

Annotation for the Semantic Web During Website Development

Peter Plessers and Olga De Troyer

Vrije Universiteit Brussel, Department of Computer Science, WISE, Pleinlaan 2, 1050
Brussel, Belgium
{Peter.Plessers, Olga.DeTroyer}@vub.ac.be

Abstract. While introducing the HTML standard to present information on the World Wide Web, the importance of being able to express the deep structure and meaning of the information was neglected. This has led to some of the limitations of the current web (e.g. its restricted query possibilities). Work has started in the domain of the semantic web which tries to solve this problem by annotating web pages with semantic information. A crucial aspect to the success of the semantic web is that we have methods available to create, integrate and use this semantic information. In this paper, we present a new approach to generate semantic information by taking the annotation process to a conceptual level and by integrating it into an existing website design method.

1 Introduction

The large majority of current information available on the web is presented using the standard HTML format. Emphasis was put in this standard on layout possibilities but the importance of being able to express the meaning of the presented information was neglected. The lack of semantic information in current websites is addressed by the vision of the Semantic Web [1]. This vision states that the information available on the WWW should be defined such that it remains usable for human interpretation, but also becomes usable for machines. In this way, we can solve some of the limitations of the current web (e.g. its restricted query possibilities, intelligent agents, ...).

A crucial aspect to realize the vision of the Semantic Web in practice is that we have methods available to create, integrate and use semantic information and this, as much as possible, in a transparent and automatic way. As mentioned in [7], the generation of semantic markup should be a by-product of normal computer use. A step towards this goal has been taken in recent years by annotation approaches such as SHOE [6], MindSwap [4] and CREAM [5]. While such tools solve a number of issues like syntactic mistakes or inconsistencies with the used ontology, a number of fundamental problems still remain. The main reason for these problems is that current tools define a linkage between an ontology and the actual data of the website on an implementation level resulting in a strong weaving of semantics and implementation. We list some of the problems we encounter in current annotation approaches:

- Despite the introduction of supporting tools, the annotation process remains a very heavy and time consuming task. In addition, in most current approaches this proc-

ess is an additional activity and the ones that will benefit from the annotations are usually not the ones that should accomplish the job. Therefore, the motivation for performing the annotation process is low.

- It is usually assumed that the granularity of the concepts defined in the ontology matches exactly the granularity of the data on the website, although this assumption cannot be taken for granted. It must therefore be possible to define a link between semantically equivalent concepts but with a different level of granularity.
- Most of the supporting tools only allow annotating static websites, page by page on an implementation level. Even approaches that support the annotation of dynamic generated websites (by annotating the database) create a direct link between the implementation structure of the database (i.e. tables and columns for a relational database) and concepts in the ontology. For static web pages this has as consequence that the work done for one page needs to be repeated for similar structured web pages and that the maintenance of the metadata becomes a heavy task with a huge cost. Also note that for both static and dynamic websites, every time one changes the implementation of the website or database, even though nothing has changed to the semantics of the presented data, the defined linkage between the web pages or database and the ontologies can be affected.

In this paper we present initial ideas for annotating websites during their design. The presented approach tries to solve the problems mentioned earlier by elevating the annotation process to a conceptual level. It is also our belief that (whenever possible) the annotation is best done while designing the website, not after it is implemented. In this way we can take advantage of the information available during the website design process to ease and improve the annotation process. Therefore, we propose to integrate the annotation process into an existing website design method. Several website design methods have already been proposed in literature. We will use WSDM (Web Site Design Method) [2] in our approach as this method is well suited for our purpose as it proposes an explicit information-modeling step at a conceptual level.

2 Approach Overview

2.1 Architecture

Figure 1 gives an overview of the global architecture of our annotation approach. The different phases of WSDM that are relevant for our annotation approach are at the left: Task Modeling, Navigational Design, Page & Presentation Design, Database Design and finally the Implementation. Our approach is integrated into the original phases of the WSDM design method. A short overview of each step of the WSDM method, together with the enhancements (if any) we made for our annotation approach, is given below.

- *Mission Statement Specification*: Specifies the subject and goal of the website and declares the target audience. No enhancements are needed in this step.
- *Audience Modeling*: In this phase the different types of users are identified and classified into audience classes. For each audience class, the different requirements

and characterizations are formulated. Also in this step, nothing additional is needed.

- *Task Modeling*: A task model is defined for each requirement of each audience class. Each task defined in the task model is elaborated into elementary tasks. For each elementary task a data model (called 'object chunk') is created, which models the necessary information and/or functionality needed to fulfill the requirement of that elementary task. ORM (Object Role Modeling) is used as the representation language for the object chunks. For our purpose, we added an annotation process to the Task Modeling phase. This results in the creation of a linkage between the object types and roles of the different object chunks and the concepts of one or more ontologies. This annotation is called the *conceptual annotation* (arrow A in Figure 1) because it is performed on a conceptual level. In this way we define the semantic meaning of the object types and roles used in the object chunks. This conceptual annotation is performed for static as well as dynamic websites.
- *Navigational Design*: In this phase of WSDM the navigational structure of the website is described by defining components, connecting object chunks to those components and linking components to one another.
- *Page Design*: During Page Design, the components of the navigational structure and their associated object chunks are mapped onto a Page structure defining the pages that will be implemented for the website. We determine which object chunks will be placed on a certain page. Using this step as well as the previous one (the navigational design) we can identify which object chunks will be placed on a page. This is necessary to know for the actual implementation which annotations we have to add to a page.
- *Presentation Design*: For each page defined in the Page Design a page template is created defining the layout of the page. This layout is defined in an implementation independent way. To implement the actual web pages making use of a chosen implementation language (e.g. HTML, XML, ...), an instantiation of these page templates can be generated. For this, the templates are filled using the proper data to obtain the actual pages.
- *Data Design*: As explained in [3] we can derive an integrated conceptual schema from the object chunks made during Task Modeling. This integrated object schema is called the Business Information Model (BIM) and can be used as the basis for a database schema from which an underlying database can be created. The Data Design is only done when we deal with dynamically generated websites querying a database. For static web pages the data design step is omitted as the actual data will not originate from a database, but will be supplied by the designer during implementation. For our approach, we need to keep track of two mappings: 1) the mapping from the object types and relationships of the different object chunks to their correspondence in the integrated BIM (called *object chunk mapping*) (B in Figure 2); and 2) the mapping between the BIM, used as the conceptual database schema, and the actual implementation (called *database mapping*) (C in Figure 2). In this way we are able to determine the mapping between the queries specified at the (conceptual) level of the object chunks, and the actual database.
- *Implementation*: In this phase of WSDM the actual implementation of a website, based on the models created in the previous phases, is generated. To this step we added the generation of the actual annotation of the website (called the *page annotation*) (D in Figure 2). Here we have to distinguish between static websites and

dynamically generated websites. For static websites only the conceptual annotation is needed. For dynamic websites also the chunk integration and the database mapping have to be taken into consideration.

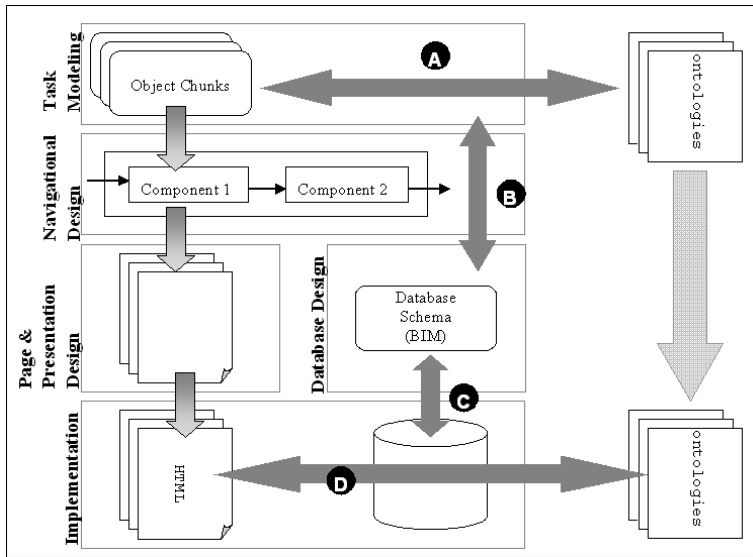


Fig. 1. Architectural overview

2.2 Advantages

The goal of our approach is to add semantic knowledge to the web pages of a new website. Opposed to current approaches, which perform the annotation on the web page level or on the database level (for dynamic websites), we define the annotation on a conceptual level. Web designers will provide the annotation during the conceptual design. Compared to currently existing annotation methods, this approach has a number of advantages:

- *The annotation is implementation independent.* Current methods define the annotations directly in the implementation of the website. Using our approach, an implementation will be generated (HTML, XML, ...) and changes can be generated without breaking the annotation, resulting in a greater level of maintainability of the annotation.
- *The annotation process is uniform for static and dynamic websites.* In current approaches the annotation for static and dynamic websites is done in a different way: respectively annotating web pages or a database. In our approach, the annotation step is done at the conceptual design which is independent on whether the website will be static or dynamic.
- *Reuse of the annotations.* In current annotation methods (for static websites), if a certain concept is used on different pages, the annotation has to be repeated for each page. In our approach, the annotation has to be defined only once and the same concept can be reused in different object chunks. Moreover, all copies of an

entity used over several Object Chunks will be updated automatically if the annotation of one copy has changed.

- *Improvement of the design process.* An important aspect of integrating the annotation into the design process is that it enables us to improve the consistency during this website design process and to speed it up by making use of the metadata already provided. It is for example possible to make suggestions to the designer about information to be included based on earlier conceptual annotations made.

3 Conclusion

In this paper, we presented an approach for the semi-automatic annotation of static as well as dynamic websites. The actual annotation process is performed during the design phase of the website. We presented the proposed approach integrated into an existing website design method, WSDM. This design method provides us a conceptual model of the website that can be used to annotate (at a type level) the information that will be available on the website, with concepts from an ontology. This is done by annotating the entities (Object Types and roles) used in the conceptual model of the website. Next, this “conceptual” annotation can be used to generate the actual page annotation by keeping track of the different transformations performed during the development process to derive an implementation.

References

1. Berners Lee, T., Hendler, J., Lassila, O.: The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* (2001) 5(1)
2. De Troyer, O., Leune, C.: WSDM: A User-Centered Design Method for Web Sites. *Computer Networks and ISDN Systems*, proceedings of the 7th International World Wide Web Conference, Brisbane Australia (1998) 85–94
3. De Troyer, O., Plessers, P., Casteleyn, S.: Solving Semantic Conflicts in Audience Driven Web Design. *Proceedings of the WWW/Internet 2003 Conference*, Algarve Portugal (2003)
4. Golbeck, J., Grove, M., Parsia, B., Kalyanpur, A., Hendler J.: New Tools for the Semantic Web. *Proceedings of EKAW 2002*, LNCS 2473. Springer (2002) 392–400
5. Handschuh, S., Staab, S., Maedche, A.: CREAM – Creating Relational Metadata with a Componentbased, Ontology Driven Framework. *Proceedings of K-Cap*, Victoria Canada (2001)
6. Heflin, J., Hendler, J.: Searching the web with SHOE. *Artificial Intelligence for Web Search*. Papers from the AAAI Workshop. WS-00-01, AAAI Press (2000) 35–40
7. Heflin, J., Hendler, J.: Agents and the Semantic Web. *IEEE Intelligent Systems Journal* 16(2) (2001) 30–37