Towards Semantic Modelling of Cultural Historical Data

In this paper a practical method is presented for creating documentation of cultural historical targets using an event-centric core ontology. By using semantic documentation templates and an XML-based query language, a domain specific documentation model can be created and flexible user interfaces can be built easily for accessing and editing the documentation. Keywords. ontologies, cultural historical documentation, information retrieval

Introduction

The most challenging feature of cultural historical data is its great variety. This variety sets challenges for documentation, information retrieval and user interface design in cultural historical data systems. Also, lately the question of information integration and especially that of cross-cultural information exchange have been raised in organisations such as museums, libraries and archives that collect and maintain cultural historical data. To solve some important problems in the field of cultural historical documentation(Bekiari et al., 2005), semantic technologies have been introduced. Conceptual Reference Model (later CRM) is a formal ontology intended to facilitate integration, mediation and interchange of heterogeneous cultural heritage information(Crofts et al., 2006). The model was created by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) on empirical bases from real-world datasets which reflect the special needs of the cultural historical field. CRM also has the status of being an ISO-standard(ISO, 2006). In this paper, an event-centric, CRM-based method for modelling cultural historical data is presented. The method is designed to help cultural historical documentation work by providing tools for semantically aware documentation of cultural historical items and events. In order to demonstrate method, a software called IDA-framework was implemented.

1. CIDOC-CRM and event-centric documentation

CIDOC-CRM consists of 86 classes and 143 properties, and it is meant to be extended by users for more specific domains. An important aspect of the CRM class hierarchy is separation between temporal entities and persistent items (Figure 20). It allows modelling of history as events, with actors participating in those events. Events can, for example, produce or modify persistent items, or persistent items can be used in the events.

1.1.Event-centric model of history

The tradition of documentation in the cultural historical field has been very item-centric(Cameron, 2007). The documentation is organised around physical items, which makes documentation of immaterial objects challenging or even impossible. In the event- centric approach, the root of documentation process is not nec-

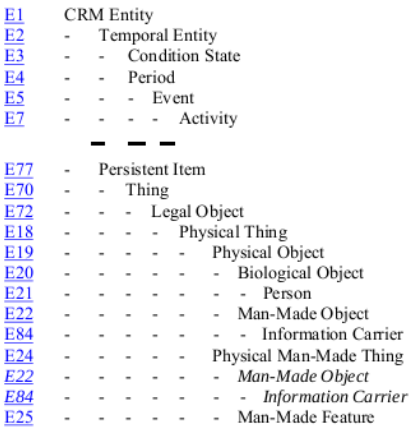

FIGURE 20 A partial CICOC-CRM class hierarchy

essarily a physical item. For example, when a hand-painted painting is modelled according to CIDOC-CRM, the painting has no direct property called author or artists. Instead, the painting is said to have been produced by a production event, which is carried out by one or more persons. Similarly, if painting is sold, damaged, or restored, these processes are modelled as individual events that affect the state of the painting.

The event-centric approach has several advantages compared to traditional, item- based approaches. First, events provide a semantically meaningful way of describing links between physical things and actions of human beings(Doerr and Kritsotaki, 2006). Second, the event- centric model allows a very flexible structure for an individual record. Events describe the history of an item, and new events can be added at any time. In practise, this means that the related documentation can be very detailed or just on a general level and that the level of details can be decided by the person creating the documentation. Therefore, documentation can be constructed based on the qualities of the target of documentation instead of some rigid structure.

Using explicit events also simplifies data structure design, because events have a common structure: someone did something somewhere during a certain time period. With this formulation, it is possible to represent, for example, a creation of an artwork, a construction of a building or having a scientific seminar. Because events are individual records, they split documentation into smaller units, therefore making it semantically more precise and more accessible.

The third, and, from the perspective of documentation, very important benefit is that using explicit events in documentation makes events themselves documentable as units. For example, a design process of a building or restoration of a painting can be documented as an individual record. In the traditional item-centric approach this would need specific fields for every event type. To be able to define the cultural context of an item, it is beneficial if the cultural object can be separated from the physical carrier object(s). For example, in the case of architectural drawings, it can be said that the immaterial architectural design is carried out by physical drawings. This way the actual design can be documented inde-

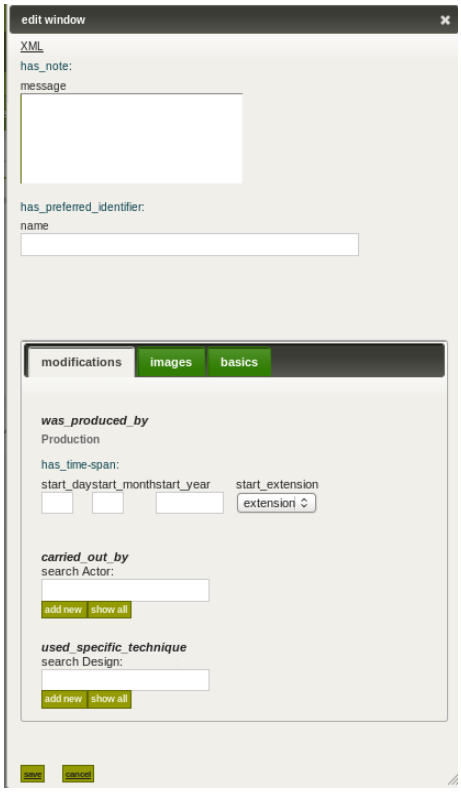

FIGURE 21 User interface created based on the documentation template

pendently from the physical documents. This separation also helps when modelling the relation between industrially made objects and conceptual designs. Industrially made furniture is produced by following a certain procedure or design. The design is documented only once even if there are several physical copies of the furniture.

However, sometimes it is easier to use a shorter path and model the relations in a shorter way. CIDOC-CRM also allows more simple modelling. For example, one way to model the relation between an architectural drawing and its target is to say that the drawing shows a building. Although this is not semantically accurate, since the drawing illustrates the design of the building and not the building itself, it can be useful when the initial data does not contain the necessary information for the design or when there are no resources to make full mappings.

2. IDA-framework

IDA-framework is a simple and flexible tool for making semantically-aware, event-centric documentation(Häyrinen, 2009). The core ontology of the framework is based on CIDOC- CRM. However, the purpose is not to produce full CIDOC-CRM mappings but to provide a semantically meaningful base that follows CIDOC-CRM's principles and conventions. IDA-framework is aimed specially for small memory organisations like local museums and highly specialized museums whose collections are densely linked. IDA-framework uses events explicitly in its doc-

umentation model. The complexity of the CIDOC-CRM is hidden from the end user, and the domain specific documentation structure is defined by semantic documentation templates(Häyrinen, 2008).

2.1.Semantic documentation template

Since CRM does not suggest what to document in any specific case, there must be a mechanism that guides users in making new records(Doerr and Iorizzo, 2008). The idea of a semantic documentation template is to provide a record-type specific documentation frame that can be defined by the organisation responsible for documentation. The template defines a typical case for a record, including default properties and default events. More precisely, the documentation template maps parts of thesauri to a partial domain ontology that is build on top of CIDOC-CRM.

```
<Building >
  <is_identified_by table="appellation" required="1" >
    <name required="0" />
  </is_identified_by >
  <has_note table="note" required="0" action="input" >
    <message required="0" width="200" />
  </has_note >
  <was_produced_by class="Production" required="1" >
    <Production >
      <has_time-span table="time_span" required="1" action="input" >
        <start_day required="0" />
        <start_month required="0" />
        <start_year required="1" width="5" />
      </has_time-span >
      <carried_out_by class="Actor" required="0" action="search_create" />
      <used_specific_technique class="Architectural_Design" required="0" />
    </Production >
  </was_produced_by >
  <has_current_location class="Place" />
</Building >
```

The documentation template does not define any user interface elements. A user interface can be freely created for the template concerned. Figure 21 shows an input form generated by Javascript in the browser. The template helps the user to create an initial target that can be later further refined.

2.2.Semantic query language

For performance reasons, IDA-framework is implemented with the help of a relational database system which excludes the use of Xquery or SPARQL as a query language.

The native query language for relational databases is SQL. While SQL is a well-known and established query language, it is also complex and it requires information about the internal data structure. Another problem with SQL is that queries can get very complex with the recursive database structures that IDA-framework uses. For this project, a new query language called IDA-QL was developed. IDA-QL is very verbal, and it should be meaningful without any prior knowledge about query languages. The following query will give the IDs of all

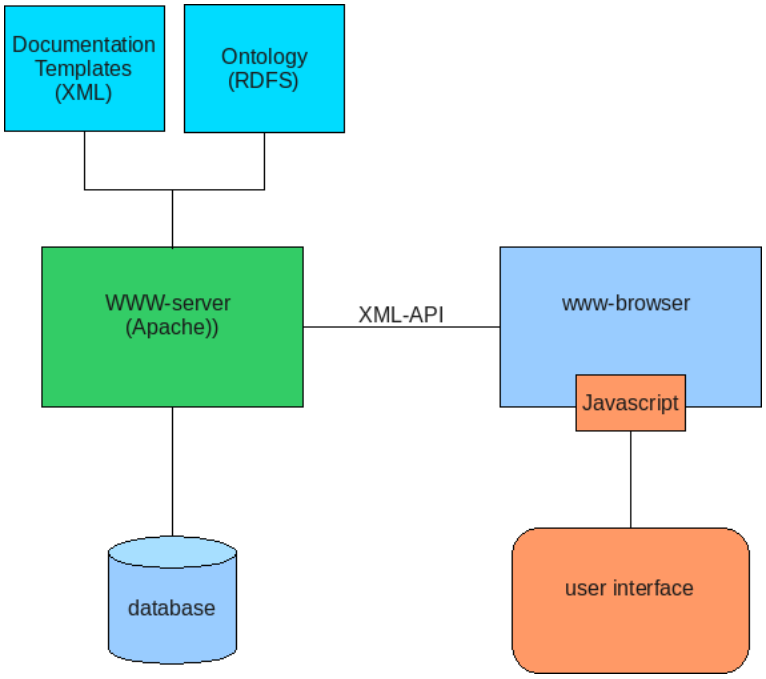

FIGURE 22 IDA-framework system architecture

persons who were born in 1953 and have "Lin*" in their name ("Linus" or "Lintunen", for example):

```
<?xml version="1.0" encoding="UTF-8" ?>
<search>
  <Person>
    <is_identified_by>
      <name>Lin*</name>
    </is_identified_by>
    <was_born>
      <Birth>
        <has_time-span>
          <start_year>1953</start_year>
        </has_time-span>
      </Birth>
    </was_born>
  </Person>
</search>
```

The purpose of IDA-QL is to provide platform independent information retrieval and data manipulation language for cultural historical data. It is a meta query language that is translated to SQL queries. IDA-QL is based on XML, and it operates on classes and properties defined by the ontology. This also means that if the ontology is translated to another language, then also the query language gets translated.

2.3.Implementation

IDA-framework server is written in PHP and it uses a relational database through the MDB2-abstraction layer. The ontology is defined by an RDFS file, and the documentation templates are described in an XML file. Communication between the server and the client is done with XML API. Client applications can be written

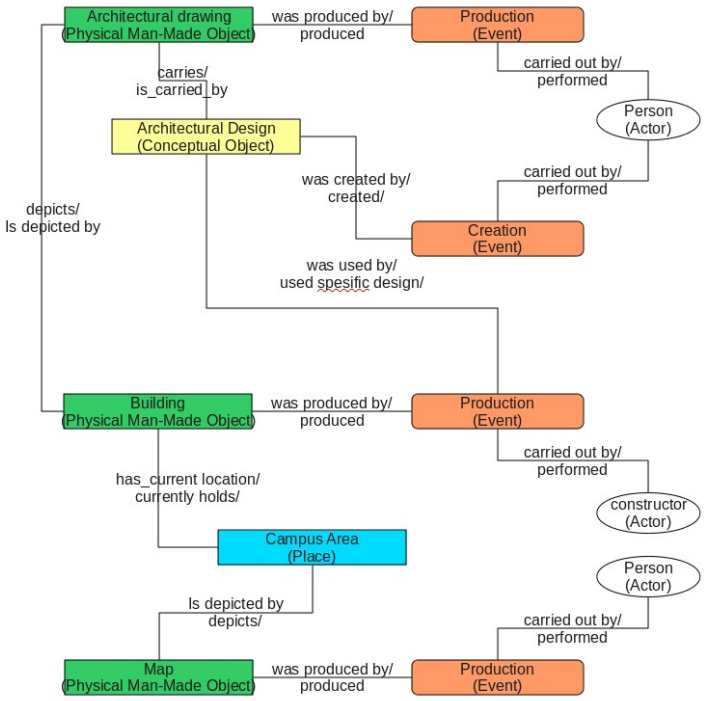

FIGURE 23 Maps and architectural drawings mapped to CRM.

with any programming language as long it can send http calls and is able to parse XML files. The overall implementation is very simple. There are fewer than 6000 lines of PHP code in the server application.

3. Case study: University Museum of Jyväskylä

Demonstrations were made with the collections of the University Museum of Jyväskylä. Two kinds of items were selected from the collections: maps and architectural drawings. There were about 200 maps and 100 architectural drawings in the museum's collection at the time of the experiment. The museum also has a collection management tool called DUO. Direct export form DUO to IDA-framework was not possible because of the differences in the documentation models. Instead, data was exported from DUO to an HTML file, which was then parsed with Javascript. The input was then manually validated and added to IDA-framework.

3.1.Domain Analysis based on CIDOC-CRM

A map depicts some geographical region produced by an individual person or organisation. In terms of CIDOC-CRM, map is a Physical Man-made Object produced by Production event and the Production event is carried out by an Actor. In this case, a simple model was used for modelling the authorship of the map. The producer of the physical map is not necessarily the creator of the map, but in

this case it is the assumption in the initial data. In addition, there is a possibility to define several map subtypes, a road map or a tourist map, among them. As stated earlier, a simple way to model a relation between an architectural drawing and its target is to say that the drawing shows a certain building. While useful in some cases, this model is not semantically accurate. The architectural drawing does not actually present the building, it merely presents the architectural design, which is an immaterial object. This relationship can be modeled with the conceptual class Architectural Design that is carried by the physical object – the drawing. In this case, a class called Architectural Design was derived from the CRM class Design_or_Procedure. This design was then linked to a production event of the building (Figure 23). This kind of modelling solves also the problem of how to document designs that were never carried out, in other words, plans that do not represent any physical building. The instance of Architectural Design exists even if there are no buildings constructed by following the design. The concept of architectural design also serves to organise drawings, because individual documents can be grouped as carriers of a single, named design.

3.2. Demo Applications

In the map application, the user can browse through maps by region, by map maker or by map type. The contents of these navigation panels are created by IDA-QL-queries. The following query gives the names of all persons and organisations who have produced a map:

```
<search>
  <Actor>
    <performed>
      <Production>
        <has_produced>
          <Map/>
        </has_produced>
      </Production>
    </performed>
  </Actor>
</search>
```

The query returns a list of actors. This list is then parsed by Javascript and displayed in a webpage. When the user clicks a certain name in the list, the following query is used to retrieve information:

```
<search result='result'>
  <Map>
    <was_produced_by>
      <Production>
        <carried_out_by>
          <Actor id='1' />
        </carried_out_by>
      </Production>
    </was_produced_by>
  </Map>
  <result>
    <depicts/>
    <has_type/>
    <was_produced_by>
      <carried_out_by />
    </was_produced_by>
  </result>
```

```
</search>
```

The query selects all the maps produced by the actor in question. The properties inside the result tag define what properties are included in the response XML. The XML response is then parsed with Javascript and the actual display is rendered. Architectural drawings can be viewed by buildings, by designs, by architects and by campus areas. Since this application is about architectural drawings, not about buildings, only the buildings that are presented in architectural drawings should be listed. The following query returns the list of buildings that are depicted by an architectural plan:

```
<Building>
  <is_depicted_by>
    <Architectural_Plan />
  </is_depicted_by>
</Building>
```

There is no concept of architect in the ontology used. However, the concept of architect can be defined by an IDA-QL query as a person who has produced a architectural drawing and thus a list of architects can be created.

4. Discussion

The case material was quite limited, and therefore it is too early to draw final conclusions. However, the material shows that the method can be used successfully in the field of cultural historical data, and a semantically rich Web 2.0 application can be built entirely with Javascript using an XML interface provided by IDA-framework. Modifications of the data structure do not require changes in the application code, which makes the system more flexible than other relational databases. The main focus of the software development was in the server side and on developing a flexible way to build user interfaces. Therefore, no actual usability tests were conducted at this point.

5. Conclusion

A practical method for CRM based documentation of cultural historical targets was presented. By hiding the complexity of the ontology with documentation templates, the system adapts semantic technologies in the field of cultural historical documentation. With a simple query language, it is possible to build flexible and browseable user interfaces without expertise needed for other semantic query languages.

Acknowledgements

This work has been funded by the Department of Art and Culture studies at the University of Jyväskylä, by the Finnish Cultural Foundation, and by Kone Foundation.