with different RDF classification, or they may just want modify some schema for the RDF document. If only one translation used, those old translated documents are not useful anymore, and a new XSLT document need write up for each museum XML document again. If only three to five museums involved, it is not a serious problem to rewrite the XSLT documents. On the other hand, if hundreds of museums involved, that means, all the works done before are wasted and hundreds of XSLT documents have to be rewritten again. It will cost lots, which include time, money, human resources, and etc.

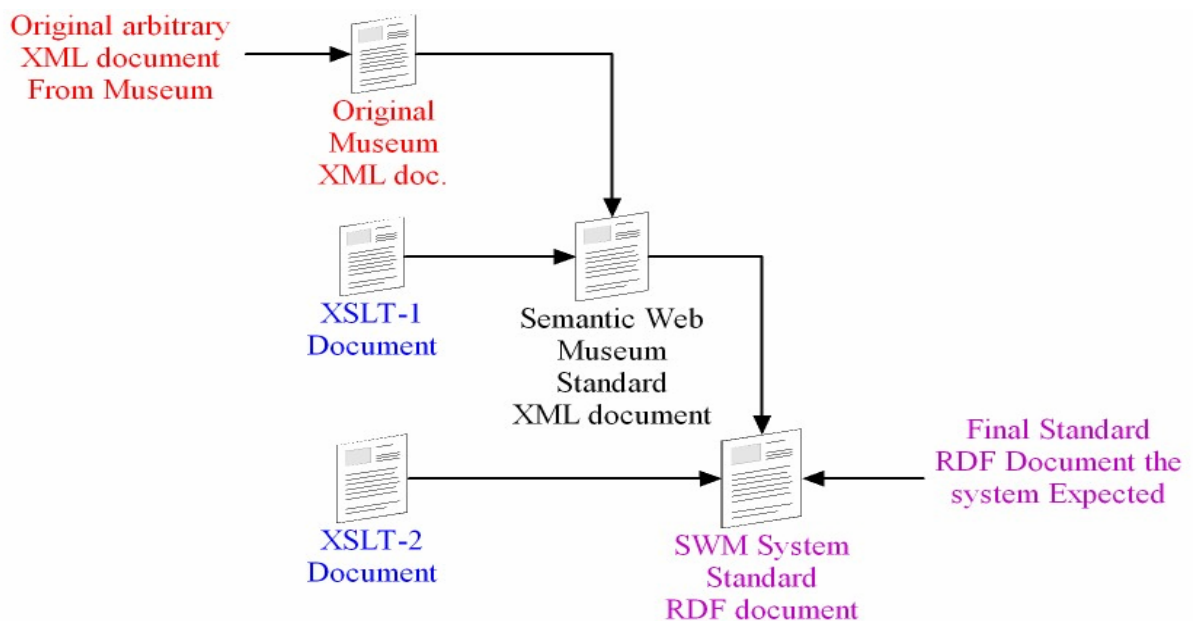
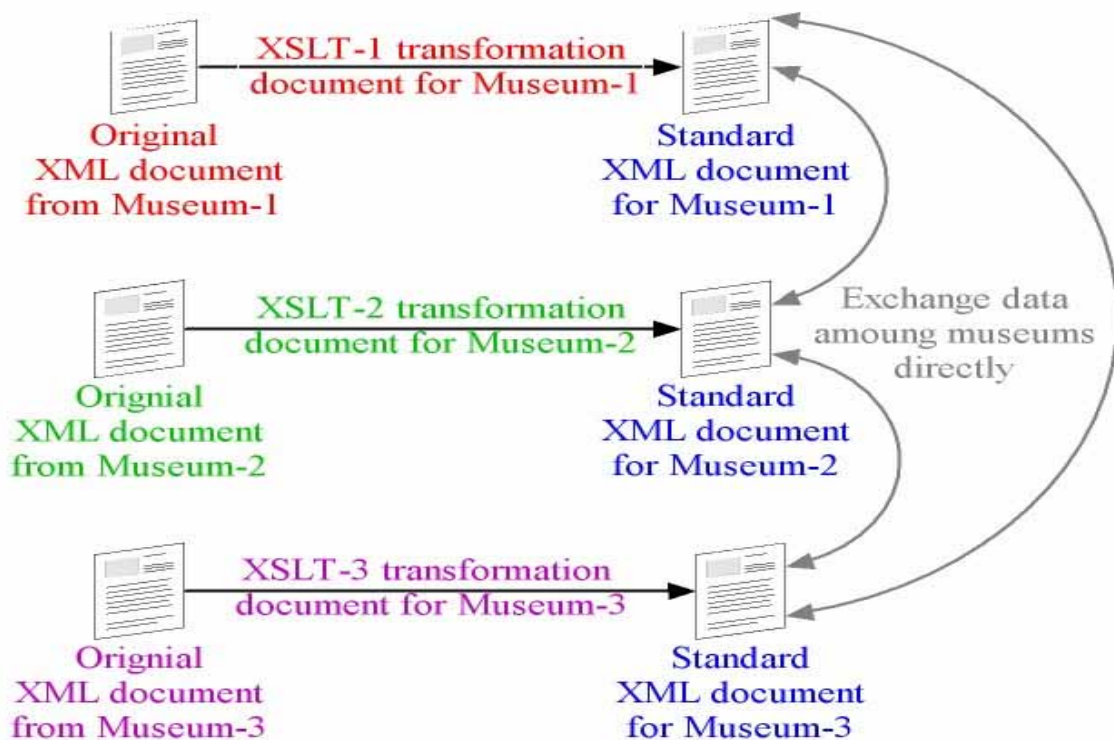


Figure 4-2: XML document transfer process

The transfer process from the original arbitrary XML document to the final standard RDF document. The XSLT-1 document transfers the original XML file from the museum to the standard XML document for the Semantic Web prototype system. The XSLT-2 document transfers the standard XML document to the standard RDF/XML document which will imported to the prototype system.

Our suggestion is to transfer the original XML documents from different museums to the server side regulated standard XML documents first. The benefit of doing this is because of the translated XML documents are all applying the same agreed standard schema for the collection data, the translated documents will be understood by both server side system and other museums. As show in figure 4-2, the museum can send their transferred XML document to each other directly if they needed. The collections' information can be shared within agreed museums if they are in the same semantic web system, whose apply the same agreed semantic schemas.



**Figure 4-3: The benefit of doing two step transformations (1).**

The museums can exchange their collection records directly through HTTP port. Museums can understand each other, as they apply the same agreed standard schemas,

After the documents transferred to the standard XML format, they can be translated to either RDF documents or other format documents (if required). In this case, only one XSLT transformation document needs to be written to transfer all the standard XML documents to the standard RDF/XML documents. If we already transferred the documents to the standard XML format, and even hundreds of museums involved to the SWM system, the server side only needs to modify one XSLT document for new rules, or only generate one new XSLT document if they need RDF data format for another purpose. As show in figure 4-4, all

transferred museum XML documents generate the RDF formate documents (RDF-1) by using the XSLT-4 transformation document. If any modification is required, just modify the XSLT-4 on the server side, (here we save the new XSLT document as XSLT-5). Then all the transferred standard museum XML documents will generate new RDF documents (RDF-2) by applying the new XSLT document.

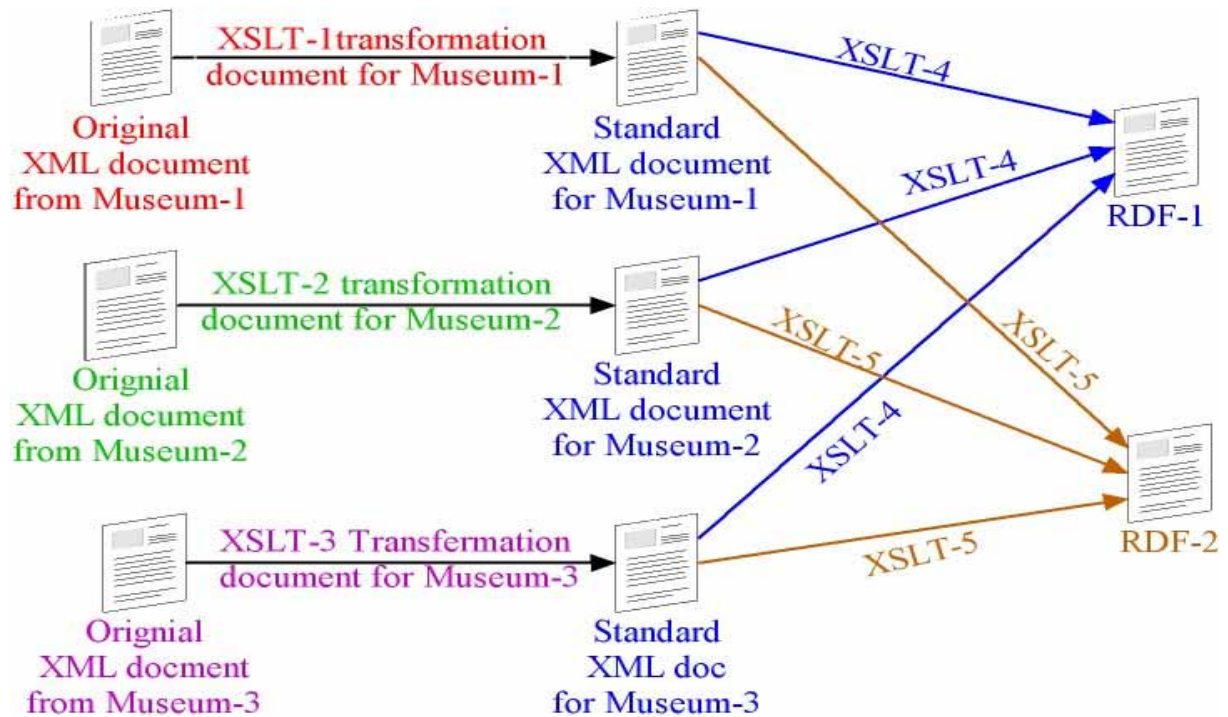


Figure 4-4: The benefit of doing two step transformations (2).

The server side can transfer all the standard XML documents to the standard RDF documents by using only one XSLT document. So if any modification of the RDF records is required, there is only one XSLT document needs to be modified.

## 5. Transformation specification

### 5.1 Chapter Overview

This chapter will explain how we transfer the arbitrary museum XML documents to server side standard XML and RDF/XML documents. Mapping tables will be given for each transformation step.

### 5.2 XSLT technology

The XSLT transformation language is using for transforming XML documents. XSLT, or XSL Transformations, is a vital component to most XML development efforts. It is a powerful language that makes it possible to transform an XML document into a variety of formats, for example, another XML document, and a HTML document for the World Wide Web, and a RDF file or a RTF (Rich Text Format) file as well. In fact, by using XSLT you can convert an XML document into almost any other text-based format you can think of. [25]

### 5.3 Mapping from Arbitrary XML to Standard XML

As we discussed before, the two museum clients (Museum Victoria and Berndt Museum) are located at different cities (Melbourne and Perth), and using different database schemas. Moreover, after we analyse the museum record, we found the Museum Victoria stored the inconsistent information with the same attribute across different records. For example:

This is very common in the real world, when the records were entered by different person with different knowledge base during different time period, the information of the same attribute in one record may have totally different information in another record. It is very hard to identify the exact meaning for each schema in the client museums, which needs mapping to the standard schema for the Semantic Web Museum System.

The inconsistent problem in museum database may need some artificial intelligence agent to solve with. In this project, we have to manually modify some of them for implementation purpose.

#### 5.3.1 Mapping to standard XML for Museum Victoria

Table 6 shows the mapping schema from Museum Victoria to the standard Semantic Web schema, as well as some extra schemes with implied values from MV.

Some database schemas in Museum Victoria are unable to mapping to the standard schema of the Semantic Web system; error will be checked by applying the “not match check” XSLT file.

**Table 6 – Mapping Table: from MV Schema to standard SWM Schema**

<b>MV Schema</b>	<b>Semantic Web Schema</b>	<b>Implied Value</b>
ColRegPrefix + ColRegNumber	DC: identifier	
MulIdentifier	DC: identifier.representation (if image provided)	
DesObjectMedium_tab	DC:type	
DesObjectDescription	DC:descripton	
DesIndividualsIdentified	DC:description	
ComPrimaryComments	DC:description	
ComSecondaryComments	DC:description.representation	
SouMakerRef/ SouPhotographerRef	DC:creator	
SouDateProduced	DCTerm:created	
DesCaption_tab	DC:title	
DesSubjects_tab/DesSubject_tab/ DesDescription_tab	DC:subject	
soucollectionData	DC:data	
ColCollectionName_tab	DC:Contributor.Collection	
souCollectionRef	DC:Contributor	
ProCountry	swm:location.Country	
ProStateProvince_tab	swm:location.StateProvince	
ProRegion_tab	swm:location.Region	
ProSpecificLocality_tab	swm:location.SpecificLocality	
<b>Extra schema with the implied value from MV.</b>		
	DCTerms:requires	Resource
	DCTerms:isPartOf	Museum Victoria
	DC:medium (if image provided)	jpg

### 5.3.2 Mapping to standard XML for Berndt Museum

The records information provided by Berndt Museum is much less than the records information provided by Museum Victoria. The BM has consistent information sorted under the same schema for each record. Because only one schema used in BM XML file, which is

DATA, the implied index of the DATA have to be used to indicate the difference among the schema. The list the standard SWM vocabularies we can transfer are showing below:

- DC:identifier
- DC:type
- swm:location.Country
- swm:location.StateProvince
- swm:location.Region
- DC:description
- DC:Contributor
- DCTerms:created
- DC:creator
- DCTerms:requires
- DCTerms:isPartOf : **Berndt Museum**

#### 5.4 Mapping from Standard XML to Standard RDF/XML

After the standard XML files have been generated, we can start to create the standard RDF/XML documents for the Semantic Web prototype system.

After we analysed the records information provided from the two museums, we decided to generate four groups of information which have been defined in the prototype system for Museum Victoria, which are Physical Object, Activity, Presentation, and Location; and three groups of information for Berndt Museum, which are Physical Object, Activity, and Location. No Presentation group is available in the RDF file as Berndt Museum did not provide any images for their collection objects. Some extra information has been added at this step as well. Table 7 shows the schemas used in each group for the museums.

Table 7 – Mapping Table: from standard SWM schemas to standard RDF/XML

Group Name of Semantic web	Standard Semantic Web Schema
<b>Physical Object</b>	DC: identifier (object attribute)
	DC: identifier
	DC: type
	DC: title
	DC: descripton
	DC: creator
	DCTerm: created
	DC:subject
	Swm:locateAt → DCTerms:isPartOf
	DCTerms:isPartOf
<b>Activity</b>	
Transfer	DC:type (value = “transfer custody”)
	DC:identifier
	swm:locateAt (attribute will relate to a specific country)
	Swm:locateAt → DC:type (value = “new location”)

	→ rdf:value (related to the museum)
	DC:contributor → DC:type (value = “new keeper”) → rdf:value (related to the museum)
	DCTerms:requires → DC:type (value = “transferred”) → rdf:value (related to the related object)
Creation	DC:type (value = “production”)
	DC:identifier
	DC:description
	Swm:locateAt (related to a specific locality)
	DC:date
	DC:creator
	DC:contributor → DC:type (value = “object creator”) → rdf:value (related to the person who created the object)
	DCTerms:requires → DC:type (value = “created”) → rdf:value (related to the object component)
<b>Representation (if image or other documents provided, only for MV)</b>	DC:identifier.representation (as representation attribute)
	DC:type:image (add for all)
	DC:identifier (DC:identifier.representation)
	DC:creator
	DC:subject
	DC:created
	DC:medium (if image provided)
	DCTerms:requires →rdf:value (related to other components) →DC:type (“shows” if it is a image)
<b>Location</b>	
	DC:type
	DC:identifier
	Swm:locateAt (related to the up level if has any)

## 6. Testing

This chapter describes the test implementation of the Semantic Web prototype system by using the real world record from two different museums. The challenges encountered will be listed in this chapter, and the recommended improvements will be given as well.

### 6.1 Test Environment

One laptop, which set up with HP Genuine Intel T2050 was used as testing machine.

Other configurations are:

CPU: Intel Duo Core1.6Ghz,

Memory: 1 G SDRAM

Operation System: Window XP with service Pack 2,

Java Run-time Environment 1.5

Database management system: MySQL 4.1

Application server: Apache Tomcat 5.0

### 6.2 Test Description

When the standard RDF/XML file is generated, the Semantic Web system can import the file. The error information can be found from the log file of the Tomcat Server, and the web browse itself (left bottom of the web browse). Some errors can be fixed by modifying the XSLT document, for other errors we either manually modify the XML source file from museums, or leave the problems and avoid to implement that part.

### 6.3 Challenges

The prototype system currently only supports IE6.0 or above. Other web browses are incompatible with the system. The test laptop installed the Mozilla Firefox web browse, and we found the display of the web page on Firefox is quit different from IE6.0 browse. The prototype system is unable to run under Firefox.

The prototype system did not support carriage return and double quotation. As the information in “description” attribute from the real museum may carry some special

characters, which will cause error in the prototype system. For example, the carriage returns and double quotation characters.

The prototype system did not configure the Java path properly. The Java path configured on Junren's laptop probably is not suitable for other computers depending on various situations such as operating system, environment variables, file path, etc.

## 6.4 Recommend Improvements

Currently, we only can use IE6.0 for testing purpose. In the future, a better web user interface should be developed for supporting multiply web browses.

For the carriage return and double quotation problem, only for demo purpose, we have to delete all the carriage return and double quotation from the museum source files manually. In the future, the carriage return problem should be fixed in the Semantic Web system by modifying the source code; the double quotation problem, the museum staffs have to avoid use the double quotation, because it is a special character in HTML/XML document.

For the Java path problem, we have to manually modify the source code for the Semantic Web system for this project. In the future, a specific configuration file will be needed for Java path configuration.

### 6.4.1 The Prototype System

- Develop a standard XML/RDF vocabulary for museum systems (Australia museum, eg. VM, MB).
- Develop a software application for connivent translate database records to XML or XML/RDF documents.
- Develop an application to assign a globe ID for each collection from different museums.

### 6.4.2 The Current Museum Databases

- Provide more clear and clean data for the collection item.
- Check if consistent information provided in every column.

## 7. User Guide

### 7.1 Configure the server side

The chapter 7 of Lei’s final report has a very clarity server side configure specification; a copy of that chapter 7.1 is in Appendix A.

### 7.2 Additional Configure

Install the Saxon software. You can install the Saxon to anywhere you like, but remember configure the class path for the Environment Variables (in System Properties for Windows XP) before you use it.

Copy both XML source files and XSLT transformation files to the Saxon folder for easy use. All the file names with the directory folders are listed in Table-8.

Table 8 – Comparison of the files for MV and BM

Museum name	XML source file	Standard XML transformation file	Standard XML after transferred	Standard RDF/XML transformation file	Standard RDF/XML after transferred
Museum Victoria	<b>mv.xml</b>	<b>mv1.xsl</b>	mvout1.xml	<b>rdf.xsl</b>	mvout2.rdf
Berndt Museum	<b>bm.xml</b>	<b>bm1.xsl</b>	bmout1.xml		bmout2.rdf
<b>Located folder name</b>	<b>source files</b>	<b>XSLT files</b>	<b>Standard XML</b>	<b>XSLT files</b>	<b>Standard RDF</b>

Then you can type “java -jar saxon8.jar -o [output file name] [source file name] [transformation file name]” under command terminal to transfer the XML files. The following section will show some examples of the implementation.

### 7.3 Implementation Example

Use command window, go to the directory where the Saxon was installed, and type the following command to transfer the Museum Victoria’s source file (mv.xml) to the standard XML file (mvout1.xml).

```
java -jar saxon8.jar -o mvout1.xml mv.xml mv1.xsl
```

After the new standard XML file has been generated for Museum Victoria, type the following command to transfer the transferred standard MV's XML file (mvout1.xml) to the standard RDF/XML file (mvout2.rdf)

```
java -jar saxon8.jar -o mvout2.rdf mvout1.xml rdf.xsl
```

You can transfer the Berndt Museum's source file to the standard RDF/XML file by using those two commands.

You can load the mvout2.rdf or bmout2.rdf file in Batch Storage page to store some data to the Semantic Web prototype system. Searching keyword "Paint, Bark" in Search page and following the link from the results can give you an overview of the search function.

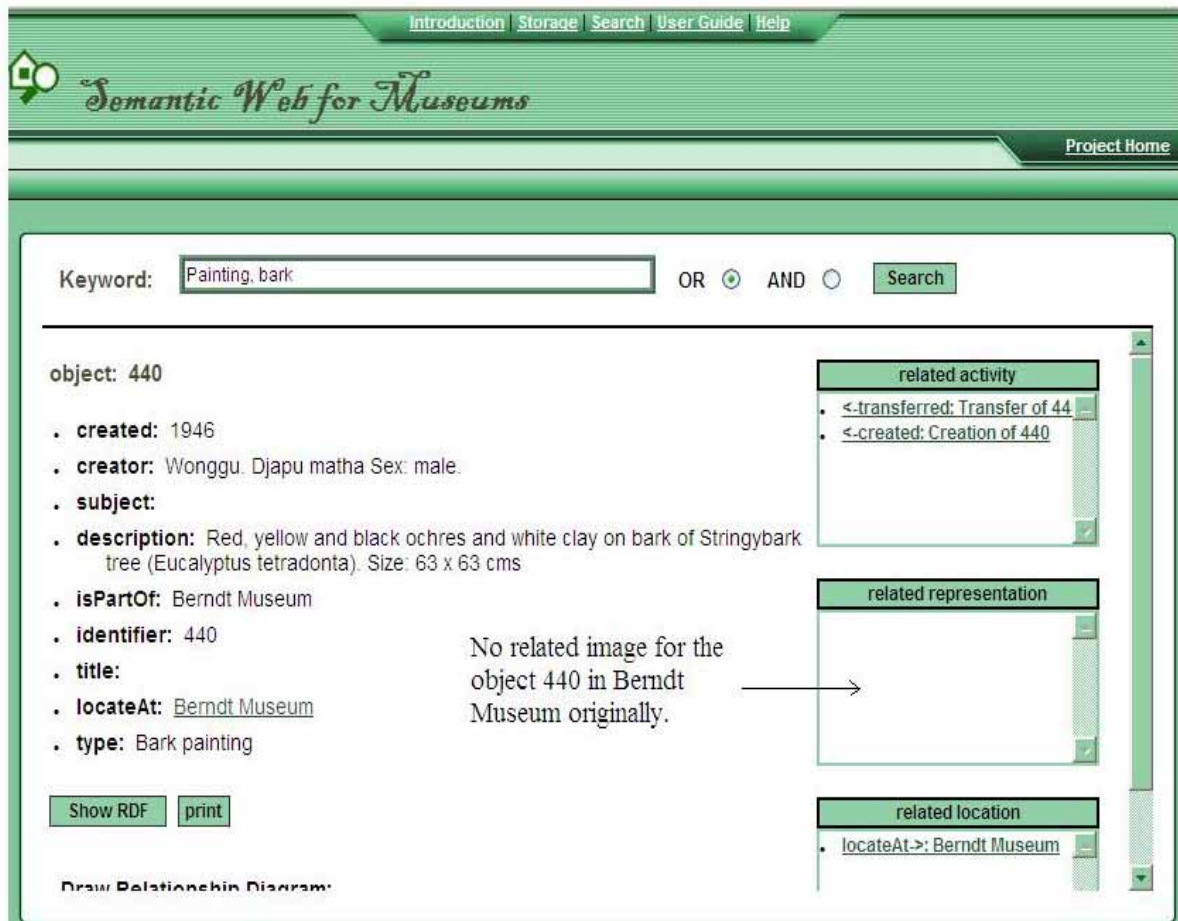


Figure 7-1: Search result before the modification  
 No related image name located in the related representation box.

Now, the information contained in mvout2.rdf and bmout2.rdf still unrelated. We have to make a little bit modification to relate the collection object. You can do many different kinds

of modify if you wish, following show an example of the modification to relating the object’s information from two museums.

Example modification:

1. Open mvout2.rdf file, find the “rdf:value” under the “swm:representation” component in mvout2.rdf file.
2. Modify the MV’s object ID “DC00xxxx” in the “rdf:value” tag to a BM’s object ID, for example “440”. Save the file.
3. Import the mvout2.rdf file into the SWM system again through the storage page.
4. Go to the search page, type “440” for searching. You will find some difference from the pervious searching results.

The Figures 7-1 and 7-2 shows the difference between the before and after the modification.

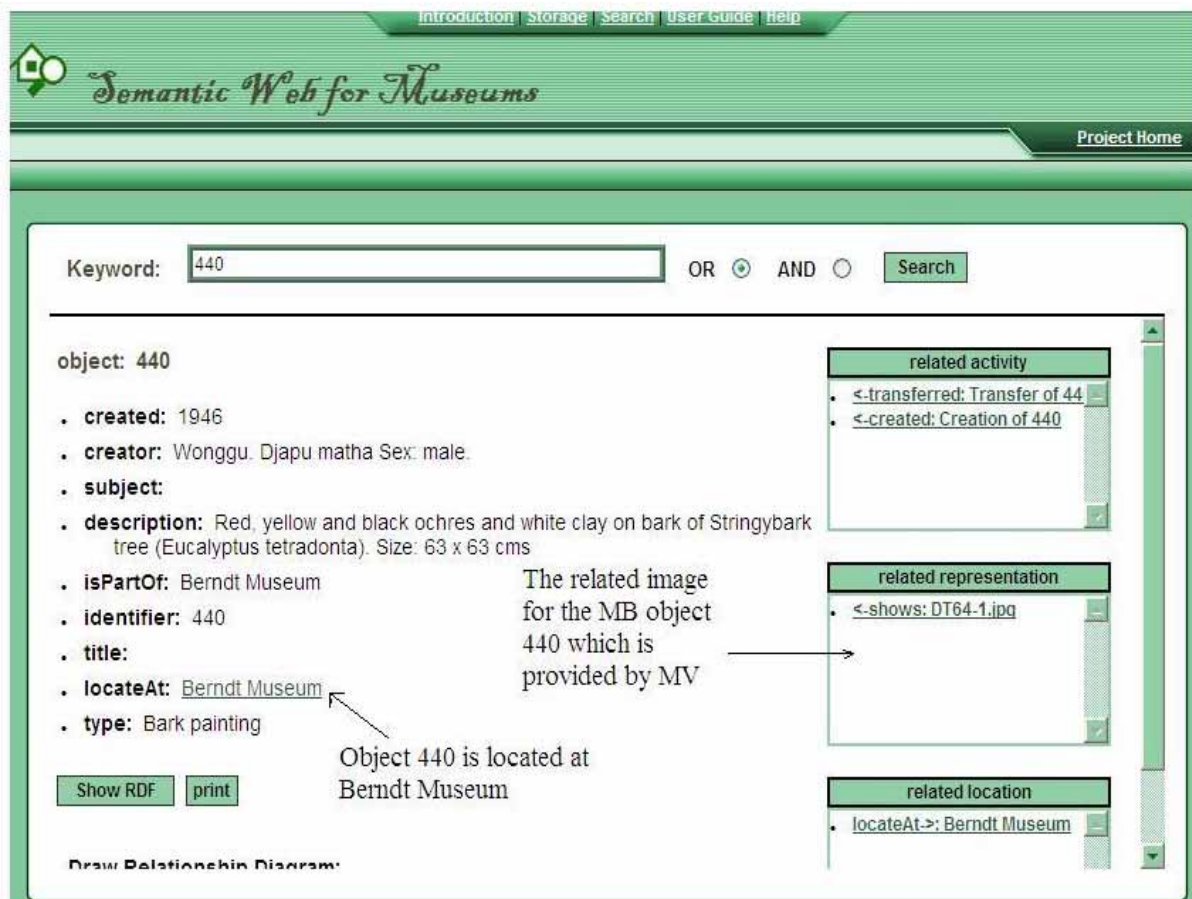


Figure 7-2: Search result after modification

The image “DT64-1.ipg” which from Museum Victoria has been related to the object “440” in Berndt Museum after the modification.

## 8. Conclusion

In this project, two sets of museum records in varied XML format from Museum Victoria and Berndt Museum have been transferred into the standard RDF/XML documents by using the standard which has been defined in Semantic Web for Museum prototype system. The transferred RDF/XML documents have been successfully imported and stored into the prototype system's repository. And the related searching can be done across the two museums after some reasonable modifications have been done for the RDF files. As a result, if more museums and their records involves to this Semantic Web system, the related searching across multiple museums can be achieved without any modify of the files.

The recommend improvements for both the prototype system and the museum databases have been analysed and listed in this report. Better implementation for the Semantic Web searching will appear, if both server side prototype system and the client side museum database systems can make some improvement as we suggested.

In summary, using what Mr. Tim Berners-Lee said, the Semantic Web will be the next generation of the World Wide Web. The searching of the Semantic Web technologies and apply them into the cultural heritage knowledge preservation for museums community should be continued.

## 9. Future Work

We understand that the Semantic Web prototype system developed by Junren is only a prototype system, problems happened when we test the system by using the real world museum record. Improvements need take place to the system for a better implementation. The followings are some recommendations we suggest to improve the prototype system.

- Create an easy configuration way for the Semantic Web system.
- If more museums join in this Semantic Web system, it is possible to modify or even generate new RDF/XML schemas for association the museum records and do more powerful research. Also, develop an ID assign application to reassign globe recognised IDs to the museum records, instead of using the ID from the museum itself.
- Each vocabulary using for the Semantic Web system should have explicit boundary with others.
- The meanings of the vocabularies need to be explained careful and clear to each museum.
- The museum should fully understand the schemas used in the Semantic Web system.

We all understand that, the museum systems in the real world are using different database management system, apply different database schemas, etc. This fact makes hard for the server side to apply the Semantic Web across multiple museums when situation changes from one museum to another. Some artificial intelligence agent for museum system could invent in the future to solve the arbitrary database system and schemas applied in multiple museums.

## Appendix A

### 7. Configuration and User Guide

#### 7.1. Installation and Configuration

##### 7.1.1. Tested Environment

Currently the system has been tested under the environment below.

Hardware: CPU 1600Mhz, 256 MB RAM

Operation System: Window 2000 Server

Web Browser: Internet Explorer 6.0

##### 7.1.2. Server Installation

**Required Software:** Tomcat5.0, MySQL

**Configuration:**

1. Install Tomcat5.0.
2. Open web.xml file in \conf directory; add  
<welcome-file>index.jsp</welcome-file> in <welcome-file-list>.
3. Copy swm.xml file to \conf\Catalina\localhost directory.
4. Copy swm folder to \webapps directory.
5. Install MySQL.
6. Add Database-jenatest, User-test in MySQL  
Database Server URL, User Name, Password, Model Name, schema and rules address  
can be changed in swm/Util/ PageImport.jsp.

Default Value and Example:

```
DB_URL = "jdbc:mysql://localhost/jenatest"; // URL of database server
DB_USER = "test"; // database user id
DB_PASSWD = ""; // database user id
modelName = "batchmodel"; // Model Name in Database
schemaUri = "./webapps/swm/RDF/swm.rdfs"; //Schema address
ruleUri = "./webapps/swm/RDF/batch.rules"; //rules address
```

##### 7.1.3. Library Installation

**Required Software and Library:** Java 1.5 or above, All library of Jena, Jdom.jar, Rome.jar, mysql-connector-java-3.2.0.jar or above.

**Configuration:**

1. Download and install Java 1.5 or above; Copy dt.jar and tools.jar from Java \lib to Tomcat 5.0\common\lib.
2. Download Jena package; Copy all library from Jena\lib to Tomcat 5.0\common\lib.
3. Download Jdom.jar, Rome.jar, mysql-connector-java-3.2.0.jar or above. Copy all these jar files to Tomcat 5.0\common\lib.
4. Copy swm.jar to Tomcat 5.0\common\lib.

##### 7.1.4. Test Installation

Start MySQL and Tomcat service. Try URL <http://localhost:8080/swm/> in your

browser. You would see the introduction page, if the system installed correctly. You can load the swm/RDF/batch.rdf file in Batch Storage page to enter some data to the system. Searching keyword “Textile silk” in Search page and following the link from the results can give you an overview of the search function. You can also enter some items using the example data, which showed on top of each text table, in Physical Object, Activity, Representation or Location Storage pages. You can install TestComplete and open Semantic Web project in test files to test functions and performance of the system.

## Reference

- [1] University of BATH 2003, Leona Carpenter, *OAI for Beginners - the Open Archives Forum online tutorial*  
Retrieved 07 August 2006 from <http://www.oaforum.org/tutorial/english/page1.htm>
- [2] Australian Research Council 2005, Investment Opportunities Detail, *Developing a prototype multi-institutional search engine for Australian Indigenous collections*.  
Retrieved 07 August 2006 from <http://apps.arc.gov.au/arcresearch/research/Detail.aspx?id=SR0566962>
- [3] Australia National University eScience Project S1/2006 , Junren Lei, *Semantic Web for Museum*,  
Final Report: Retrieved 07 August 2006 from <http://escience.anu.edu.au/project/06S1/JunranLei/JunranLeiFinalReport.pdf>  
Project Artefacts: Retrieved 07 August 2006 from <http://escience.anu.edu.au/project/06S1/JunranLei/JunranLeiProjectArtefacts.zip>
- [4] Archives & Museum Informatics, David Bearman and Jennifer Trant (eds.), *Museums and the Web 2004 – Finnish Museums On The Semantic Web*,  
<http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html>
- [5] W3C 1999, *XSL Transformation (XSLT) Version 1.0*,  
Retrieved 21 2006 from <http://www.w3.org/TR/xslt>
- [6] Museum Victoria 2002,  
<http://melbourne.museum.vic.gov.au/index.asp>
- [7] Berndt Museum of Anthropology 1998,  
<http://www.berndt.uwa.edu.au/>
- [8] Systinet 2000, Miloslav Nic, RDF Tutorial – Part I: Basic syntax and containers  
<http://www.zvon.org/xxl/RDFTutorial/General/book.html>
- [9] Wikipedia, *RDF Schema*  
[http://en.wikipedia.org/wiki/RDF\\_Schema](http://en.wikipedia.org/wiki/RDF_Schema)
- [10] W3C 2004, Ivan Herman, Jim Hendler, *Web Ontology Language (OWL)*  
<http://www.w3.org/2004/OWL/>
- [11] Wiley 2004, David Hunter, Andrew Watt, Jeff Rafter, Jon D... *Beginning XML*
- [12] Hewlett-Packard Development company 2000-2006, *Jena-A Semantic Web Framework for Java*  
<http://jena.sourceforge.net/>
- [13] Tim Berners-Lee 1998, *Semantic Web Road Map*,  
<http://www.w3.org/DesignIssues/Semantic.html>

[14] w3school, Refsnes Data, 1999-2006, *Introduction to RDF*,  
[http://www.w3schools.com/rdf/rdf\\_intro.asp](http://www.w3schools.com/rdf/rdf_intro.asp)

[15] Tim Berners-Lee 1998, *What the semantic Web can Represent*  
<http://www.w3.org/DesignIssues/RDFnot.html>

[16] Saxon XSLT Processor  
<http://saxon.sourceforge.net/>

[17] W3C, Ivan Herman, 1994-2006, *Semantic Web Activity*  
<http://www.w3.org/2001/sw/>

[18] CIDOC 2005, *Definition of the CIDOC Conceptual Reference Model*, Version 4.2  
Retrieved 05 April 2006 from [http://cidoc.ics.forth.gr/official\\_release\\_cidoc.html](http://cidoc.ics.forth.gr/official_release_cidoc.html)

[19] DCMI Dublin Core Metadata Initiative 1999, *Dublin Core Metadata Element Set, Version 1.1: Reference Description*  
<http://dublincore.org/documents/dces/>

[20] DCMI Dublin Core Metadata Initiative 2005, *Using Dublin Core - Dublin Core Qualifiers*,  
<http://dublincore.org/documents/dces/>

[21] ILRT, 2000, Libby Miller, *Ontologies and Metadata*  
<http://ilrt.org/discovery/2000/11/lux>

## XSLT Tutorial

W3school, Refsnes Data, 1999-2006, *XSLT tutorial*  
<http://www.w3schools.com/xml/default.asp>

Roger L. Costello. 2000 , GNU GENERAL PUBLIC LICENSE,  
<http://www.xfront.com/xsl.html>

[25] SYBEX 2002, Chuck White, *Mastering XSLT*