Chapter 1
Dealing with Structure Heterogeneity in Semantic Collaborative Information Systems

Eva Zangerle
University of Innsbruck, Austria

Wolfgang Gassler
University of Innsbruck, Austria

ABSTRACT

The creation of content within semistructured, collaborative information systems imposes the problem of having to deal with very heterogeneous schemata. This is due to the fact that the semistructured paradigm does not restrict the user in his choice of nomenclatures for the data he intends to store within the information system. As many users participate in the creation of data, the structure of this data is very heterogeneous. In this chapter the authors discuss two main movements that aim at dealing with heterogeneity. The first approach is concerned with efficiently avoiding structure heterogeneity within collaborative information systems by providing the users with suitable recommendations for an aligned schema during the insertion process. The second approach is mainly focussing on overcoming structure heterogeneity by providing efficient means for querying heterogeneous data.

INTRODUCTION

Most online, collaborative information systems, such as wiki systems, provide means to easily add, modify and delete information, which does not have to adhere to any predefined schema or structure. In contrast, traditional (relational) databases are strictly-structured and enforce the user to store information in a predefined schema. Such structured data stores provide the big advantage of structured access, which enables complex query capabilities. Traditional wiki systems only support full-text search which is not feasible for complex

DOI: 10.4018/978-1-4666-0894-8.ch001

queries such as "Which Austrian cities have more than 10.000 inhabitants and have a female mayor who has a doctoral degree?" Nevertheless, wiki systems are able to cope with very large amount of collaboratively created information with very heterogeneous structures and schemata.

Weikum et al. (2009) observed that modern information systems have to be able to support both structured and unstructured data to combine the advantages of both worlds and be able to answer such complex questions. This need of combination initiates the development of collaborative, semistructured information systems. They provide mechanisms for the combination of both unstructured and structured storage of data. Semistructured data features a structure without having to specify a fixed schema. As this paradigm does not restrict the user and the used schema at all, the massive collaborative creation and editing of content by hundreds or thousands of users obviously leads to the usage of very heterogeneous schemata and structures in collaborative environments. Even Wikipedia, which has a very committed community dealing with heterogeneity, is also not able to avoid heterogeneity within its schema.

In the following sections we discuss the problem of heterogeneity in semistructured information systems and show approaches which are able to deal with heterogeneous schemata, data and the collaborative paradigm of creating and managing knowledge and information.

Schema and Heterogeneity

Modern collaborative Information Systems mostly use the semistructured paradigm, as it features the possibility to structure information without having to adhere to a predefined schema. The most popular example of a semistructured data format is RDF which is often used in the underlying storage layer of semantic, collaborative information systems. RDF consists of triples with the form <subject,predicate,object>. For example information about the subject Albert Einstein can be stored by using the triples shown in Listing 1.

Listing 1. Semistructured description of Albert Einstein

```
<AlbertEinstein,name,Albert Einstein>
<AlbertEinstein,born,1897/03/14>
<AlbertEinstein,bornIn,Ulm>
<AlbertEinstein,wonAward,Nobel Prize>
<AlbertEinstein,wonAward,Max Planck Medal>
```

RDF distinguishes between URIs which describe resources and literals for specifying values. For reasons of simplification, the URI prefixes are omitted in the example in Listing 1. RDF can also be represented as a graph $G = (N, L, E)$ where N contains all nodes. The edges of the graph are defined in the set E with $e(n_1, n_2)$ and $n_1, n_2 \in N$. The possible labels of edges are denoted by L. In the graph representation, subjects and objects are modelled as nodes. The predicates are modelled as labels and the triple $< s_i, p_i, o_i >$ itself is defined by an edge $e(s_i, o_i)$ with the label p_i.

The schema or structure of a graph is defined similar to a classical relational database schema. A database schema is defined by the columns - also called attributes - of a relation. Each row also called record of a relation has to conform to a predefined schema. As semistructured data or RDF data are not restricted in any way, each record can consist of arbitrary many attributes. In the context of RDF a record is called subject and attributes are called predicates. The schema or structure of a RDF subject S_i is defined by the used attributes

$$Schema_{S_i} = \{p \mid p \in L, e(S_1, n) \ with\ label\ p\}$$

In contrast to classical relational databases each record can constitute its own schema. Exactly this feature of RDF is one of the most important