In CD-ROM Proceedings of the WWW2003 Conference, Budapest, Hongary (2003)

Conceptual View Integration for Audience Driven Web Design

Olga De Troyer
Vrije Universiteit Brussel, WISE
Pleinlaan 2
B-1050 Brussel, Belgium
Tel: +32-2-629 3504
Olga.DeTroyer@vub.ac.be

Peter Plessers
Vrije Universiteit Brussel, WISE
Pleinlaan 2
B-1050 Brussel, Belgium
Tel: +32-2-629 37 54
Peter.Plessers@vub.ac.be

Sven Casteleyn
Vrije Universiteit Brussel, WISE
Pleinlaan 2
B-1050 Brussel, Belgium
Tel: +32-2-629 37 54
Sven.Casteleyn@vub.ac.be

ABSTRACT

In an audience driven approach to website design, the requirements of the different audiences are modeled as separated tiny conceptual schemas comparable to views. We present a new approach to integrate these conceptual views. The preparation for the integration is already done during the design phase itself. Semantic information about the concepts used during conceptual modeling is stored in an ontology. Later on, the ontology is used to do the actual integration. This approach has several advantages. The role of an overall domain expert is limited; the ontology can be used to assist the designer during the modeling process; the ontology can be reused in other designs; and the use of an ontology paves the way for the semantic web.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia *architectures, theory.*

H.1.1 [**Models and Principles**]: Systems and Information Theory *information theory.*

General Terms

Design

Keywords

view integration, schema integration, website design, audience-driven

1. INTRODUCTION

WSDM ([2], [3]) uses a so-called 'audience driven' approach to web site design. The design process takes into account that different types of users (audience classes) may exist and that they may have different needs and requirements. WSDM starts the design process with the identification of the different audience classes and the description of their needs and requirements. Next, the different requirements are modeled separately, resulting in a number of schemas called *chunks*. Every chunk models one requirement of a specific audience class. To link the information modeled in the different chunks, all chunks are integrated into a single information model, called the *Business Information Model*. In this paper, we present an approach to integrate these chunks in a semi-automatic way.

The problem of integrating chunks is strongly related to the problem of *schema integration* and more in particular *view integration* [1] (chunks can be compared to views on the Business Information Model), but nevertheless there are some differences: (1) Chunks also allow to model functionality. In information system design, local schemas or views are not used to model functionality; (2) Constraints in a chunk are only valid for the audience class and the requirement under consideration. In local database schemas, constraints are defined to allow checking the integrity of the database and are valid for the entire domain modeled. Therefore, the integration techniques developed in the context of databases cannot be used as such.

2. WSDM'S INTEGRATION APPROACH

One of the fundamental problems in the integration of information is *semantic heterogeneity* [5]. Two objects with the same name can refer to different concepts in the domain; and two objects with different names can refer to the same concept. Semantic information is needed to detect and to solve these conflicts. We will use an ontology for this. From previous research in schema integration we know that the semantic information needed for the integration is mainly the different kind of relationships that can exist between the different concepts in the domain. Ontologies are defined as an explicit specification of a conceptualization [4]. They describe concepts in a domain as well as relationships between these concepts and the terminology used. Therefore, ontologies are well suited for a formal description of this type of semantic information. During the modeling of the different chunks the semantic information needed will be collected into an ontology. In principle, it is also possible to use an existing ontology for this purpose, but this is part of further research. Further on, the approach is based on the classical schema integration framework [1] consisting of the phases: *Pre-Integration, Schema Analysis, Schema merging* and *Restructuring.*

The structure of our ontology is as follows:

- **Object Concepts.** We distinguish between Lexical Concepts (e.g. 'person name') and Non-lexical Concepts (e.g. 'person'). This distinction helps to reduce the amount of information that needs to be specified by the designer (see further).
- **Relationships.** Relationships in the ontology express the relationships that may exist in the domain between object concepts (e.g. 'works-for'; the Object Concepts involved are 'person' and 'company').
- **Tuples**. A tuple is a grouping mechanism for concepts (e.g. ('first name', 'family name')).

WWW 2003, May 20-24, 2003, Budapest, Hungary.
ACM xxx.

Each concept in the ontology is uniquely identified by an *identifier* (ID); has a set of *labels* that are possible names (synonyms) for the concept (e.g. 'film', 'movie'), one label is the *preferred label* and should be unique (we will use in the examples the preferred label to refer to a concept); has a *comment* that is a short description or explanation (text) of the concept. Relationships also refer to the domain and the range of the relationship.

The ontology also contains *dependencies* that exist between the concepts. These dependencies can be considered as pre-defined relationships, i.e. their meaning is fixed.

- ***EquivalentTo***. Expresses that a concept is equal to another concept (e.g. *name* is equal to (*first name*, *family name)*.
- ***SubtypeOf***. Expresses that one concept is a subtype of another concept (its population is a subset of the population of the other). E.g. *Man* is a subtype of *Person*.
- ***OverlapWith***. Expresses that two concepts are partially overlapping (their populations have a nonempty intersection). E.g. *Student* and *Employee* are overlapping concepts.
- ***PartOf***. Expresses that one concept is a part of (component of) another one concept. E.g. *first name* is a part of *name*.

Part of the information in the ontology is entered by the designer(s) during conceptual design while making the chunks; the rest is derived from the information provided in the ontology using a set of rules (see further). The elements used in the chunks are linked to the concepts in the ontology. Therefore, every chunk element (object type, role, relationship, …) will refer to exactly one ontology concept. In this way, we can have two chunk elements with different names that refer to the same ontology concept. E.g. by using the same ontology concept for *EmployeeId* and *AdminNumber* we state that they are in fact the same concepts (*EmployeeId* and *AdminNumber* become synonyms). Similarly, we can have two different chunk elements with the same name that refer to different ontology concepts.

We will now sketch the different phases of the integration process. In traditional schema integration *pre-integration* is used to translate the local schemas into a common language. In our situation all chunks are modeled using the same language Therefore, we use the pre-integration phase to collect the necessary semantic information for the object types and roles introduced during modeling. The best moment to collect semantic information about something is when it is introduced. Therefore, if a new concept is used (by a designer) it is introduced in the ontology and possible dependencies with other concepts are identified. The designer can also provide a (text) comment explaining the meaning of the concept and the role it fulfils in the domain. Other designers can use these comments to quickly identify relevant concepts in the ontology and to investigate possible dependencies between concepts. We deliberately have incorporated the collecting of semantic information into the conceptual modeling process. Although this may slightly slow down the modeling process we believe that it is a better and less time consuming solution than collecting it afterward. If it has to be entered after the modeling is finished (like in classical integration approaches) more errors will be made and more time will be needed (we may not always remember the exact meaning of a concept). An additional advantage is that, in case multiple designers are involved, each designer can enter the semantic information for his own concepts and relate it directly to the concepts already entered by the other designers or reuse concepts from other designers. In addition, the semantic information already gathered can be used to make suggestions to the designers.

Also note that the designer does not have to enter semantic information for all concepts introduced. In fact, only semantic information for Lexical Concepts and Relationships need to be given. The semantic information for the Non-lexical concepts can be derived (using some rules). This is done during the *Schema Analysis* phase.

When conceptual modeling is finished, the ontology constructed defines all concepts and relationships used in the chunks. This information can then be used to construct the integrated schema. Also the constraints defined in the chunks need to be integrated into the Business Information Model. How this is done is omitted because of space restrictions.

The *Restructuring* phase has the same purpose as in the classic schema integration framework: enrichment, quality improvement and error correction.

3. ADVANTAGES

Some of the advantages of the approach are already mentioned: it is more effective to enter semantic information when it is introduced and the semantic information can be used to make suggestions to the designers. Another advantage is that the role of the domain expert can be limited considerable. In traditional integration solutions an overall domain expert has to provide all the semantic information during the integration process to solve any possible conflicts. This is a cumbersome and erroneous task. Here, the role of such an overall domain expert is minimized because each individual designer only needs to have knowledge about the part of the domain he or she is designing, and is only responsible for providing semantic information about this part. For large domains, this is a great benefit. In addition, less semantic information than usual need to be given by the designer because some can be derived. One more advantage is that by linking the concepts to an ontology during the conceptual design of the web site paves the way to the semantic web. In the semantic web, the information in a web site is annotated with ontology concepts. This allows exploring the knowledge available in the ontology when the web site is e.g. queried. In our approach the annotation with ontology concepts comes for free.

4. REFERENCES

[1] Batini C., Lenzerini M., Navathe S. B.: "A Comparative Analysis of Methodologies for Database Schema Integration", ACM Computing Surveys, 18(4): pages 323-364, 1986

[2] De Troyer O.: "Audience-driven web design", In Information modelling in the new millennium, Eds. Matt Rossi & Keng Siau, IDEA GroupPublishing, ISBN 1-878289-77-2, 2001

[3] De Troyer, O., Leune, C.: "WSDM: A User-Centered Design Method for Web Sites," In Computer Networks and ISDN systems, Proceedings of the 7th International World Wide Web Conference, Elsevier, pp. 85 – 94, 1998.

[4] Gruber T.: "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, 5(2): pages 199-220, 1993

[5] Hull R.: "Managing Semantic Heterogeneity in Databases: a Theoretical Prospective", In PODS'97, Proceedings of the 16th ACM SIGMOD Int. Conference in Management of Data, pages 51-61, New York USA, 1997