



Vrije Universiteit Brussel

Vrije Universiteit Brussel
Faculty of Science
Department of Computer Science

Converting Web User Interfaces into Virtual Reality Interfaces using Semantic Annotations

A thesis submitted to the Vrije Universiteit Brussel in fulfillment of the requirements for Master of Applied Computer Science

By:

Ola H. Anan

Promoter:

Prof. Dr. Olga De Troyer

Supervisor:

Dr. Bram Pellens

Academic Year 2008-2009

Abstract

Virtual Reality (VR) applications are gaining more and more popularity on the internet in the last few years. Due to the rapid development in the hardware manufacturing and networking, users are nowadays enabled to possess an affordable personal computer with extremely powerful capabilities. In the initial evolution of the World Wide Web technology, users have been required to learn to perceive and interact with two-dimensional (2D) interface that looks nothing like the world they live in. With VR applications, users are enabled to visualize and interact with a computer generated world that looks feels and sounds more realistic and more appealing than the classical appearance of the 2D interfaces.

The development of VR applications on the web is still a specialized, time-consuming and expensive process. On one hand, several approaches have been developed to facilitate the process of designing and implementing VR applications. A new approach called VR-WISE (Virtual Reality – With Intuitive Specifications Enabled), developed at the WISE (Web and Information System Engineering) lab of the Vrije Universiteit Brussel (VUB) is an example of such approach. It aims to enable users with limited VR background to design and build Virtual Reality Environments using natural language specifications and mappings. On the other hand, some research is taking place on transforming classical websites into three-dimensional (3D) VR interfaces. Most of the existing approaches on this topic perform the transformation in a straight forward manner that lacks the semantics of the web site that is being transformed. The purpose of this thesis is to provide a more advanced approach that considers web site's semantics in the transformation process in light of several existing technologies.

This thesis proposes an approach of several steps to analyze the semantics of an existing 2D website, including presentation and content semantics, and then uses this gathered information as an input to the VR-WISE approach, and results in the generation of VR scenes that represents the transformed web site.

Acknowledgment

I would like to express my deepest gratitude to Prof. Olga De Troyer for giving me the chance of working on this topic and for her continuous support. A lot of appreciation to Dr. Bram Pellens for his helpful guidance and encouragement throughout this work.

I would like also to thank the Erasmus Mundus External Cooperation Window for making it possible for me to be part of this valuable experience of studying in the capital of Europe.

I thank my dear friends Karim Bayoumi, Ayat Ismail and Mukarram Abbas for being there for me at all times and circumstances. Also to the Mushtaha's, Abdalghani and Amani, for their continued support during my stay in Brussels.

Last but not least, I cannot thank enough my family and my parents, Prof. Haidar Anan and Mrs. Karima Anan and my beloved late brother Salim, for their unconditional love and nurturing throughout my lifetime.

Table of Contents

ABSTRACT	
ACKNOWLEDGMENT	
TABLE OF CONTENTS	
TABLE OF FIGURES	
CHAPTER 1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 OBJECTIVE	3
1.3 THESIS STRUCTURE	3
CHAPTER 2 BACKGROUND	5
2.1. SEMANTIC WEB	5
2.1.1. <i>Ontologies</i>	7
2.1.2. <i>Semantic Annotations</i>	8
2.2. ANNOTATION FRAMEWORKS	10
2.3. WEBSITE TRANSFORMATION APPROACHES	12
2.3.1. <i>Dante</i>	12
2.3.2. <i>SADie</i>	13
2.4. VIRTUAL REALITY	15
2.5. VR-WISE	15
CHAPTER 3 APPROACH	19
3.1. OVERVIEW OF THE APPROACH	19
3.2. IDENTIFYING ONTOLOGY	21
3.2.1. <i>Presentation Concepts</i>	21
3.2.2. <i>Content Concepts</i>	23
3.2.3. <i>Combined Ontology</i>	24
3.3. ADDING SEMANTIC ANNOTATIONS	24
3.3.1. <i>Manual Semantic Annotations</i>	24
3.4. COMBINED ONTOLOGY CONVERSION	25
3.4.1. <i>Protégé format vs. OntoWorld format</i>	25
3.5. USING VR-WISE	27
3.5.1. <i>Specification Phase</i>	27
3.5.2. <i>Mapping Phase</i>	28
3.5.3. <i>Generation Phase</i>	29
3.6. REFLECTION	29
CHAPTER 4 CASE STUDY	31
4.1. OVERVIEW	31
4.2. IDENTIFYING ONTOLOGY	31
4.2.1. <i>Defining WISE Presentation Concepts</i>	31
4.2.2. <i>Defining WISE Content Concepts</i>	33

4.2.3.	<i>Defining WISE Combined Ontology</i>	35
4.3.	ADDING SEMANTIC ANNOTATIONS TO WISE PAGES	35
4.4.	USING VR-WISE APPROACH	38
4.4.1.	<i>Implementing Specification Phase</i>	38
4.4.2.	<i>Implementing Mapping Phase</i>	38
4.4.3.	<i>Implementing Generation Phase and Viewing the Results</i>	39
4.5.	CONCLUSION	39
CHAPTER 5	CONCLUSIONS & FUTURE WORK	41
BIBLIOGRAPHY	43

Table of Figures

Figure 1-1: Virtual Seminar in Second Life	2
Figure 1-2: 3D version of the WISE website using Exit Reality	2
Figure 2-1: Semantic Web Layers	6
Figure 2-2: Screenshot of Protégé previewing the Movie Ontology.....	9
Figure 2-3: Architecture of CREAM	11
Figure 2-4: Screenshot of OntoMat	11
Figure 2-5: Overview of Dante Approach	14
Figure 2-6: SADle Ontology structure.....	14
Figure 2-7: Overview of VR-WISE approach	17
Figure 3-1: Overview of our approach	20
Figure 3-2: Combined Ontology Structure	22
Figure 3-3: CSS elements view of Amazon.com	22
Figure 3-4: Highlights on previous annotations of the home page of WISE lab	26
Figure 3-5: Phases of using OntoWorld	26
Figure 4-1: WISE CSS elements as viewed in Opera Browser	32
Figure 4-2: Overview of ISWC ontology as viewed in Protégé	34
Figure 4-3: Overview of the Combined Ontology.....	36

Chapter 1 Introduction

During the past few years, a lot of technological developments have taken place in the field of Virtual Reality (VR) and the World Wide Web. It seems very likely that the web interface is moving towards becoming a true Virtual World. A lot of approaches have been developed for designing and creating VR applications on the Web. Some web sites allow subscribed users to create their own three-dimensional (3D) space, as in *Second Life*¹, *Active Worlds*² and *GogoFrog*³. Most people would think of *Second Life* as being the most famous and successful 3D attempt. It is a huge 3D world that can be navigated by users using an avatar. They can also meet other users, set up their own 3D space, participate in various activities and so on. Figure 1-1 illustrates a view of a virtual seminar in *Second Life*.

Other web sites provide users with the ability to browse any classical web site in a fancy 3D environment, such as *Exit Reality*⁴, *Browse3D*⁵ and *SphereXPlorer*⁶. *Exit Reality*, in particular, is a 3D browser which works with a plug-in is a cross between *Second Life* and the current browser. The 3D browser can convert any 2D website into a 3D one. Parts of the 2D website are placed in a space where the user can walk through. Currently, only Internet Explorer and Firefox under Windows are supported. A test with the WISE lab web site for example, resulted in a practical empty Virtual Environment with a continuous scrolling panel that represent the main page of the website, as shown in Figure 1-2.

1.1 Motivation

As mentioned, a lot of 3D creation approaches are being developed, either by building a VR environment from scratch, or by transforming the existing web sites into a VR environment. The problem with these existing approaches, *Exit Reality* for instance, is the lack of semantics consideration while performing the transformation from two-dimensional (2D) web sites to 3D environments. In this thesis we investigate the possibility to perform this transformation in an advanced way, by including semantic annotations in the process.

¹ <http://secondlife.com/>

² <http://activeworlds.com/>

³ <http://gogofrog.com/>

⁴ <http://exitreality.com>

⁵ <http://browse3d.com/>

⁶ <http://spheresite.com/>



Figure 1-1: Virtual Seminar in Second Life



Figure 1-2: 3D version of the WISE website using Exit Reality

By including semantic annotations in the transformation process, the content of existing 2D websites can be given more meaning. Consequently, the transformation process of a 2D web site would result in a VR environment that considers the actual meanings behind the web site's content instead of relying on simple object (e.g. links, pictures .. etc.) to perform the transformation.

1.2 Objective

The work of this thesis has the following objectives:

1. To investigate the current state of the art in transforming existing 2D web sites into 3D VR web sites.
2. To advance the state of the art in this field by exploring the possibility of improving the transformation process by including semantic annotations.
3. To compare the transformation process from 2D websites into 3D environments with other web transformation approaches.
4. To take advantage of the VR-WISE approach to

The objectives of this thesis are (1) to investigate the current state of the art in transforming existing 2D websites into 3D VR websites and (2) to advance the state of the art in this domain by investigating how semantic annotations can improve this transformation.

1.3 Thesis Structure

This document is structured as follows

Chapter two elaborates the background concepts underlying this thesis, in addition to providing an overview to some related approaches in the field of web site transformation.

Chapter three provides a detailed illustration of the proposed approach for transforming a 2D web site into a VR environment using semantic annotations and by applying the VR-WISE approach.

Chapter four includes a demonstration of a case study to transform the WISE lab web site into a VR environment by performing the steps of the proposed approach.

Finally, Chapter five concludes our findings and elaborates the future work.

Chapter 2 Background

This chapter provides background on the core concepts underlying this thesis. It defines briefly the concepts of the Semantic Web. It will introduce the concept of ontology and then go into more detail on the activity of annotation. Furthermore, a number of approaches similar to ours, namely transforming websites to some other representation, are discussed. Finally, the chapter will give an overview on Virtual Reality and the VR-WISE approach used as the basis for this work.

2.1. Semantic Web

After several decades of the birth of the World Wide Web (the Web for short), the content of the Web is increasing massively day by day. It has grown to be so wide and huge to the point that it becomes difficult to deal with all this content manually. The content of the current Web is meant to be read and understood by humans. However, it is rapidly evolving into a content that could be meaningful to machines as well as people. This is the ideal concept behind the Semantic Web.

The idea of the Semantic Web was thought up by Tim Berners-Lee, the inventor of the most important technologies around the Web that we know nowadays, such as URLs, HTTP and HTML. The Semantic Web in short is a web with meaning. It could be defined as an extension of the existing Web in which information is given a well-defined meaning, so that it is understandable to both computers and humans (Berners-Lee, Hendler, & Lassila, 2001). The Semantic Web principles are implemented in the layers of Web technologies and standards, illustrated in Figure 2-1 (reproduced from (Koivunen & Miller, 2001)).

The Semantic Web Layers can be described as follows:

- The Unicode and Uniform Resource Identifier (URI) layers are to make sure that international characters sets are being used, as well as defining general means for identifying the Semantic Web objects, respectively. The most popular URI's on the World Wide Web are Uniform Resource Locators (URL's).
- The XML layer, in addition to namespaces (NS) and XMLschema definitions are used to insure the integration between the Semantic Web definitions and other XML-based standards.

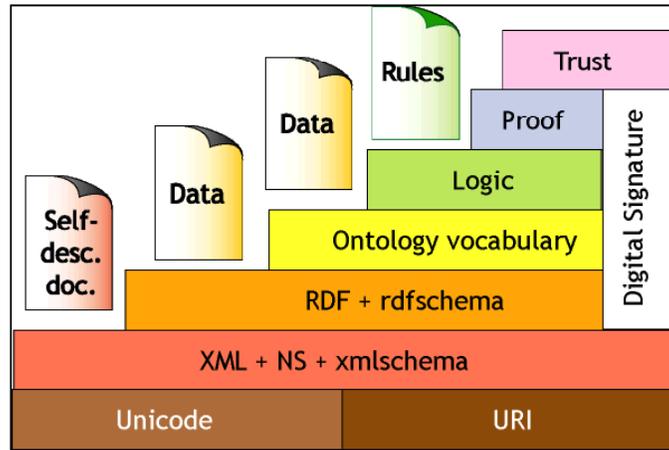


Figure 2-1: Semantic Web Layers

- The Resource Description Framework (RDF) and RDF-Schema are currently W3C standards. They are used to make statements about objects with URI's and to define vocabularies that can be referred to by URI's. In this layer, types are attributed to resources and links.
- Next, the Ontology layer, at which shared vocabularies are used to define concepts and relations between concepts of the various domains.
- A digital signature is basically the electronic identity of the sender of a message or the author of a document. The digital signature layer is to insure the security over the Semantic Web by checking whether the original content has been altered or not.
- The Logic layer enables the writing of rules while the Proof layer executes the rules and evaluates together with the Trust layer mechanism for applications whether to trust the given proof or not.

A lot of research is still taking place in various directions to improve, extend and standardize the Semantic Web, where many languages, publications, tools, etc. have already been developed and created.

2.1.1. Ontologies

One of the essential concepts underlying the Semantic Web is the concept of Ontology, and has gained much popularity over the last few years. In the English language, the word "Ontology" refers to a philosophical discipline which relates to the study of existence, the nature of reality and the common features of all beings. This term is being borrowed in Computer Science, where it is defined as a description of a specific domain. This description consists of a set of terms; their definitions and their inter-relationships. Usually a hierarchical structure is used to organize the terms, and there is an agreement between users of an ontology in such a way that the meaning of the terms is used consistently (Nieto, 2003).

In the Semantic Web, the term "Ontology" is defined as an abstraction of a computer-based lexicon, vocabulary, glossary, extended with knowledge about a given domain (Gruber, 1993; Guarino and Garetta, 1995 in Kleinermann et al., 2005). In our work, we tend to use the term "Domain Ontology" to refer to the set of concepts which are needed from the domain under consideration. In particular, building a Virtual World is based on the information contained in the domain ontology (e.g. the domain terminology is used to declare objects of the Virtual World being generated). For example, when one wants to build a Virtual Worlds representing chemical structures, this domain ontology would contain concepts such as "atoms", "proton", "electron" and so on.

Different Web Languages have been used to represent ontologies. One example of such languages is the DAML+OIL. DAML⁷ (DARPA Agent Mark up Language) which is an American project; has been developed as an extension to XML and the Resource Description Framework (RDF). On the other hand, OIL⁸ (Ontology Inference Layer) which is a European project, has been proposed as a web-based representation and inference layer for ontologies. It combines the widely used modelling primitives from frame-based languages with the formal semantics and reasoning services provided by description logics. Later on, both projects have been combined into the DAML+OWL ontology language. Another Ontology Language is OWL⁹ (Web Ontology Language), which is considered to be the descendant to the DAML+OIL. It is a W3C standard language, and is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics (OWL Web Ontology Language Overview).

Domain ontologies could be either extended from an existing ontology, or they could be created from scratch. Nowadays, there are several tools for editing and creating ontologies. An example is Protégé¹⁰, which is a free open source ontology editor and knowledge-base framework. Figure 2-2 illustrates Protégé preview on the tree structure of an ontology about movies.

2.1.2. Semantic Annotations

Another important feature of the Semantic Web is the concept of Semantic Annotation. Usually, when reading a book or an article, people tend to write their notes and comments as footnotes or on the pages' margins. These comments are called annotations. In Computer Science, an annotation is a set of remarks attached to a document (Handschuh, Staab, & Ciravegna, 2002)

Semantic annotations are added to specific resources on the Web (Web pages, pictures, multimedia content, etc.) in order to enrich it. Usually, annotations are added to page elements in order to give them descriptions that could later on be extracted and manipulated by other means to meet the needs of end users. For example, a group of links may be annotated as a menu, images may be annotated as advertisements and paragraphs could be annotated as the main content of a web page (Lunn, SADLe: Structural semantics for accessibility and device independence, 2005).

⁷ <http://www.daml.org/>

⁸ <http://www.ontoknowledge.org/oil/>

⁹ <http://www.w3.org/TR/owl-features/>

¹⁰ Developed at the Stanford University School of Medicine.

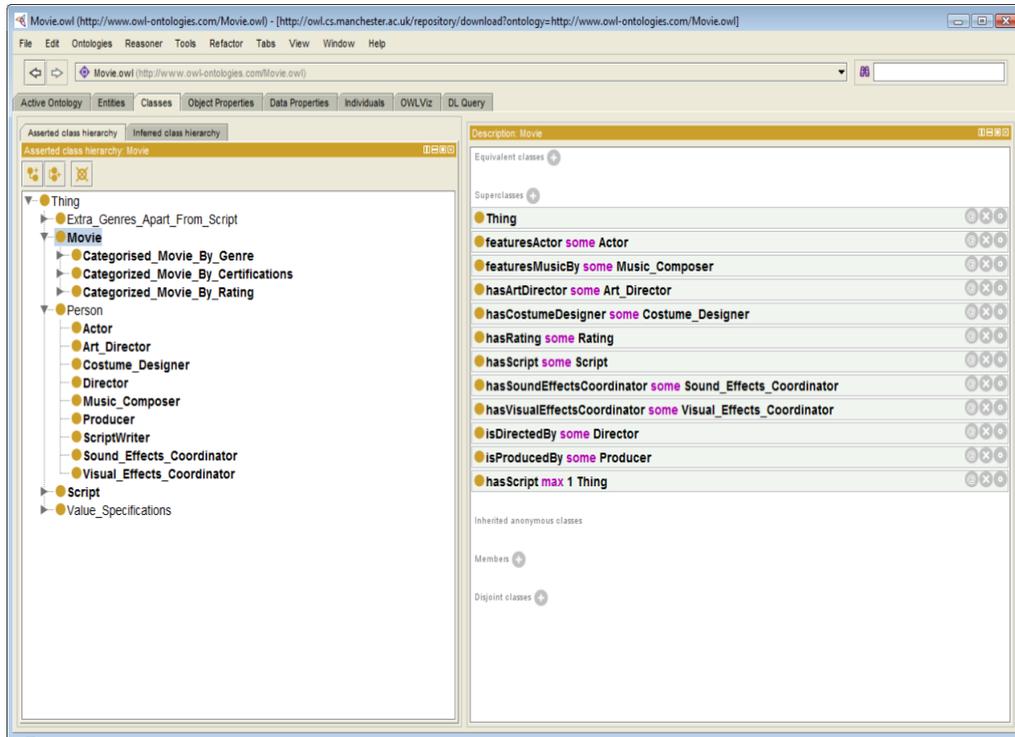


Figure 2-2: Screenshot of Protégé previewing the Movie Ontology

In the context of the Semantic Web, adding semantic annotations to documents enhance the ability to process the content of the Web by machines, and hence enable computers and people to work in cooperation (Kleineremann, De Troyer, Creelle, & Pellens, 2007).

The content of Web pages is annotated with the use of semantic information from domain ontologies using annotation tools. Several frameworks have been developed to produce semantic annotations that are based on a vocabulary presented by ontologies (Kashyap, Bussler, & Moran, 2008).

2.2. Annotation Frameworks

Domain Ontologies are used in the context of the Semantic Web for capturing semantic information from documents by annotating their content. The Semantic Annotation process results in embedded mark-up within the web pages. This mark-up is called Semantic annotations. They identify concepts and relations between concepts in documents. Furthermore, they are machine interpretable (Kashyap, Bussler, & Moran, 2008).

Several frameworks have been developed for creating Semantic annotations. A complete comparison between the various frameworks could be found in (Kashyap, Bussler, & Moran, 2008). One of the most well known frameworks is the framework called Dante. Dante has been used to support a number of domain applications, including the generation of semantic annotations for visually impaired users (Yesilada, Harper, Goble, & Stevens, 2004). This framework has been implemented in an annotation tool called COHSE, which is supposed to work as a plug-in in web browsers such as Mozilla Firefox and Internet Explorer. Nonetheless, this tool was no longer available for testing during this thesis work. Dante will be discussed in more details in Section 2.3.1.

Another annotation tool is SMORE, which allows users to mark-up HTML documents in OWL using Web Ontologies, it provides a user friendly way to create classes, properties, and instances from existing ontologies, and enables the creation of new ontologies using terms from web documents. This tool has been tested throughout this thesis work, but was a bit complicated and not friendly to use compared to the other framework: CREAM.

CREAM¹¹ (Creating RELational, Annotation-based Metadata) is a framework used for creating annotations with respect to the context of the pages being annotated. The architecture of CREAM, illustrated in Figure 2-3, consists of an Annotation Interface, a Document Management component, an Annotation Inference Server and an Information extraction and Re-recognition component.

¹¹ Developed at the University of Karlsruhe – Germany.

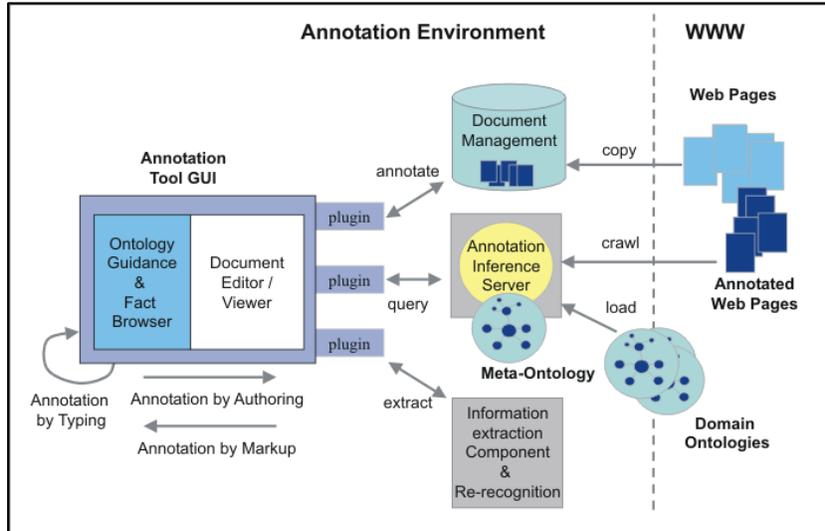


Figure 2-3: Architecture of CREAM

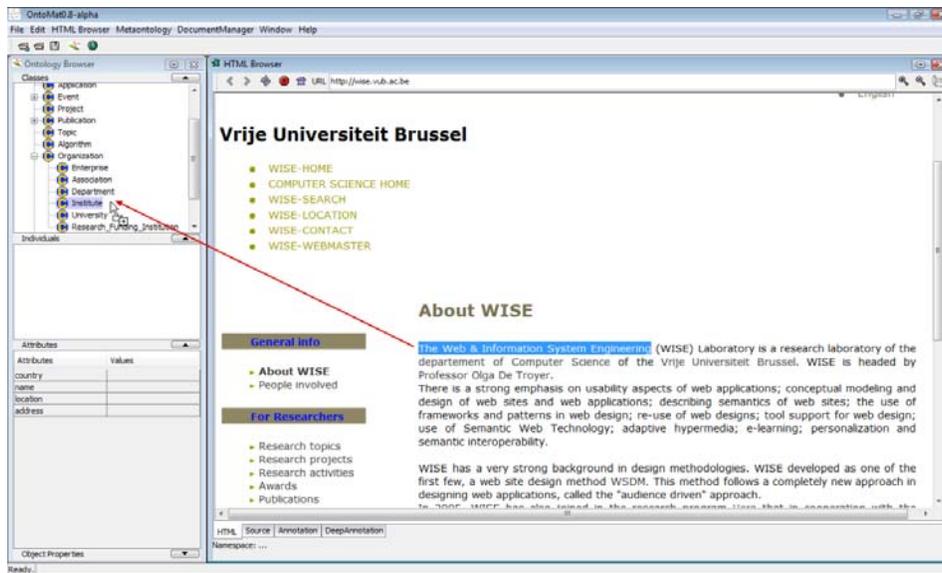


Figure 2-4: Screenshot of OntoMat

Users of CREAM are enabled to edit and annotate HTML documents, creating annotations which contain relationship instances of concepts presented in a certain domain Ontology. CREAM donates to the W3C standard formats with annotations made in RDF or OWL, while XPointers are used to locate the annotations in the text (Uren, et al.).

The OntoMat Annotizer is one example for annotation tools based on the principles of the CREAM framework. As shown in Figure 2-4, the OntoMat's interface consists of an HTML Browser (on the right side) to display the web page being annotated, in addition to an Ontology Browser for ontologies (on the left side). Users create semantic annotations by dragging and dropping text from the HTML Browser into the considered ontology concept appearing in the Ontology Browser.

In the current version of OntoMat, a straight forward file-system based document management approach is used for tracing annotations of a document. The OntoMat uses URI's to detect the re-encounter of previously annotated documents and highlights annotations in the old document for the user.

OntoMat has been extended to include support for semi-automatic annotation and deep annotation of databases. A newer version called M-OntoMat is meant to support manual annotation for multimedia documents such as images and videos (Kashyap, Bussler, & Moran, 2008).

2.3. Website Transformation Approaches

Recently, several approaches have been developed in the area of transforming websites so that it can be viewed on any of the increasing number of diverse devices. In particular, some approaches are concerned in transforming websites in the context of accessibility (e.g. transcoding a website into an appropriate format for screen readers). In this section we discuss two approaches for transcoding websites to allow accessibility for visually impaired users.

2.3.1. Dante

Dante is a web accessibility research project developed at the Human Centred Web Lab of the University of Manchester. Dante aims to overcome the lack of navigation and orientation of web pages using a normal screen reader, due to the fact that some implicit knowledge contained in web pages is not accessible by visually impaired users. It suggests transcoding web pages into a more suitable format based on its content. The approach proposes annotating components of a certain web page in order to identify their roles. For example, a set of links could be annotated as a navigation menu, an image as an advertisement and so on. By using annotations to mark the roles of elements and components of a web page, the page could be described in a way that assist the screen reader to navigate through the

content of the page easily (Lunn, SADle: Structural semantics for accessibility and device independence, 2005).

The research on Dante has been combined with several related projects: an Ontology called WAfA (Web Authoring for Accessibility), and a semi-automatic annotation tool called COHSE. On one hand, the WAfA ontology encapsulates knowledge about concepts related to how objects on a page are presented and used. On the other hand, COHSE uses the concepts of WAfA for annotating web pages components and elements. Dante then uses these annotations to transcode Web pages into alternative forms to be easily accessed by screen readers (Yesilada, Harper, Goble, & Stevens, 2004). Figure 2-5 shows an overview of the Dante Approach. Currently, the HCW Lab labels Dante as being a closed project, while the COHSE annotator components are no longer maintained.

2.3.2. SADle

SADle¹² is an acronym for Structural-Semantics for Accessibility and Device Independence. It was developed at the Human Centred Web Lab of the University of Manchester. Similar to Dante, it has been proposed to allow web accessibility for visual impaired users. SADle goal is to provide an approach for transcoding web sites with the minimum amount of manual annotation of web pages. SADle achieves this goal by annotating the CSS files (Cascading Style Sheets) underlying a number of web pages as an alternative of annotating the web pages itself. By doing this, the process of annotation becomes less time consuming; annotating a single CSS file instead of annotating every page in the web site, assuming that all the pages within the web site use the same CSS files. Those annotations can be used afterwards for transcoding the actual pages of the web site.

Besides defining the roles of the elements and the components in CSS files, SADle defines elements of a web page either to be a removable or a non-removable element based on its importance in the page. It also gives a number of priority levels to be attached to web page elements according to the order of their appearance in the resulted transcoded page, which would be an input for any device, e.g. a screen reader (Lunn, SADle: Structural semantics for accessibility and device independence, 2005).

As shown in Figure 2-6, SADle's ontology structure consists of two parts: an Upper Level generic Ontology, and a web site Specific Extension to the upper ontology. The upper ontology defines a set of high level concepts for classifying the CSS elements roles that may exist in any web site. While the web site Specific Extension contains the elements of the CSS underlying the specific web site to be transformed. These elements are annotated with the roles that they play from the terms found in the upper ontology (Lunn, Bechhofer, & Harper, 2008).

¹² <http://hwc.cs.manchester.ac.uk/research/sadie/>

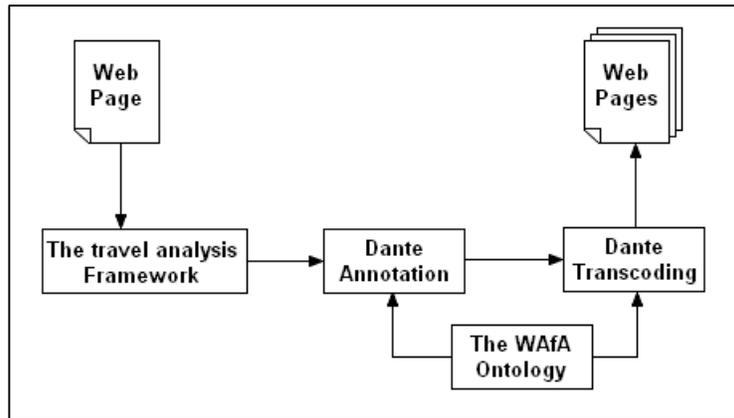


Figure 2-5: Overview of Dante Approach

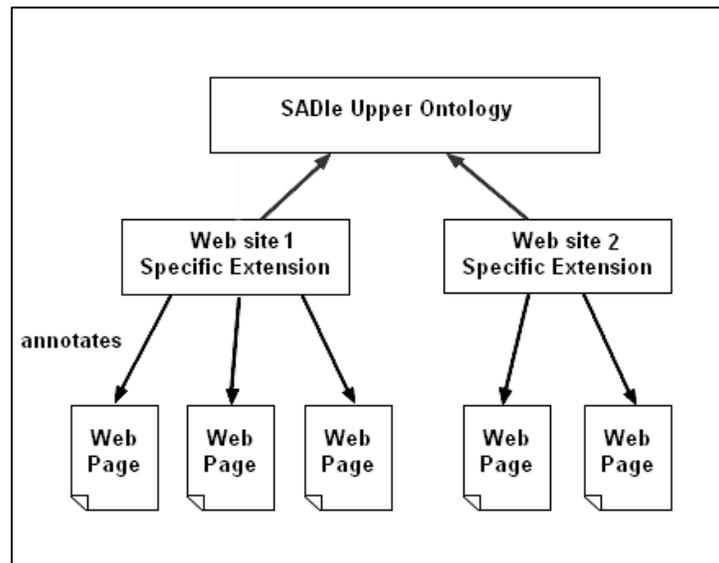


Figure 2-6: SADle Ontology structure

2.4. Virtual Reality

The term Virtual Reality (VR) is used for a technology that simulates three dimensional environments in which users can interact in real time with 3D objects that resembles real-world entities. Recently, a lot of applications and games are developed, turning the field of Virtual Reality into a popular topic for research and improvement, especially with the rapid evolution in the hardware of personal computers and network capabilities (De Troyer, Kleinermann, Pellens, & Bille, 2007).

A lot of approaches have been developed for designing and creating virtual reality applications. Nowadays, a lot of those approaches are available on the Web. Some websites allow subscribed users to create their own three dimensional space, as in SecondLife¹³, Active Worlds¹⁴ and GogoFrog¹⁵. Other websites provide users with the ability to browse any classical website in a fancy three dimensional environment, such as Browse3D¹⁶, SphereXPlorer¹⁷ and ExitReality¹⁸. The problem with those approaches is the lack of semantics while performing the transformation from 2D websites to 3D environments. In this thesis we investigate the possibility to perform this transformation in an advanced way, by including semantic annotations in the process. One example of using knowledge representation and semantics in building virtual reality applications is the VR-WISE approach, explained in the next section.

2.5. VR-WISE

The VR-WISE approach (Virtual Reality – With Intuitive Specifications Enabled) is a high-level model-based approach for developing virtual reality applications using semantics, developed at the WISE lab of the Vrije Universiteit Brussel. This approach facilitates the complicated task of developing virtual reality applications by using a domain-oriented approach. VR-WISE uses ontologies for describing the concepts of the domain for which we want to build a Virtual World.

VR-WISE simplifies the process of designing and creating virtual reality applications for non-expert users by requiring less technological background and thus making the virtual reality technology available to a broader public (Pellens, Kleinermann, De Troyer, & Bille, 2006).

¹³ <http://secondlife.com/>

¹⁴ <http://activeworlds.com/>

¹⁵ <http://gogofrog.com/>

¹⁶ <http://browse3d.com/>

¹⁷ <http://spheresite.com/>

¹⁸ <http://exitreality.com/>

The development process in VR-WISE (shown in Figure 2-7) is divided into three main phases, the Conceptual Specification phase, the Mapping phase and the Generation phase (De Troyer, Kleinermann, Pellens, & Bille, 2007).

At the Conceptual Specification phase, the designer specifies at a high level the Domain Ontology which captures the concepts needed for the Virtual World to be built, as well as specifying the possible relations that exist between those concepts. VR-WISE defines two levels of specification: Domain Specifications and World Specifications. The Domain Specification describes the concepts of the Virtual World, as well as adding possible relationships between those concepts, similar to creating classes and subclasses in the Object Oriented Modelling. In addition, concepts may have a list of properties or attributes which could later have a specific use in the Virtual World to be built (De Troyer, Kleinermann, Pellens, & Bille, 2007). The World Specification, on the other hand, defines the actual objects of the Virtual World. This means creating a list of instances of the Domain Specification concepts. These instances would later be used to populate the Virtual World. In OntoWorld, the user has to manually create an instance and link it to a specific concept, and it resembles, in our approach, the semantic annotation objects that are saved in the annotation files of the website. This could be enhanced by adding a feature to OntoWorld that would accept an XHTML file as an input, then detect the annotations and automatically link them to the list of concepts.

In addition to defining the Domain and World Specifications, the designer should specify the possible spatial relationships between objects of the Virtual World. A spatial relation specifies the position of an object relative to some other object in terms of a direction and a distance (De Troyer, Kleinermann, Pellens, & Bille, 2007). Spatial relations can be used in the domain specification as well as in the world specification. In the domain specification, the spatial relations are used between concepts and specify default positions for the instances of a concept.

Using OntoWorld we could, for example, state that an object A is positioned "Above" object B with a distance of 20 meters. Other spatial relations could be specified for an object, such as its position by coordinates or its orientation with respect to another object.

By defining the Domain Ontology and the World Specification, a Virtual World can be described at a conceptual level. However, this description is abstracted from the actual representation in a 3D environment. In order to make this link, a mapping is required. Afterwards, at the Mapping phase, the designer specifies a set of mappings between the concepts from the domain Ontology into the actual Virtual Reality objects. Similar to the Specification Phase, the Mapping phase is also divided into Domain Mappings and World Mappings. Domain Mappings is basically the mapping between a domain concept and a 3D object or model.

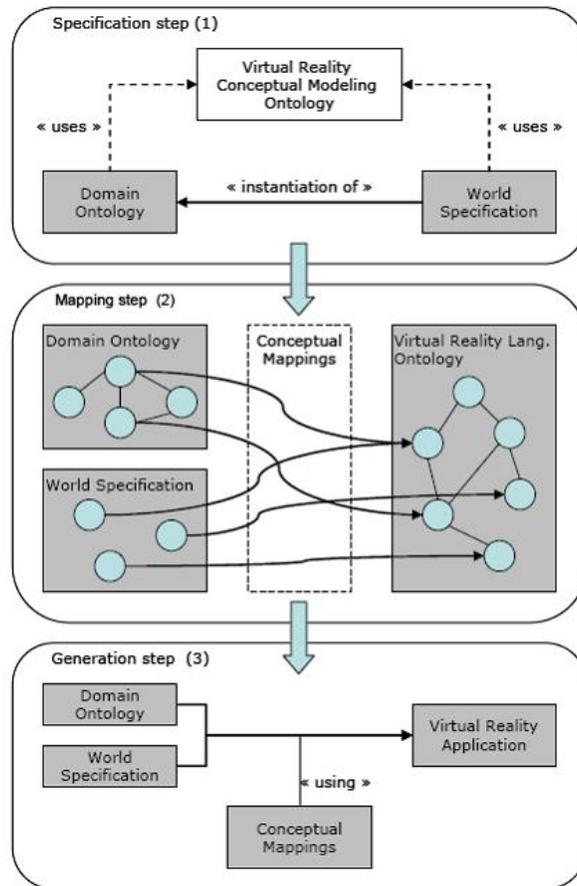


Figure 2-7: Overview of VR-WISE approach

By doing this, the designer specifies defaults for the visualization of the domain concepts in the Virtual World. For example, all instances of class A would be represented as a cube in the Virtual World. If the user is not satisfied with the default mappings between a concept and a 3D object, he could specialise the representation of a specific instance to be mapped into another 3D object. For example, all instances of class A would be represented as a cube, instead instance B, which would be represented as a sphere.

The Mapping Phase enables the transformation into the implementation level at the Generation phase, where VR-WISE uses the information gathered from the previous two phases in generating the source code for building the Virtual World (De Troyer, Kleinermann, Pellens, & Bille, 2007).

The development of a Virtual World according to the VR-WISE approach is supported by a toolbox called OntoWorld¹⁹, where the phases of VR-WISE are implemented in a user-friendly interface.

¹⁹ Developed at the WISE Research Lab of the Vrije Universiteit Brussel (VUB).

Chapter 3 Approach

The process of designing and building a virtual world, either manually or (semi-) automatically can be a very complicated task. Many approaches have been developed lately in order to simplify this task, and to enable non-expert users to create their own virtual worlds based on their domain knowledge and perspective.

In the previous chapter, we have pointed out that the VR-WISE approach enables designers to generate a Virtual World using a formal description of the domain under consideration. Thus, specifying the Virtual World in terms of concepts and objects from the problem domain without the need for special programming skills or technical knowledge for working with existing Virtual Reality editing tools. In this chapter we discuss the fundamental stages for an approach that relies on the VR-WISE along with other technologies discussed in previous chapter to transform a classical website into a Virtual World.

3.1. Overview of the Approach

In section 2.3, we have introduced several approaches that enable novice users to create a virtual reality environment easily and without requiring any programming skills (e.g. Second Life). Other approaches available on the web perform a straight-forward transformation from a classical website to a virtual world (e.g. Exit Reality). This thesis provides a better approach by including semantic annotations in the transformation process. First an Ontology is defined in order to describe the terminology of the website to be transformed, including both presentation and content concepts. Afterwards, the resulting Combined Ontology is used to add semantic annotations to the web site pages. Subsequently, we use the Combined Ontology and the annotated pages as an input to the toolbox of the VR-WISE: OntoWorld. Then OntoWorld is used to specify the visual mappings between concepts and virtual reality objects. As a result, OntoWorld generates the source code of a virtual world representing the transformed web site. Figure 3-1 illustrates an overview of the proposed approach.

In the following sections, we go into more details on each of the steps in order to complete the transformation process from 2D web site into 3D environment. For each step, an explanation is provided, including how to perform it in our approach. A reflection is given for each step discussing how it can be improved and other considerations to take into account to further automatise the transformation process.

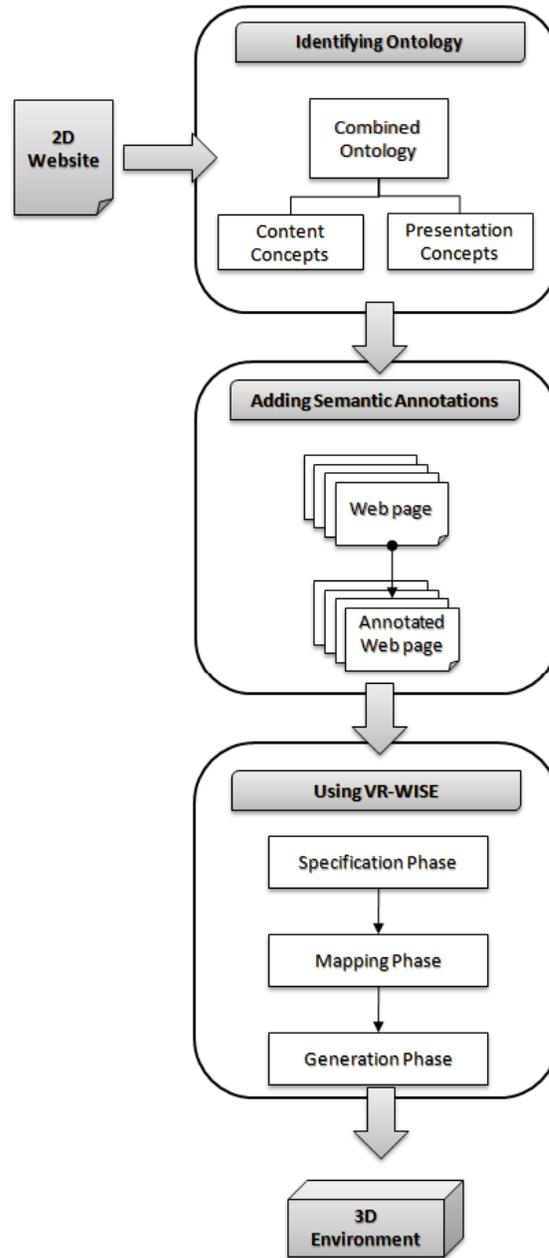


Figure 3-1: Overview of our approach

3.2. Identifying Ontology

As described in section 2.1, Ontologies are used to capture concepts and terminology of a particular domain of interest. In our approach, an Ontology is needed to capture the concepts for describing and/or annotating websites. Since a website typically contains both visual information as well as content information, both the presentation structure of a website and the knowledge held within it needs to be captured. In order to do this, we propose identifying an Ontology with two parts: one to describe the presentation concepts, and the other to describe the content concepts, as shown in Figure 3-2.

The top level concept of the ontology is called “Thing”. Two subclasses of Thing are defined, namely PresentationConcept and ContentConcept, to respectively describe the visual information and the content information. The main modeling concepts are defined as subclasses of either one of these two classes. They are described below.

3.2.1. Presentation Concepts

In order to identify the presentation concepts of a website, we extended the principles of the SADle approach, which we have discussed in section 2.2.2. SADle uses a general upper level ontology containing high level abstract concepts representing the potential roles of web page elements. In particular, the upper ontology contains classification classes for determining the type of web elements (e.g. MainMenu, SubMenu, etc), in addition to priority classes for indicating the importance of a web element by determining a value of priority (e.g. HighPriority, MediumPriority, LowPriority). The benefit of using an upper ontology is for it to act as an interface between SADle and the web page to be transcoded. A complete view on SADle structure and design could be found in (Lunn, 2005).

For a specific website, an extension of the upper ontology has to be created by analysing the CSS of the website we wish to adapt. The simplest way to classify the CSS elements in a website specific extension is to create direct subclasses of the classes in the upper level ontology. We can consider this extension to be an annotation of the style sheet elements – it provides information to indicate, for example, the type and the importance of a particular web element in the specific website under consideration (Lunn, Bechhofer, & Harper, 2008). This reduces the overhead required for annotation as each website is assumed to have a single CSS source to annotate. For example, assuming that web developers give meaningful identification names to CSS elements, when considering a website in the domain of e-commerce, the extension to SADle’s Upper Ontology would contain CSS elements related to commercial terms such as ‘product’, ‘provider’, ‘advertisement’ and so on.

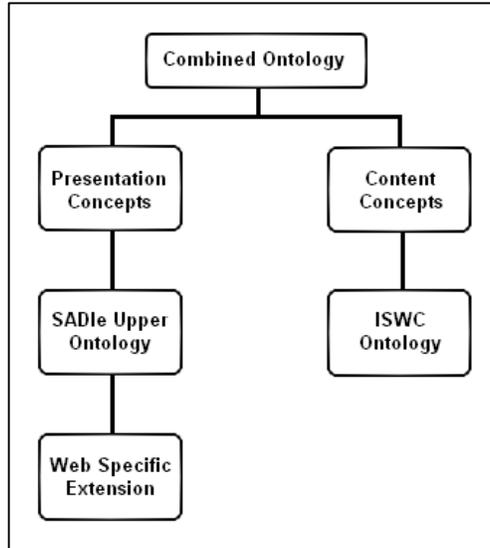


Figure 3-2: Combined Ontology Structure



Figure 3-3: CSS elements view of Amazon.com

3.2.1.1. CSS classes Inspector

In practice, we need to use a certain tool for inspecting the CSS classes in a web page in order to clearly identify the classes that are used by the web page and to see what their function and role is within the website. Such a tool is available within the Opera Web browser²⁰, which includes a built-in functionality to view CSS elements within a web page. Figure 3-3 shows the CSS elements of the homepage of Amazon²¹, a popular online shopping store for books and electronics. The box model was added as an overlay to the website by viewing the Class and Id attributes within the CSS of this website (Lunn, 2008).

Using the Class and Id values for inspecting the CSS elements is beneficial, because normally web designers use those attributes for determining the layout of web pages, the thing that helps capturing the semantics of the web page implicitly. Class attributes allow us to build up a semantic representation of the document and its structure. The values that determine how objects are displayed can also tell us what the objects are.

3.2.1.2. Ontology Editor

Another tool is needed for the creation and manipulation of ontologies. An example of such a tool is the Protégé Ontology Editor, which we have briefly mentioned in section 2.1.1. Protégé is a java-based, open source ontology editor. It creates ontologies in the Web Ontology Language (OWL) format. In this part Protégé is used to import the SADle upper ontology, and then extend it to describe the roles of the CSS elements that have been inspected for the website under consideration. A detailed report on implementing the SADle approach using Protégé can be found in (Lunn, 2008).

Up to this point, we have created the Presentation Concepts part of the combined Ontology by implementing the SADle principles. The next step is to determine the Domain Ontology that we will use to represent the concepts and the terminology used in the web site to be transcoded.

3.2.2. Content Concepts

A Domain Ontology describes a set of terms existing in that domain, in addition to the inter-relationships between those terms among each other. Domain ontologies could be either re-used from an existing ontology, or created from scratch. For example, a web site in the medical domain will contain concepts that reflect medication prescriptions, diseases, medical departments and so on. If we consider re-using an existing ontology, it is advisable to search ontology repositories, such as DAML Ontology Library²², TONES²³ and SchemaWeb²⁴. Such repositories could be

²⁰ <http://www.opera.com/>

²¹ <http://www.amazon.com/>

²² <http://www.daml.org/ontologies/>

browsed according to a certain keyword, class, property or URI. Other similar portals for browsing ontologies could also be used. In both cases, the user of an ontology will need to edit the concepts or the properties of the concepts using an Ontology Editor, e.g. Protégé, for editing concepts' names and properties whenever is needed.

3.2.3. Combined Ontology

As a result of the above steps, one Combined Ontology consisting of the two parts would be created: an Ontology that contains both the Presentation Concepts and the Content Concepts of the website under consideration. This ontology as a whole is used to semantically annotate the web pages that we wish to transform into a Virtual World.

3.3. Adding Semantic Annotations

An annotation is a remark attached to a portion of an HTML document. Semantics annotations are based on a certain ontology in order to describe the elements within that document. Several frameworks are available to facilitate the process of adding semantic annotations to HTML documents, such as the CREAM framework described in section 2.1.3. In this section we describe how CREAM is used in our approach by using its supporting implementation: OntoMat, also mentioned previously in section 2.1.3.

3.3.1. Manual Semantic Annotations

OntoMat is an annotation tool built as an implementation to the principles of the CREAM approach. In our approach, OntoMat is used to add semantic annotations to the web pages we are considering to transform into a virtual world. These annotations are based on the terminology of the Combined Ontology which has been created previously. This process results in a group of HTML documents, embedded within them a set of instantiations of the concepts (i.e. classes) in the Combined Ontology.

OntoMat provides the user with a web browser that displays the web page being annotated. On the other hand, it enables the creation of semantic annotations in a user friendly manner, such as using drag-and-drop for indicating that a piece of text is an instance of a specific class of the Combined Ontology.

In practice, OntoMat saves the annotations internally in the XHTML file in RDF/OWL format to represent class instances, in addition to using XPointer objects to show where each annotation resides in the base document. Actually, XPointer is an

²³ <http://owl.cs.manchester.ac.uk/repository/browser>

²⁴ <http://www.schemaweb.info/schema/BrowseSchema.aspx>

XML-based language which can be used for a fragment identifier for any URI reference that locates a resource. It can additionally locate points and ranges inside the XHTML documents.

Another important feature of OntoMat is the ability to track saved annotations of a document by detecting the URI's and highlighting annotations in the previously saved document for the user. Figure 3-4 shows how OntoMat underlines stored annotations in an XHTML document.

So far, we have semantically annotated the web pages using a Combined Ontology. Afterwards, we use the concepts of the VR-WISE approach to create a virtual world of the website we want to transform. The VR-WISE approach is implemented by a toolbox called OntoWorld. VR-WISE and OntoWorld have been briefly elaborated in section 2.4.

3.4. Combined Ontology Conversion

3.4.1. Protégé format vs. OntoWorld format

The previous step of identifying the Combined Ontology has produced an output of an ontology file in OWL/RDF extension according to Protégé's format settings. This output file would subsequently be used as an input for the VR-WISE's toolbox OntoWorld, which requires a specific XML OntoWorld format. For example, the movie ontology (shown previously in Figure 4) includes classes named "Person", "Actor" and "Director". Protégé would save those classes of the movie ontology in the following format:

```
<owl:Class rdf:about="#Person"/>
<owl:Class rdf:about="#Actor">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#Director">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
```

The above code defines a super class "Person" then defines two subclasses "Actor" and "Director". In order to define those classes and their relationships in an OntoWorld-compatible format, an XML file should be rewritten similar to the following format:

```
<ConceptualGraph cls="OntoWorld.Core.Graph.ConceptualGraph">
  <Nodes>
    <ConceptNode name="Person" group=""
      cls="OntoWorld.Core.Statics.ConceptNode">
    <ConceptNode name="Actor" group=""
      cls="OntoWorld.Core.Statics.ConceptNode">
```

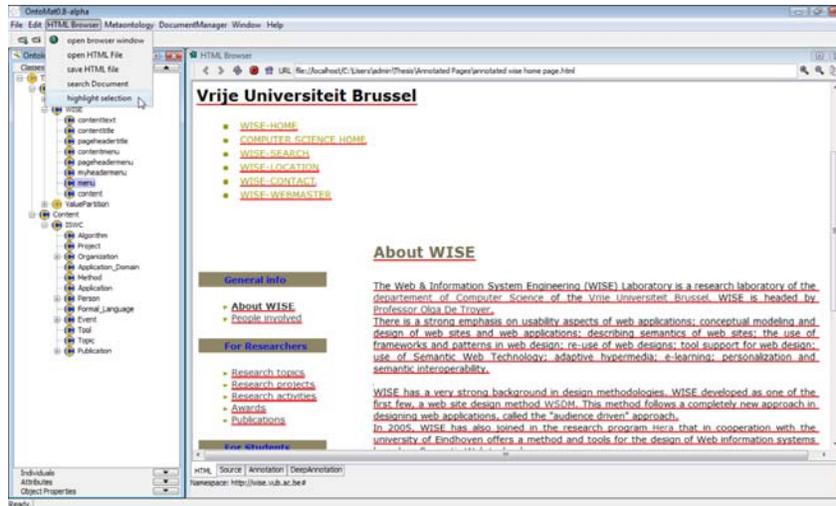


Figure 3-4: Highlights on previous annotations of the home page of WISE lab

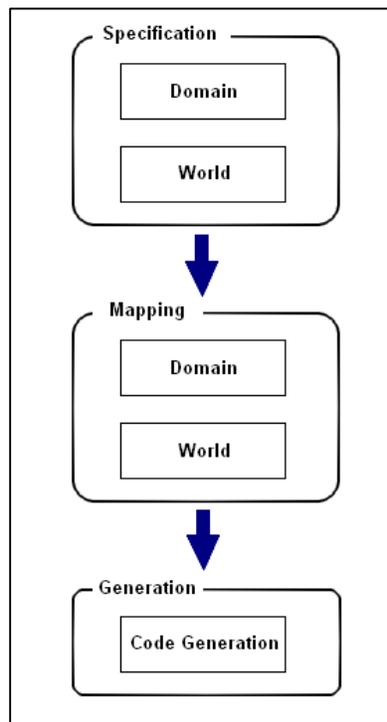


Figure 3-5: Phases of using OntoWorld

```

    <ConceptNode name="Director" group=""
      cls="OntoWorld.Core.Statics.ConceptNode">
  </Nodes>
  <Edges>
    <IsARelationEdge cls="OntoWorld.Core.Statics.IsARelationEdge">
      <Source>Actor</Source>
      <Target>Person</Target>
    </IsARelationEdge>
    <IsARelationEdge cls="OntoWorld.Core.Statics.IsARelationEdge">
      <Source>Director</Source>
      <Target>Person</Target>
    </IsARelationEdge>
  </Edges>
</ConceptualGraph>

```

3.5. Using VR-WISE

VR-WISE approach is a high-level model-based approach for developing virtual reality applications using formal specifications of a specific domain. A brief description of VR-WISE approach has been introduced in section 2.4. In our approach, VR-WISE is used for producing a virtual world of a specific website after creating a combined Ontology of the website, adding semantic annotations to its pages and transforming the output into a proper input for the VR-WISE approach. OntoWorld is a toolbox for implementing the principles of VR-WISE. Working with OntoWorld goes through several phases, as shown in Figure 3-5, where the user specifies a list of configurations to the virtual world to be generated. In this section we describe what should be done in each of these phases in the context of transforming a website.

In practice, going through the three phases is not in one direction as the arrows would implicitly indicate. However, the process is iterative, as the user can begin by generating an initial sample of the virtual world objects and then repeat the process until the generated world is complete and satisfying.

3.5.1. Specification Phase

In this phase, the user adds the concepts of the domain of the web site under consideration, while specifying their inter-relationships. We have discussed previously that the input in this phase is the list of concepts that have been created in the Combined Ontology structure. However, some conversion is required in order to change the RDF/OWL file format into a different format which is compatible with OntoWorld. In any case, the converted XML file would be treated as the Domain Specifications for the 2D web site under consideration.

3.5.1.1. Domain Specification

VR-WISE uses the term Domain Specification to elaborate the concepts of the Virtual World, as well as adding possible relationships between those concepts, in addition to their properties that could later have a specific use in the Virtual World that is going to be built using VR-WISE (De Troyer, Kleinermann, Pellens, & Bille,

2007). In OntoWorld, the user is required to add a list of concepts one by one or by creating an XML file that includes class definitions. In our approach, we have used the Combined Ontology concepts to reflect the Domain Specification classes.

3.5.1.2. World Specification

The world specification defines the actual objects of the Virtual World. This means creating a list of instances of the Domain Specification concepts. These instances would later be used to populate the Virtual World. In OntoWorld, the user has to manually create an instance and link it to a specific concept, and it resembles, in our approach, the semantic annotation objects that are saved in the annotation files of the website. This could be enhanced by adding a feature to OntoWorld that would accept an XHTML file as an input, then detect the annotations and automatically link them to the list of concepts.

3.5.1.3. Spatial Relations

In the previous steps, we have created a formal representation of the Virtual World's concepts and instances. It is essential at this point to specify some visualization properties to our virtual objects. In particular, we could begin by specifying the spatial relations between our objects.

In practice, spatial relations are added manually to each of the Virtual World objects; however, it is possible to automise this process by adding a script that could extract the XPointer objects from the annotated XHTML files. This would initially generate the Virtual World objects with the same layout of a certain web page, (e.g. the position of an XPointer object on a web page would be taken as the initial position for that object), but of course the user would be allowed to edit those initial values as needed.

3.5.2. Mapping Phase

At the Mapping Phase, the user specifies the visualisation properties of Virtual World Concepts. In particular, the user specifies what 3D object represents a formal concept or instance at the Specification level. This phase is also divided into two steps.

3.5.2.1. Domain Mappings

At this step, the user maps between a domain concept and a 3D object or model. In OntoWorld, the user is enabled to choose from an existing library of 3D objects. By doing this, the designer specifies defaults for the visualization of the domain concepts in the Virtual World. For example, all instances of class A would be represented as a cube in the Virtual World.

3.5.2.2. World Mappings

The user could settle for the default mappings between a concept and a 3D object, or he could specialise the representation of a specific instance to be mapped

into another 3D object. For example, all instances of class A would be represented as a cube, instead instance B, which would be represented as a sphere.

3.5.3. Generation Phase

During this step the actual source code for the virtual world is generated. This means that the conceptual specifications are converted into a working application by means of the mappings given during the mapping phase. In principle, different VR implementation languages can be supported. The current tool supporting VR-WISE allows generating X3D. Further details on this phase is found in (Pellens, 2007)

3.6. Reflection

In this chapter, we have provided an overview of the proposed approach for transforming a 2D web site into a VR three dimensional interface. We begin by defining a Combined Ontology that includes presentation and content concepts underlying the specified web site. Then we used this ontology to semantically annotate the web site pages. The result of those two stages would be used as an input to the VR-WISE approach. In the meantime, there is a gap between those results and VR-WISE. However, some additional features could be included in later versions of OntoWorld in order to minimize the manual operations that have been done throughout this thesis work.

For example, an improvement to OntoWorld would be to develop a script that can automatically perform the conversion process between Protégé and OntoWorld. This script would take an RDF/OWL file as an input and result in an XML file which is combatable with OntoWorld, and consider this output as the list of Domain Specification concepts.

Another improvement is needed to translate semantic annotation objects into World Specification instances. As mentioned previously, the semantic annotations are saved in RDF/OWL format to represent class instances, in addition to XPointer objects to show where each annotation is located in the base document. Those XPointer objects could be automatically extracted by OntoWorld to represent the spatial relations between the VR objects inside the virtual world. There must be also some way, for example to translate paragraphs into 3D text and pictures into 3D banners and so on.

Chapter 4 Case Study

4.1. Overview

This chapter applies the proposed approach described in the previous chapter to transform a classical 2D website into a 3D Virtual Reality environment using semantic annotations. The case study uses the website of the WISE lab as a starting point through the various steps into getting a three dimensional interface representing the domain on the WISE lab.

The transformation process of the WISE lab follows the steps of the proposed approach described earlier:

1. Identifying Ontology
 - a. Defining WISE Presentation Concepts
 - b. Defining WISE Content Concepts
 - c. Defining WISE Combined Ontology
2. Adding Semantic Annotations to WISE pages
3. Using VR-WISE approach
 - a. Implementing Specification Phase
 - b. Implementing Mapping Phase
 - c. Implementing Generation Phase
4. Viewing Results

4.2. Identifying Ontology

4.2.1. Defining WISE Presentation Concepts

In this step, we are concerned with identifying the presentation concepts of the WISE lab. We do this by implementing the SADle approach by using the Upper Ontology of SADle and create a WISE specific extension. In the technical report of SADle, the authors state the implementation steps as follows:

1. Inspecting the CSS elements of the WISE Website

In order to inspect the presentation concepts underlying the WISE lab, a CSS Inspector tool is used. There are several tool options for inspecting CSS elements in a web page. However, in our case study we use the built-in viewing capabilities of the Opera Browser. In a new window of Opera browser, navigate to the View menu Style options, and select Class and Id attributes and Outline options. Meanwhile, we navigate the browse to the WISE home page, the page is viewed as in Figure 4-1.

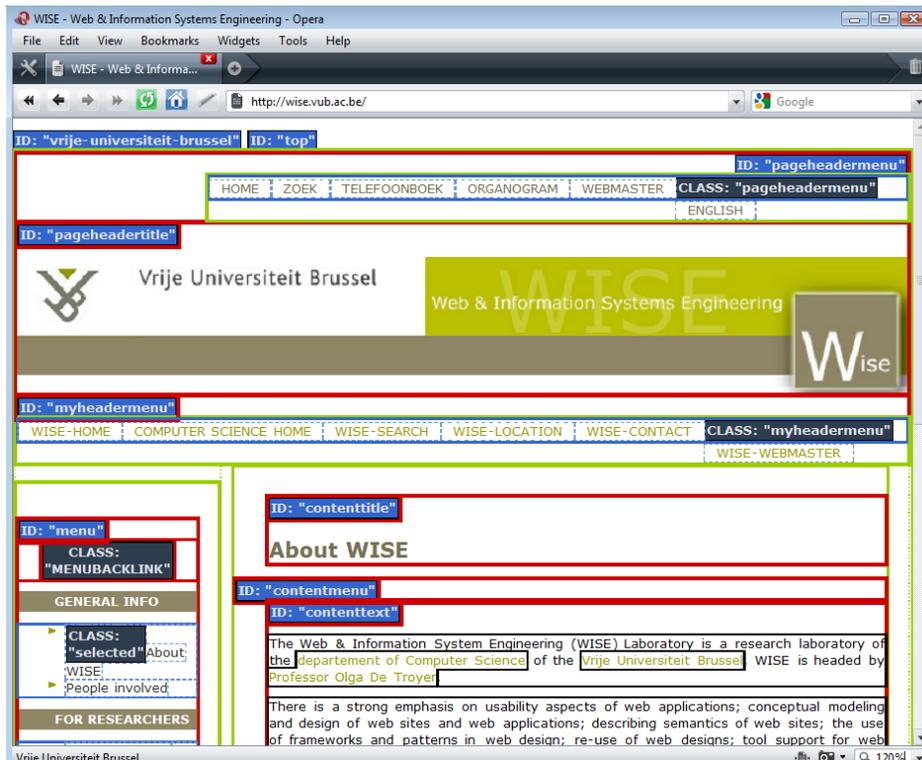


Figure 4-1: WISE CSS elements as viewed in Opera Browser

By viewing Class and Id Attributes, we were able to detect the following CSS elements in the home page of WISE, as well as the other pages of WISE website:

- pageheadermenu: this includes the top menu which appears in every web page that belongs to the Vrije Universiteit Brussel in general. It appears in the top right corner of each web page.
- pageheadertitle: this includes the WISE banner picture, which appears in every page that belongs to the WISE lab.
- myheadermenu: this CSS element borders the main menu of the WISE lab website. It appears below the banner on every page of the WISE website.
- menu: this element contains the side menu of the WISE website. This menu is also constant in every page of the WISE website, and is divided into subsections for categorization.
- contenttitle: this is the title of the current page. Normally this element has a distinct style for more attention.
- contenttext: this element includes the actual content of the page. In the home page of WISE it contains a descriptive paragraph about the WISE lab.
- contact: this element appears in the footer of WISE web pages and contains the contact information of the Vrije Universiteit Brussel.

We consider those elements to be the WISE extension to the SADIE Upper Ontology.

2. Extending the SADle Upper Ontology

Before we create the WISE Website ontology, we need to extend the SADle Upper Ontology to provide us with the properties and roles that we will add to the elements of the WISE Website. We do this by using the Protégé Ontology Editor to create a Combined Ontology based on the SADle Upper Ontology. We have to input some technical details for the new ontology, such as a unique URI, a physical location to store the ontology file, and a name for the ontology. Next, we need to import the SADle Upper Ontology using its URI²⁵. Subsequently, we define the roles of the WISE CSS Elements that we have inspected previously on the Opera Browser.

3. Defining the WISE CSS Roles

We do this in order to link each CSS element of the WISE website with an element of the SADle Upper Ontology. In other words, we are going to annotate the CSS elements of WISE with the concepts of the SADle Upper Ontology. First of all, we create a container class as a subclass of the “Thing” object, into which we will add all the CSS elements. Next, we have to identify the role of each CSS element of the WISE website. In particular, we could link each CSS element to a certain level of priority, according to the Element’s importance on the website. This information could be used later in the transformation process as a reference to the importance of a certain web element on the WISE website.

4.2.2. Defining WISE Content Concepts

In this step, we are creating the Content Concepts of the WISE website. This is achieved by either create a new Ontology or use an existing ontology that describes the domain concepts underlying the WISE website. In particular, we should use an ontology that describes the domain of universities and education. We could use some existing ontology repositories for finding an existing ontology to describe our domain instead of having to create one from scratch. In our case, we use the ISWC²⁶ ontology that concerns the domain of educational institutions. An overview of the ISWC ontology is provided in Figure 4-2.

²⁵ Located at: <http://hcw.cs.manchester.ac.uk/experiments/sadie/ontologies/SADle.owl>

²⁶ Located at <http://annotation.semanticweb.org/iswc/iswc.owl>

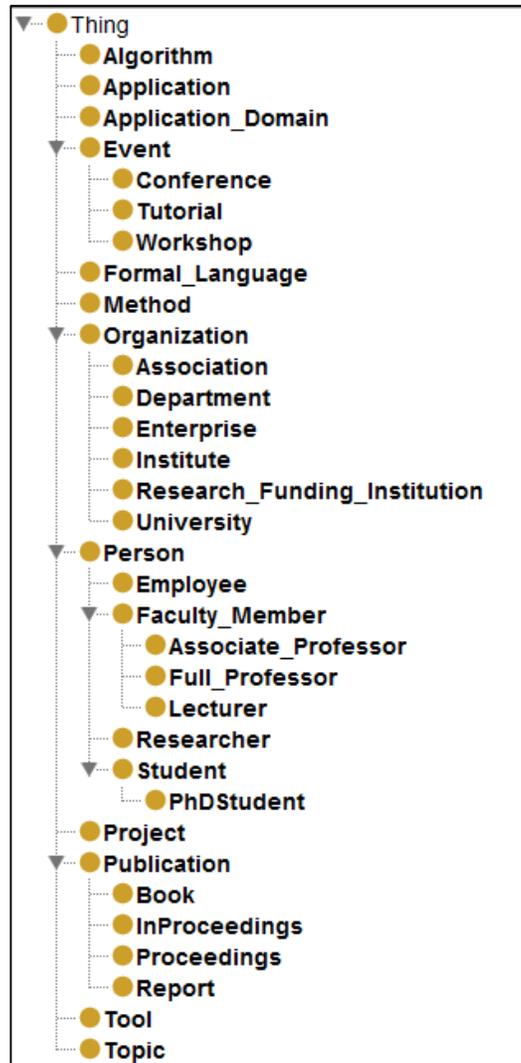


Figure 4-2: Overview of ISWC ontology as viewed in Protégé

4.2.3. Defining WISE Combined Ontology

The Combined Ontology representing the WISE website consists of the Presentation Concepts and the Content Concepts. The Presentation Concepts are denoted by the SADle Upper Ontology with the WISE extension; while the Content Concepts are denoted by the ISWC Ontology. We use the Protégé to combine both concepts by making two subclasses of the “Thing” object: Presentation and Content. The complete structure of the Combined Ontology is illustrated in Figure 4-3.

4.3. Adding Semantic Annotations to WISE pages

In this step, the semantic annotations are added to the web pages of the WISE website, based on the concepts of the Combined Ontology. We do this using the implementation of the CREAM framework: OntoMat, as previously discussed in Section 3.3.

OntoMat’s interface consists of a web browser, in which we could input the web addresses of each page of the WISE website that we wish to annotate. In addition to the web browser, OntoMat has an Ontology Browser, in which we could load the Combined Ontology file that we have created previously using Protégé. By having a complete view of the Combined Ontology structure on one side and a WISE web page on the other side, we could add annotations by dragging and dropping selected text into specific classes of the Combined Ontology. This process is considered as the instantiation process.

For example, annotating the home page of WISE results in creating instances of the Combined Ontology classes, as in the following sample code:

```
<WISE_SADle_ISWC:contenttitle
  rdf:about="http://wise.vub.ac.be#About_WISE">
  <rdfs:label>About_WISE</rdfs:label>
</WISE_SADle_ISWC:contenttitle>

<WISE_SADle_ISWC:University
  rdf:about="http://wise.vub.ac.be#Vrije_Universiteit_Brussel">
  <rdfs:label>Vrije_Universiteit_Brussel</rdfs:label>
</WISE_SADle_ISWC:University>

<WISE_SADle_ISWC:pageheadermenu
  rdf:about="http://wise.vub.ac.be#WisePageHeaderMenu">
  <rdfs:label>WisePageHeaderMenu</rdfs:label>
</WISE_SADle_ISWC:pageheadermenu>

<WISE_SADle_ISWC:contenttext
  rdf:about="http://wise.vub.ac.be#AboutWiseParagraph_">
  <rdfs:label>AboutWiseParagraph</rdfs:label>
</WISE_SADle_ISWC:contenttext>
```

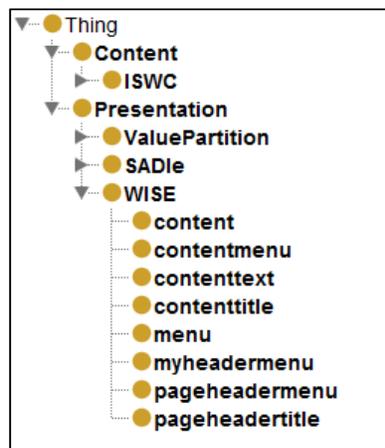


Figure 4-3: Overview of the Combined Ontology

In addition to class instances, OntoMat creates an XPointer object for each class instance to indicate the exact position of that instance in the HTML document:

```
<ontomat:ReificationDataIndividual>
  <ontomat:CreationSource>
    http://wise.vub.ac.be#xpointer(//point() [440]/range-
      to(//point() [450])
  </ontomat:CreationSource>
  <ontomat>AboutIndividual>
    http://wise.vub.ac.be#About_WISE
  </ontomat>AboutIndividual>
  <rdfs:label>about: http://wise.vub.ac.be#About_WISE</rdfs:label>
</ontomat:ReificationDataIndividual>

<ontomat:ReificationDataIndividual>
  <rdfs:label>
    about: http://wise.vub.ac.be#Vrije_Universiteit_Brussel
  </rdfs:label>
  <ontomat:CreationSource>
    http://wise.vub.ac.be#xpointer(//point() [76]/range-
      to(//point() [102])
  </ontomat:CreationSource>
  <ontomat>AboutIndividual>
    http://wise.vub.ac.be#Vrije_Universiteit_Brussel
  </ontomat>AboutIndividual>
</ontomat:ReificationDataIndividual>

<ontomat:ReificationDataIndividual>
  <ontomat:CreationSource>
    http://wise.vub.ac.be#xpointer(//point() [24]/range-
      to(//point() [75])
  </ontomat:CreationSource>
  <ontomat>AboutIndividual>
    http://wise.vub.ac.be#WisePageHeaderMenu
  </ontomat>AboutIndividual>
  <rdfs:label>
    about: http://wise.vub.ac.be#WisePageHeaderMenu
  </rdfs:label>
</ontomat:ReificationDataIndividual>

<ontomat:ReificationDataIndividual>
  <ontomat:CreationSource>
    http://wise.vub.ac.be#xpointer(//point() [459]/range-
      to(//point() [1668])
  </ontomat:CreationSource>
  <rdfs:label>
    about: http://wise.vub.ac.be#AboutWiseParagraph
  </rdfs:label>
  <ontomat>AboutIndividual>
    http://wise.vub.ac.be#AboutWiseParagraph
  </ontomat>AboutIndividual>
</ontomat:ReificationDataIndividual>
```

Adding semantic annotations to the pages of the WISE website would eventually result in a group of HTML documents embedded within them the class instances and their associated XPointer objects.

4.4. Using VR-WISE approach

In this section, we will use the VR-WISE approach and its implementation toolbox: OntoWorld, for creating a virtual world that represents the WISE website. As previously described in Section 3.4, implementing the VR-WISE approach to transform WISE website into a Virtual Reality Environment goes through three phases: the Specification phase, the Mapping Phase and the Generation Phase. We are going to demonstrate implementing those phases in the following subsections.

4.4.1. Implementing Specification Phase

The Specification Phase of the VR-WISE approach concentrates on defining the concepts of the domain underlying the virtual world to be built. In our approach, the specification phase involves defining domain and world specification as being defined previously in the Combined Ontology and the annotations respectively. In other words, in order to define the Domain Specifications for the virtual world we are going to use the concepts of the Combined Ontology, and for defining the World Specifications we use the semantic annotations (e.g. class instances).

The OntoWorld GUI allows the designer to create concepts and instances in a user-friendly manner, by specifying its name and possible relations with other objects. In our case study, the Domain specification of WISE has the same structure as the Combined Ontology; while the World Specification contains a list of instances reflecting the semantic annotations which have been added to the web pages of the WISE website.

The designer of the Virtual World could specify further details about the spatial relations between instances. For example, in our case study, we have reflected the design of the home page of WISE into the virtual reality scene. For instance, we have specified that the AboutWiseParagraph is positioned to the Right of GeneralInfoMenu with a distance of 30 meters from the right side of GeneralInfoMenu. Similar specification has been added to the rest of the instances.

4.4.2. Implementing Mapping Phase

In the Mapping phase, we will link the WISE specification concepts with the actual 3D objects that will represent it in the virtual world. In the beginning we will specify the domain mappings, meaning that each domain concept will have a specific 3D object to represent it, and this is the default mapping for all instances of this concept. If a specific instance has a different 3D object other than the default object, then this specific instance is mapped with that object at the second step of the mapping phase.

Several applications are used for designing and viewing 3D objects: Google Sketchup²⁷ and Flux Studio²⁸. Google Sketchup is used for modelling 3D objects with a collection of useful tools. There is a possibility to reuse existing models by searching the Google 3D warehouse where other designers share their models and supply them for anyone to download and use directly in Google Sketchup. An additional plug-in is used on top of Google Sketchup to export models in DirectX format, which is compatible with OntoWorld.

In the case study, the concept “menu” is represented as a pin, and we need to keep this default object for all of the instances of the concept “menu”. The concept “Person” has a default avatar to represent it. Instances may have different mappings according to the role of this person at the university; being a researcher, a full professor, a student or an employee.

4.4.3. Implementing Generation Phase and Viewing the Results

The third phase in the Virtual World creation is the generation phase. The OntoWorld uses the information gathered from the previous two phases to generate a three-dimensional scene that represents the Virtual World. In our case study, we have created a Virtual World that resembles the home page of the WISE home page. OntoWorld generates a file in FXW extension that can be viewed in Flux Studio, which is a Web3D authoring tool.

4.5. Conclusion

Transforming a classical two-dimensional website into a three-dimensional interface using semantic annotations is an incremental process of several steps, as the outcome of each step is used as an input to the next step and so on. In our approach we use both the Presentation Concepts and the Content Concepts of a website to define semantics and add annotations to the web pages that we wish to transform into a three dimensional environment. Using VR-WISE facilitates the process of translating formal concepts into 3D objects in an initiative manner that does not require special skills or knowledge about building Virtual World applications. Our approach can be further extended in order to automatise the process of transformation by developing some additional features into OntoWorld, such as a feature for importing an ontology.

²⁷ <http://sketchup.google.com/>

²⁸ http://wiki.mediamachines.com/index.php/Flux_Studio

Chapter 5 Conclusions & Future Work

The development process of Virtual Reality applications is considered to be a complex process, where the developer is required to have an adequate knowledge in a lot of related issues concerning programming 3D environments, along with extensive background in graphics and designing techniques. A lot of approaches are being developed in order to facilitate this process in a way that enables inexperienced users to create, design and customise their own virtual reality applications or interfaces. As a result to the rapid evolution of the hardware manufacturing and networking capabilities, it is becoming worthwhile to provide the users of the web with Virtual Reality applications. Several successful approaches are already appearing on the World Wide Web, and it is likely for user interfaces to transform and evolve into VR interfaces in the foreseen future.

Several initiatives are taking place on the web to transform classical web sites into 3D interfaces. However, most of those approaches are straight forward, and simply transform the obvious web elements (such as links and photos) into Virtual Reality objects. In this thesis work we have been investigating the possibilities of improving those straight forward methods into a transformation that considers web site semantics into the conversion process. The result of this research is an approach that includes semantic annotations and several other techniques in the process of transforming an existing 2D web site into a VR environment.

In order to achieve that goal, our proposed approach goes through several steps. First of all, the presentation and contents of the considered web site is analysed. The analysis process would end up with a list of concepts that represents the web site terminology, formally described in a form of a Combined Ontology.

Second of all, the Combined Ontology is used to add semantic annotations to a selected group of the web site HTML pages. The outcome of the annotation process is a set of annotated pages that includes the original HTML code, in addition to the embedded annotations information.

Next, the results of the previous two steps are used as an input to the VR-WISE approach, which is an approach for creating VR environments based on domain specifications. VR-WISE uses the gathered information from the Combined Ontology and the semantic annotations as a starting point to creating a VR environment that resembles the considered web site. Further mappings and visibility settings are required to be specified by a designer in order to translate the web site formal concepts into 3D objects. As a result, VR-WISE generates the VR source code.

The implementation of the VR-WISE approach is fulfilled in a tool box called OntoWorld, developed at the WISE lab of the Vrije Universiteit Brussel. It is intended to be used while performing the steps of the VR-WISE approach to create

VR environment based on the specifications of a particular domain. However, we have used this tool box to achieve our proposed approach.

The future work of this research would be to extend the current version of the OntoWorld toolbox in order to minimize or eliminate –if possible- the manual operations that have been performed during this thesis work. For example, OntoWorld could be extended with some additional features to translate Ontology files into domain specifications. Another extension would be a style translator that uses CSS style sheets to decorate and set up the VR interface. OntoWorld could be as well integrated with certain existing tools for designing 3D objects in order to be used in the VR library of OntoWorld, and so on.

This thesis is meant to improve the transformation of 2D web sites into 3D interfaces. Nonetheless, there is a great potential for further improvements to be done in this area.

Bibliography

Bechhofer, S., Harper, S., & Lunn, D. (2006). SADle: Semantic annotation for accessibility. *5th International Semantic Web Conference*. Citeseer.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution*.

Bille, W., De Troyer, O., Kleinermann, F., Pellens, B., & Romero, R. (2004). Using ontologies to build virtual worlds for the web. *The International Conference on WWW/Internet 2004*. Madrid, Spain.

COHSE: Conceptual Open Hypermedia Service. (n.d.). Retrieved July 2009, from <http://cohse.cs.manchester.ac.uk/>

DANTE: Mobility Support for Visually Impaired Web Travellers. (n.d.). Retrieved July 2009, from <http://dante.cs.manchester.ac.uk/>

De Troyer, O., Kleinermann, F., Pellens, B., & Bille, W. (2007). Conceptual modeling for virtual reality. *Twenty-Sixth International conference on conceptual modeling*. New Zealand.

ExitReality - The Entire Web in 3D. (n.d.). Retrieved March 2009, from <http://www.exitreality.com/>

Flux Studio. (n.d.). Retrieved August 2009, from http://wiki.mediamachines.com/index.php/Flux_Studio

Google SketchUp. (n.d.). Retrieved August 2009, from <http://sketchup.google.com/>

Handschuh, S. (2007). Semantic Annotation of Resources in the Semantic Web. In R. Studer, S. Grimm, & A. Abecker, *Semantic Web Services: concepts, technologies and applications*. Springer.

Handschuh, S., & Staab, S. (2002). Authoring and annotation of web pages in CREAM. In *Proceedings of WWW-2002*. CiteSeer.IST.

Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-cream-semi-automatic creation of metadata. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*. Springer.

Handschuh, S., Staab, S., & Maedche, A. (2001). Metadata and the Semantic Web—and CREAM. In *Proceedings of the DELOS-2001 workshop*. Darmstadt: ERCIM.

Harper, S., & Yesilada, Y. (2007). Web authoring for accessibility (WAFA). *Web Semantics: Science, Services and Agents on the World Wide Web*, 5 (3), 175.

Kashyap, V., Bussler, C., & Moran, M. (2008). *The semantic web: Semantics for data and services for the web* (Vol. XVI). Springer Berlin Heidelberg.

Kleineremann, F., De Troyer, O., Creelle, C., & Pellens, B. (2007). Adding semantic annotations, navigation paths and tour guides to existing virtual environments. *Proc. 13th International Conference on Virtual Reality Systems and Multimedia (VSMM'07)*. Brisbane, Australia.

Kleineremann, F., De Troyer, O., Mansouri, H., Romero, R., Pellens, B., & Bille, W. (2005). Designing semantic virtual reality applications. *Proceedings of the 2nd INTUITION International Workshop*, (pp. 5-10). Senlis, France.

Koivunen, M., & Miller, E. (2001). W3C Semantic Web Activity. *Proceedings of the Semantic Web Kick-off Seminar*, (pp. 27-44). Finland.

Lunn, D. (2008). *Building ontologies for the SADle transcoder*. HCW technical report, University of Manchester, School of computer science.

Lunn, D. (2005). *SADle: Structural semantics for accessibility and device independence*. Master Thesis, University of Manchester, School of Computer Science.

Lunn, D., Bechhofer, S., & Harper, S. (2008). A user evaluation of the SADle transcoder. *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (pp. 137-144). New York: ACM.

Nieto, M. (2003). *An Overview of Ontologies*. Universidad De Las Américas Puebla, Interactive and Cooperative Technologies Lab.

OntoMat Annotizer. (n.d.). Retrieved July 2009, from <http://annotation.semanticweb.org/ontomat/index.html>

OWL Web Ontology Language Overview. (n.d.). Retrieved July 2009, from <http://www.w3.org/TR/owl-features/>

Pellens, B. (2007). *A conceptual modelling approach for behaviour in virtual environments using a graphical notation and generative design patterns*. Ph.D. Thesis, Vrije Universiteit Brussel, Brussels, Belgium.

Pellens, B., Kleineremann, F., De Troyer, O., & Bille, W. (2006). Model based design of virtual environment behavior. (H. Zha, Ed.) *Lecture Notes in Computer Science* (4270), 29-39.

Second Life. (n.d.). Retrieved March 2009, from <http://secondlife.com/>

Simpson, J. (2002). *XPath and XPointer: Locating Content in XML Documents*. O'Reilly.

SMORE - Create OWL Markup for HTML Web Pages. (n.d.). Retrieved July 2009, from <http://www.mindswap.org/2005/SMORE/>

Structural-Semantics for Accessibility and Device Independence (SADle). (n.d.). Retrieved June 2009, from <http://hcw.cs.manchester.ac.uk/research/sadie/>

The Protégé Ontology Editor and Knowledge Acquisition System. (n.d.). Retrieved July 2009, from <http://protege.stanford.edu/>

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., et al. (n.d.). Building Semantic Intranets: What is needed in the annotation toolbox?

Vivaty Player. (n.d.). Retrieved July 2009, from <http://www.vivaty.com/>

Yesilada, Y., Harper, S., Goble, C., & Stevens, R. (2004). DANTE: Annotation and transformation of web pages for visually impaired users. *Proceedings of the 13th international World Wide Web conference* (pp. 490 - 491). New York: ACM.

Yesilada, Y., Stevens, R., Harper, S., & Goble, C. (2007). Evaluating DANTE: Semantic transcoding for visually Disabled Users. *ACM Transactions on Computer-Human Interaction* , 14 (3).

