2008 Annual Conference of CIDOC
Athens, September 15 – 18, 2008
Ari Häyrinen

A TEMPLATE BASED, EVENT-CENTRIC DOCUMENTATION FRAMEWORK

Ari Häyrinen
Department of Art and Culture Studies
University of Jyväskylä
Department of Arts and Culture Studies P.O. Box 35 (JT) 40014 University of Jyväskylä
Jyväskylä
Finland
E-Mail: arihayri@jyu.fi
URL: www.opendimension.org/ida

Abstract

By using explicit events in documentation and ontology based on CIDOC-CRM, it is possible to built a very flexible documentation application. However, CIDOC-CRM itself is too abstract to be used as an information model and it was never meant to be used as such. Therefore there must be a more concrete layer of semantics top of CIDOC-CRM that defines the structure of an individual record. This was accomplished by using documentation templates, which are used to describe special features of different types of documentation targets. The developed framework offers easy-to-use, open-source framework for small organisations and makes adoption of semantic-aware systems possible for wider audience by hiding the complexity of semantic ontologies.

INTRODUCTION

Museums are experts on documentation and good documentation applications should respect that knowledge. If documentation application has a very rigid structure, it may prevent museum to make documentation of their area of interest in an appropriate way. Therefore documentation applications should be flexible in order to fit different kind of purposes even in the same domain. However, this flexibility decreases interoperability between different documentation data since the data structure of applications differs between them. This problem can be solved by mapping data to a core ontology like CIDOC-CRM [2]. Nevertheless, if the data structure is highly dynamic in order to reach required flexibility, the mapping process comes much more difficult. But if there is a shared semantic core and mechanism to define a record type specific semantics top of it, then it is possible to achieve semantic interoperability between different data sets without losing flexibility.

Properties and classes defined in CIDOC-CRM provide a good model for event-centric documentation, even though it is claimed that core ontologies like CIDOC-CRM are not needed in order to achieve semantic interoperability [5]. CIDOC-CRM's strength is that event-centric approach is built in it and it has predefined structure for handling them through properties. CIDOC-CRM has been created on empirical bases from real-world datasets which reflects the special needs of cultural historical field and CRM also has also status of being an ISO-standard [6].

However, very little attention is given to possibilities and possible problems event-centric approach creates for making new documentation. Most of the research efforts and application development around CIDOC-CRM, and semantic interoperability of cultural historical data in general, are related to problems concerning of mapping already existing material to ontologies and theoretical issues behind that process. An event-centric ontology is a working solution for information integration [6] but this approach can also be applied for documentation work itself. By using core ontology that is based on CIDOC-CRM and documentation templates, it is possible to create a very flexible documentation tool that still produces semantically interoperability data.

EXPLICIT EVENTS IN CULTURAL HISTORICAL DOCUMENTATION

An event-centric documentation model means that events are explicitly present to the person who is making a documentation. Explicit event-centric documentation approach is used for example in genealogy application called GRAMPS [4]. Birth, death and marriage are some of the event types that can be added to a person record and events can have their own documentation. GRAMPS can also produce different kind of reports of person. Museo24 -project [1] uses CIDOC-CRM as core ontology and events are explicitly present in annotation tool but that requires at least some knowledge about CIDOC-CRM.

The event-centric approach has several advantages compared to traditional, non-event-based approaches. First, human actions are the cultural context of museum items and that must be expressed somehow in documentation. Events provides semantically meaningful way of describing links between things and actions of human beings [3]. Secondly, the event-centric model allows very flexible structure for an individual record. Events form a history of an item and new events can be added at any time. In practise this means, that documentation can be very detailed or just in general

level and that the level of details can be decided by the person making the documentation. Therefore documentation can be constructed based on qualities of the documentation target, instead of rigid structure that documentation application provides. Using explicit events also simplifies data structure design because events have a common structure: someone did something somewhere in 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.

The third, and maybe the largest benefit of all, is that using explicit events in documentation makes events themselves documentable units. This allows more detailed, and less item-centric, documentation since it is possible to represent the history of the target with different kind of events. For example, it is possible to document design process of buildings or restoration of paintings. Events split documentation into smaller units therefore making documentation semantically more precise and more accessible. The fourth benefit of using explicit events in documentation is the fact that because events are basically time periods, the processing of time is relatively simple. One can create cross-sections of time easily and see what events of which target occurred during that time period. Using explicit events also allows very easy creation of timeline presentations, like Timeline [7]

In order to realise practical application using event-centric documentation model, a dynamic data structure is needed and the user-interface design must reflect that structure. A highest level of documentation flexibility would be achieved if there were just properties that user would pick up as they are needed. This is not a very practical solution because that would require a lot of knowledge about underlying ontologies and properties. Therefore a mechanism is needed for hiding the complexity of ontologies and defining what properties are typical for different kind of records.

DOCUMENTATION TEMPLATE

The idea of a semantic documentation template is to provide a record type specific documentation frame that can be defined by organisation responsible of documentation. The template defines a typical case for a record including default properties, required events, possible events and possible parts. More precisely, documentation template maps a part of thesauri to a partial domain ontology that is build on top of CIDOC-CRM. It is a reversed mapping that is done before actual data input

but with additional information of how documentation should be made. It also serves a guide for a good documentation of a specific type of record.

Here is a very simple example template for a person:

```
<class id="E21" title="Person">
   <property id="P131F" title="is_identified_by" required="1">
       <table id="personname"/>
       <order>1</order>
       <action>input</action>
   </property>

   <property id="P98B" title="was_born" required="0">
       <class id="E67" title="Birth" />
       <order>2</order>
       <action>default_event</action>
   </property>

   <property id="P100B" title="died_in">
       <class id="E69" title="Death" />
       <order>2</order>
       <action>possible_event</action>
   </property>
</class>
```

A person must be identified by name and every person must have a birth event, since this is required for person's existence. Death of a person is defined as a possible event. Birth itself has its own template:

```
    <class id="E67" title="Birth">
        <property id="P96F" title="by_mother">
            <class id="E21" title="Person" />
            <order>2</order>
            <action>search_add</action>
        </property>

        <property id="P4F" title="has_time-span">
            <table id="date"/>
            <order>1</order>
            <hides>
                <field>end_year</field>
                <field>end_month</field>
                <field>end_day</field>
                <field>end_comment</field>
                <field>end_extension</field>
            </hides>
            <action>input</action>
        </property>
    </class>
```

Together these definitions form a template for the simplest possible person record. **Figure 1** shows

Figure 1: The graphical user-interface generated from the person template

user-interface generated from the template. Saving the person record creates two equal records, the person and the birth, which are linked with a property. Both of them can be documented for example with texts and photographs.

After the record is saved, it can be opened for editing, which allows attaching documents and adding new events and parts. When editing the record, possible events for that record are displayed in the menu from where they can be added.

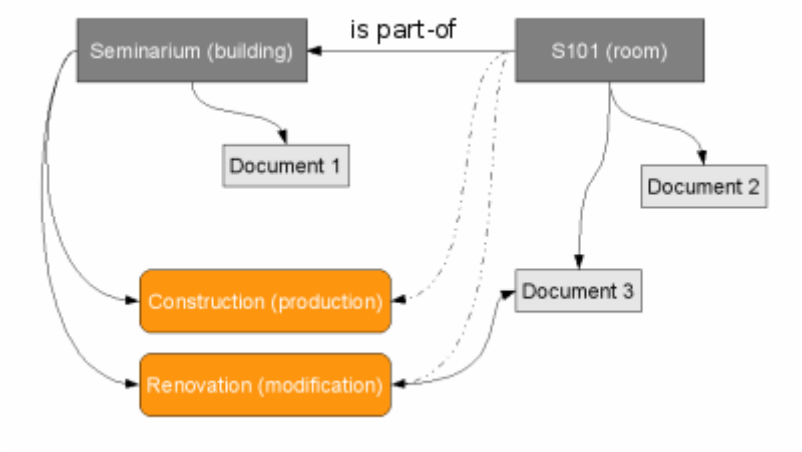

Figure 2: Inheritance of events

Like stated earlier, using explicit events increases the resolution of documentation by splitting the history of a target to events and making those events themselves documentable. However, there must be also way to split target themselves to smaller units. In other words, express relations between targets and their parts. Therefore it is possible to define possible parts in a documentation template. Parts are individual records like any other record, but when for example a room is created as part of a building, it gets linked to the building with a property that expresses a part-of relationship between these two like shown in **Figure 2**.

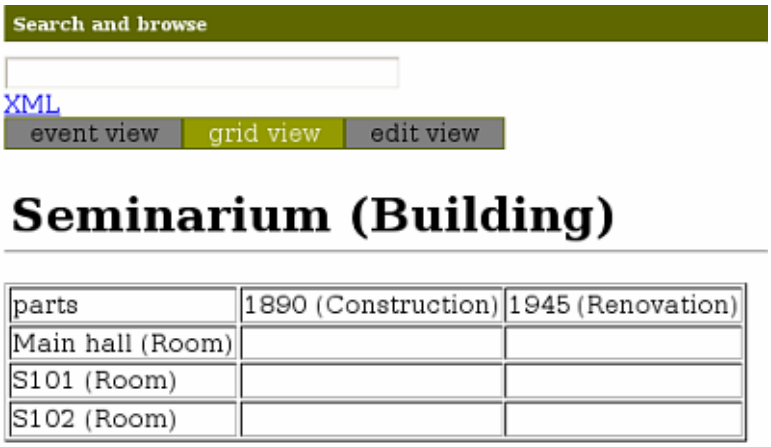

Figure 3: Events and parts of the building presented as a grid. Documents are not displayed.

The problematic aspect of this is how events behave with part-of relationships. If there was a renovation in the building, was there a renovation also in all parts of the building? From the documentation point of view a very practical solution was made. The event of the whole defines a time-period

and by inheriting that event to all parts, it is possible to document what happened to parts during that time period. Figure 2 shows how events behave in this solution. The construction and the renovation events are inherited to the room (S101) from the building (Seminarium) but only the renovation event has a document (Document 3) that is related also to the room. This solution makes documentation faster because there is no need to repeat events for every part and it is easy to provide a table presentation of events and parts like shown in **Figure 3**.

IMPLEMENTATION

An open-source application called IDA-framework was developed in order to test ideas in practise. IDA-framework aims to be a simple and flexible tool for making semantically-aware, event-centric documentation. The core ontology of 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.

The application is divided in two parts: the server and the client. The server-side is written in PHP and it stores the data and takes care of application logic. The server uses relational database through MDB2-abstraction layer. The client is written with Javascript (AJAX) and it is responsible of creating graphical user-interface. Communication between the server and the client is done with XML-files. Using Ajax in the client-side makes overall user interface very flexible. User interface can be built in any layout with just a few lines of Javascript. The same data can be displayed differently just by modifying the script that generates the visual display in browser. It also makes possible to embed database to any website like for example Google Maps.

ACKNOWLEDGEMENTS

This work has been funded by Department of Art and Culture studies in University of Jyväskylä and Finnish Cultural Foundation.

REFERENCES

1. Bognár, K., Karjalainen, M. , Saraniva, A., Szász, B., Unzeitig, M., Cultural Heritage on the Semantic Web – the Museum24 project (2006), available from http://www.seco.tkk.fi/

events/006/2006-05-04-websemantique/presentations/articles/Szasz-museum24Paris.pdf

2. Crofts, N., Doerr, M., Gill, T., Stead, S., Stiff, M. (2006), *Definition of the CIDOC Conceptual Reference Model.* Available from http://cidoc.ics.forth.gr/docs/cidoc_crm_version_4.2.1.pdf; accessed May 2008.

3. Doerr, M., Kritsotaki, A. (2006), *Documenting Events in Metadata*. Available from http://cidoc.ics.forth.gr/docs/fin-paper.pdf; accessed May 2008.

4. Gramps geneology software. Available from http://www.gramps-project.org/wiki/index.php?title=Main_Page; accesses May 2008.

5. Hyvönen, E., Ruotsalo T. (2007), *An Event-based Approach for Semantic Metadata Interoperability.* Available from http://www.seco.tkk.fi/publications/2007/Ruotsalo-Hyvonen-event-based-interoperability.pdf; accessed May 2008.

6. ISO-standard 21127:2006, *A reference ontology for the interchange of cultural heritage information*, available from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=34424; accessed May 2008.

7. Timeline software. SIMILE-project, MIT. Available from http://simile.mit.edu/timeline/