APPLICATION OF A GRAPH DATABASE AND GRAPHICAL USER INTERFACE FOR THE CIDOC CRM

Jonas Bruschke, Markus Wacker

HTW Dresden, Friedrich-List-Platz 1, 01069 Dresden, Germany - (jonas.bruschke, wacker)@informatik.htw-dresden.de

KEY WORDS: documentation, 3d-reconstruction, data exploration, graph database, version control

ABSTRACT:

In archaeology, like in many research fields, collected data reach enormous dimensions and cause more and more efforts to review and to interpret this amount of data and its relations. In this context, digital knowledge databases such as ontologies provide a good basis for the representation and organisation of this data and the related knowledge. In the field of cultural heritage and museum applications CIDOC offers a flexible, standardised, generally accepted, and expandable basis for such a representation. One of the challenges are the different types of the (not only) digital archived material. So far, the input and maintenance of the data, the navigation therein and a suitable data visualisation is exhausting and still concentrating on textual descriptions. For an understandable representation of CIDOC we recommend a graph database equipped with a graphical user interface. The graph database and its way of storing the data are just made for digital representation of an ontology consisting of entities and their relations to each other. Enormous benefits are novel query types, the connection of data and a high query performance. Especially when handling the data (input, maintenance, navigation, visualisation) the role/rights of the user has to be exactly analysed to ensure an interface to the database which is as easy as possible, adapted to the workflow and supporting the user. Especially for archaeological 3d reconstructions we recommend a graphical user interface supporting the work with diverse sources.

INTRODUCTION

When developing an application always the question arises how to manage all the known, available and future information and knowledge which should be provided to the user. This question should be considered carefully especially when designing applications handling lots of data and a multiplicity of relationships between them. Typically, databases respectively database management systems are applied in such cases which allow querying and navigating within the data. Consequently, the question arises how to organise and structure the data within the database. In the context of cultural heritage the CIDOC CRM offers a flexible, standardised, and expandable basis for such an implementation. Relevant research issues consist in the implementation of the structure of the CIDOC CRM within a database and in the adequate presentation of the mass of data to the user.

RELATIONAL DATABASES

Relational databases are the most common type of database and have been in use since four decades now. As all the data is stored in tables, no direct connection between datasets in different tables exists. Semantic data modelling is still possible with relational databases, and several projects prove a successful implementation (Hiebel, 2010; Inkari, 2002). Usually in use are tables with the datasets and so called JOIN tables where relationships between data are stored as a set of two IDs. Strictly following the structure of the CIDOC CRM would mean that there is a table for each class and a JOIN table for each property (Fig. 1).

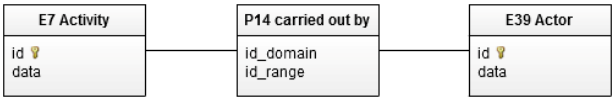

Figure 1: Example of a relational database scheme

However, those relationships are not available on request. The IDs of the datasets need to be compared and matched at runtime. A JOIN creates a Cartesian product of all potential combinations of rows, and then filters out those that are matching the WHERE clause. To gain the interested datasets all relevant tables are processed but in average 99 % of the datasets are discarded (Partner, 2013; Robinson, 2013). Additionally, more data content in those tables will result in more intensive load for processor and memory. Dealing with cultural heritage data and the CIDOC CRM often implicates highly connected data. Usually, the queries are compositions of several relationships, so there would be lots of JOIN clauses within one query (Fig. 2). Handling with several thousands of datasets may then result in a severe performance decrease.

```
SELECT e7.data AS activity,
       e39.data AS actor
FROM e7
JOIN p14 ON e7.id = p14.id_domain
JOIN e39 ON p14.id_range = e39.id
WHERE e7.data = "..."
```

Figure 2: SQL query with JOIN clauses for only one relationship

GRAPH DATABASES

Graph databases are just one representative of NoSQL-databases (Not only SQL) and are especially suited for highly connected data. They consist of nodes and relationships, where the relationships are directed and properties can be stored on each node and relationship (Robinson, 2013) (Fig. 3). The most established graph database is Neo4j. It is transactional and constraints can be assigned to ensure uniqueness. Labels provide a way to attach one or more simple types to nodes and relationships. Neo4j comes with its own query language called CYPHER which is designed with ASCII-art in mind (Hunger, 2014) (Fig. 4).

Figure 3: Basic graph database scheme

```
MATCH (n:E7)-[:P14]->(m:E39)
WHERE n.data = "..."
RETURN n AS activity, m AS actor
```

Figure 4: A simple CYPHER query

An ontology respectively the CIDOC CRM is basically a graph and so the CRM matches exactly the structure of a graph database. The logical conclusion is to use such a graph database for storing and querying cultural heritage data based on the CIDOC CRM. The nodes are labelled with the equivalent entity class and the relationships with the property type. The actual datasets are stored as properties of the nodes. Moreover, there are no IDs to handle anymore as for the relational database case.

In contrast to relational databases graph databases do not have the above mentioned JOIN performance issue as the relationships are stored directly within the database (Hunger 2014). The query starts from a node and then navigates along the relationships to the next nodes (i.e. traversing a graph). Only local operations on each node have to be executed regardless of the total count of nodes and relationships (Partner, 2013). Further benefits are a new types of queries. To comprehend the relation of two nodes, the shortest path between these nodes can be determined. Other queries are sub-tree matching or breadth-first search. In relational databases such queries are rather difficult as the table names have to be explicitly declared and there are no recursive JOIN statements.

The problem with sub-properties

However, one aspect of the CIDOC CRM cannot be directly transferred into the graph database scheme. There are 14 sub-properties specifying a property by connecting it with an *E55* Type entity (ICOM/CIDOC, 2011), e.g. *E7* Activity *P14* carried out by *E39* Actor *P14.1* in the role of *E55* Type (cf. Fig. 5). A traditional graph database can only create relationships between nodes.

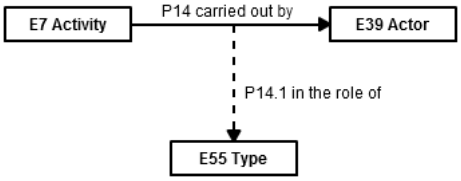

Figure 5: Example for a sub-property

One solution is to split the relationship (Alexiev, 2011) and insert a new node (Fig. 6) which is labelled identically to the relationship and the labels of those new relationships are extended by the literals *a* and *b*. The *P14* node is then connected via the *P14.1* relationship with the *E55* entity. To keep consistency all relationships should be split this way. However, this approach makes the graph and the queries more complex. Only 14 of 149 properties are allowed to have optional sub-properties. Considering cost-benefit ratio this approach is not very practical.

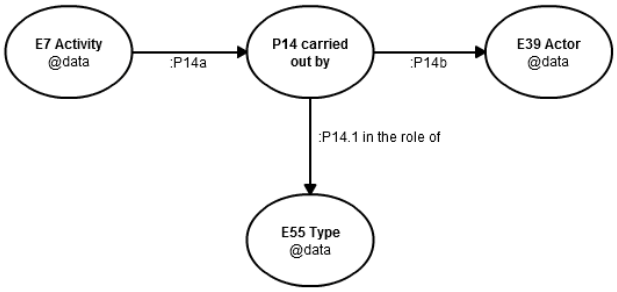

Figure 6: Split relationship with new node

In our approach we consider those sub-properties as a specification of the range entity by creating a relationship between that range entity and the *E55* Type entity (Fig. 7). Of course, this specification is bound to the condition of having the appropriate basic relationship. That's why we introduce an ID for only those relationships. The IDs need to be matched in a WHERE clause within a query. Considering the whole amount of properties compared to only 14 sub-properties, this approach is justifiable. It also enables simple queries, e.g. "Get all roles of an actor", or the other way round "Get all actors which had a specific role".

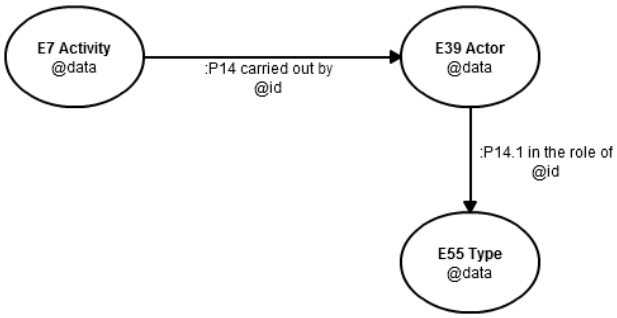

Figure 7: Relationships with an ID property

A documentation tool for digital reconstructions

Nowadays, digital reconstructions are becoming more and more common in archaeology and architecture. They visualize lost, but also present structures, can broaden the comprehension of the reconstructed object and point out historical and constructional relationships. Furthermore the process of reconstruction leads to an aggregation of knowledge and has become a substantial part of scientific work.

However, such projects usually lack of a proper, traceable, and valuable documentation practice. In the final reconstruction state the reference of a source for a certain object may only been known to experts in the project. Understanding from an external point of view often becomes a cumbersome process. Most research for documentation practice is concentrated to theoretical approaches; valuable practical tools are still missing (Pfarr, 2010).

We introduce a documentation tool for 3d reconstruction supposed to accompany a project and to support frequent tasks (Münster, 2014) in digital reconstruction processes. All used sources can be (easily) inserted into the system and connected to the reconstructed objects. Simultaneously, the whole development process is logged automatically. With suitable navigation functionality the user can explore/compare the 3d model together with the sources and information (Fig. 8). Furthermore there is a special mode for briefings: comments of the participating users can be logged and sketches can be drawn directly on (cuts of) the model or plans which will be available for the

Figure 8: Prototype of explorer with 3d model and sources

following modelling process. Version control ensures that edited objects are synchronised with the database to record all development steps and to be able to access older versions of the model. This tool not only may help enormously during the reconstruction process but also can be applied for final presentation of the results to experts or e.g. museum visitors.

All the data storage and processing is achieved by the graph database as described above. Media files or binary files are referenced. However, we want to ensure that the user will not see very much of the CIDOC CRM. If the user is not familiar with this ontology, especially those entities with an abstract nature can lead to confusion: e.g. to draw a connection between an actor and a document, the user would usually say that the actor has created the document. But instead the CIDOC CRM defines a detour via the creation event (E65). Those CIDOC conform connections are automatically created by the tool and the user is not bothered with confusing details. At the end of project it will be possible to export the data to an appropriate format for further processing elsewhere, as a presentation tool for visitors in a museum.

CONCLUSIONS

A graph database is a powerful tool for dealing with lots of highly connected data. It is predestined for the use with cultural heritage data and the CIDOC CRM. Our tool for the documentation of digital reconstructions proofs this concept as valuable. The tool supports the user inserting data and drawing connections. The user does not need to think about the right entity classes or relationships. The tool provides a highly automated documentation of the creation process of a 3d model and finally an application for presentation of the project results for a broader public.

REFERENCES

Alexiev, V., 2011. Types and Annotations for the CIDOC CRM Properties. Sofia, Thesis.

Hiebel, G. (et al.), 2010. "A relational database structure and user interface for the CIDOC CRM with GIS integration." 22nd CIDOC CRM SIG meeting. Nuremberg, Speech.

Hunger, M., 2014. *Neo4j - Eine Graphdatenbank für alle.* Entwickler.press, Frankfurt, M., Book, pp. 7-19.

ICOM/CIDOC CRM Special Interest Group, 2011. *Definition of the CIDOC Conceptual Reference Model.* Version 5.0.4.

Inkari, J. (et al.), 2002. "The Finnish National Gallery Database implementation." 5th CIDOC CRM SIG meeting. Rethymonon, Crete, Speech.

Münster, S., (2014). Interdisziplinäre Kooperation bei der Erstellung virtueller geschichtswissenschaftlicher 3D-Rekonstruktionen. TU Dresden, Dissertation.

Partner, J. (et al.), 2013. *Neo4j in Action.* MEAP Edition, Manning Publications, Book, pp. 3-11.

Pfarr, M., 2010. Dokumentationssystem für Digitale Rekonstruktionen am Beispiel der Grabanlage Zhaoling, Provinz Saanxi, China. TU Darmstadt, Dissertation.

Robinson, I. (et al.), 2013. *Graph Databases.* O'Reilly Media, Sebastopol, Book, pp. 1-23.