Designing Semantic Virtual Reality Applications

F. Kleinermann, O. De Troyer, H. Mansouri, R. Romero, B. Pellens, W. Bille

WISE Research group, Vrije Universiteit Brussel, Pleinlaan 2,
1050 Brussels, Belgium

frederic.kleinermann@vub.ac.be

Abstract
Nowadays, the development of a Virtual Reality (VR) application is still very time-consuming and hard for a non VR-expert. In addition, the VR applications often lack the necessary semantic information to make the VR application more intuitive and more attractive for the end-user. This paper describes a new approach for designing VR applications using ontologies, which make the design of the VR application more domain-oriented and more intuitive. It also describes how our approach can be used to add richer semantic information that is more domain-oriented during the design process of a VR application. We illustrate the advantages of such a semantic annotation process by means of a semantic search engine for VR.

Keywords: Virtual Reality, Ontology, Search Engine, VR furniture shop

1 Introduction

Nowadays t-he design and the development of Virtual Reality (VR) applications are still time-consuming and expensive. Although there are a number of tools to help the designer in the creation of a VR application, they are not non-VR expert friendly. For this reason, the design and development of a VR application are usually done by a VR expert. Problems may arise when the VR expert is not familiar with the domain. Then, the VR expert may lose a lot of time to get acquainted with the domain. The first prototype of the VR application often presents many shortcomings and usually does not match all the requirements of the client. As a consequence, several iterations will be needed before the result reaches an acceptable level of satisfaction for both the client and the end-users. Therefore, the development process is time-consuming and expensive (Bille et al, 2004), (De Troyer et al, 2003). Furthermore, the maintenance of a VR application may pose problems if it is a non VR expert who has to do it and if he has not been involved with the development of the VR application.

Once a VR application has been developed, it often lacks the necessary semantic information in order to be more usable and more intuitive for less experienced end-users (who may not be VR experts or neither domain experts) (Cavazza and Palmer, 2000) (Otto, 2005). This statement can be illustrated with search engines currently developed for searching objects and locations inside a virtual world. Although they may be advanced in terms of algorithms incorporating the latest Artificial Intelligence techniques such as Fuzzy logic (Ibanez, 2004), they are often limited in terms of possible queries or even possible answers as the VR application simply does not contain enough semantic information about the domain of the application. Usually, the information associated with a virtual world is limited to the usual low-level information such as the type of geometry, the size and material (Ibanez, 2004). Furthermore, the process of adding some semantic information is often done after the virtual world has been created (Ibanez, 2004). This again increases the development time and cost of a VR application.

Therefore, to make VR development accessible for a broader audience and in order to reduce the overall development time and cost, as well as to enhance the usability of a VR application for a larger audience, it is necessary to find ways which will allow to involve the domain expert more into the design process and which will facilitate the addition of semantic information during the design process. This paper describes how ontologies can make the development of a VR application more domain-oriented, allowing the designer to express the VR application using domain terminology and at a high level, free from any specific low-level VR technology concepts. The paper also shows how semantic domain-oriented information can be added during the design. The paper is structured as follows. Section 2 will present our approach, called VR-WISE. Section 3 presents how the semantic information added during design can be exploited by a search engine to provide more powerful queries. Section 4 presents conclusion and further work.

2 VR-WISE Approach

Our approach called “VR WISE” (Bille et al, 2004), (De Troyer et al, 2003) uses high-level specifications and domain ontologies to allow the designer to express a VR application in a more domain-oriented way (i.e. using the concepts and the terminology of the domain) and more intuitively (by specifying the design of the VR application at a higher-level i.e., free from any implementation and current technical limitations). VR-WISE approach also uses ontologies as the underlying representation formalism. In its most simple form, we can say that an ontology is an abstraction of a computer-based lexicon, thesaurus, glossary or some other type of structured vocabulary, extended with knowledge about a given domain (Gruber, 1993; Guarino and Giaretta, 1995). Ontologies are used in VR-WISE for two different purposes: (1) explicitly during the design process for representing knowledge about the domain under consideration; (2) internally as general information representation formalism.

The design process in the VR-WISE approach consists of three sequential steps, namely the specification step, the mapping step and the generation step (see figure 1).

In the rest of the paper, a common example will be used to illustrate the approach. The example is a furniture shop like IKEA which will be reconstructed virtually. The user can walk through the virtual shop and buy some furniture. He can also ask for information about any kind of product in this virtual shop. A search engine is available to find products in the virtual furniture shop.

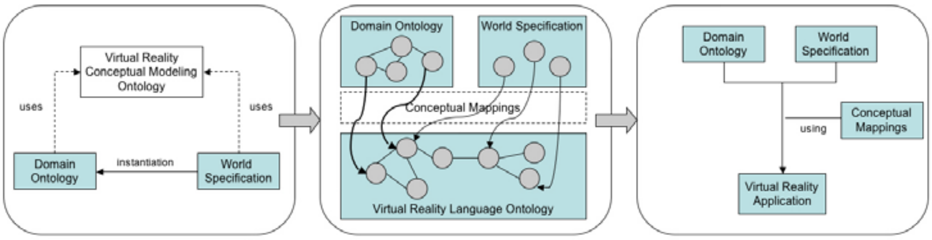

Figure 1: Overview of the VR-WISE approach

2.1 The Specification Step

The specification step allows specifying the VR application at a high level, using domain knowledge, and without taking any implementation details into account. The specification is done at two levels: the type level and the instance level. The type level is specified using a *Domain Ontology* and describes the concepts needed from the domain under consideration. The instance level is specified by means of the *World Specification*, which will contain the actual conceptual description of the VR application to be built.

During the specification step, extra semantic information can be added by the designer to concepts, instances and behaviors. This will be done by the process of annotation which can be semi-automatic for some type of semantic information and manual for other type of semantic information. To clarify what type of semantic information is used in our approach, we classify the semantic information into two categories namely those related to the visual properties of an object and those related to non-visual properties of an object. The first category can be a semi-automatic process of annotation where the designer does not have to specify all the semantic information of an object as they can be retrieved from its geometrical structure. This category of semantic information is not very domain oriented as it relates to the general visual representation of objects inside the virtual world. The second category of semantic information is more domain-oriented as it deals with the attributes of an object from the perspective of the particular domain.

2.1.1 Concepts

Concepts can be compared to class definitions from the Object Oriented (OO) modeling paradigm. The *Domain Ontology* describes the domain concepts by means of their properties as well as their relationships. For example, in a furniture shop domain, this ontology contains concepts like beds, wardrobes, desks, sofas and so on. It is also possible that such a domain ontology already exists (originally created for other purposes). In that situation, this ontology can be reused. Concepts are specified by means of their attributes (properties). In our example of the virtual furniture shop, an example of an attribute is the size of furniture.

Semantic information can be associated to a concept by annotating it further using its attributes. The first category of semantic information will be related to its attributes defining its visual representation inside a virtual world such as the size, and the material of a concept. The second category of semantic information will be related to other non-visual oriented attributes such as in case of the example its weight, its packing size, its price.

In our example of the virtual furniture shop, the concept of “Lamp” will have attributes like height, shape, and color, which are defining the visual representation of a Lamp, and other attributes (like price and weight), which define the concept as a product from the viewpoint of the shop domain. Not only attributes can be annotated, but also the overall concept may be annotated with some semantic information describing for instance that the Lamp is an electricity lamp and it must be used with a voltage of 220 volts.

Once the *Domain ontology* has been created, its concepts become available for the *World Specification*.

2.1.2 Instances

In the World Specification, the designer specifies the actual virtual world by instantiating concepts from the Domain ontology. These instances represent the object that will populate the virtual world. Instances inherit the attributes of its concept. As value for an attribute, the default value specified in the concept definition can be used or a different value can be given.

Like for a concept, semantic information can be associated to the attributes of an instance. Furthermore, the instance will inherit the semantic information associated to its concept. This provides a way of automatically annotating instances. However, the semantic information inherited can be removed or changed. New semantic information can also be added to the attributes of an instance.

In our example of the virtual furniture shop, a study lamp can be created by instantiating the concept of “Lamp”. The instance “study lamp” will inherit all the attributes of the concept “Lamp” as well as its semantic information. The designer can change the default values inherited by the concept as well as the semantic information. In the case of the study lamp, the user may want to change the default value of the weight, and he may want to express that the study lamp is working with a voltage of 110 volts.

2.1.3 Relations

To position and orientate instances inside the virtual world, our approach provides high-level relations (Pellens et al, 2005) that may help the designer to structure the virtual world more easily and more intuitively. Although the approach also allows to position instances by means of coordinates and to orientate them by means of an angle, VR-WISE provides a more intuitive way where instances are positioned and oriented with respect to other instances. At this moment, two kinds of high-level relations are provided namely *spatial relations* (binary relations) and *Orientation relations*. A spatial relation uses high-level concepts (such as Left, Right, Front, Back, Above, Under, Middle and On Top) to position an instance with respect to another instance. For example in the virtual furniture shop, an instance “study lamp” can be positioned with respect to the instance “desk” by using a spatial relation to express that the study lamp is placed on top of the desk. Orientation relations are used to orientate an instance with respect to another by specifying which part of which side of an object is oriented to which part of which side of the other object. For instance, the designer can express that the right front of the chair is placed towards the left front of the table. Furthermore, a graphical notation has been developed so that the designer can use this graphical notation to specify the position and orientation of each instance (Pellens et al, 2005).

All these high-level relations provide some kind of semantic relation between the instances. The semantics of these relations can be used later on to ask questions like “what is left of product A”.

Another way to add semantics to instances is by means of the notion *Rule*. Rules provide a way to compare different instances of a same concept at a semantic level. A rule provides some adjectives together with a meaning so that two instances of that concept can be compared using these adjectives. For instance, a rule can be specified to decide if “a study lamp is cheap”. A rule has four elements namely an adjective (e.g. cheap), the property of the concept on which this adjective applies (e.g. price), the comparison operator (e.g. less than) and a judgment value that provides some meanings for the comparison (e.g. 15 Euros). When an instance of the concept Lamp is created like the study lamp and if a value of 14.95 Euros is given for the attribute “price”, then the instance “study lamp” will be tagged automatically as “cheap”. Later on, a search engine for VR can use this information to answer end-user questions like “Is this objects cheaper than that object ?”. It is important to note that these rules must be created for each concept separately since for a specific adjective, the judgment values may vary from one concept to another.

2.1.4 Behaviors

It is also possible to assign behaviors and interactions to instances and concepts in our approach (Pellens et al, 2005). Our approach provides two different categories of behaviors namely basic VR behavior used to specify simple behaviors and domain-specific behavior used to provide behavior specific to a domain. For instance, for the domain of the furniture shop, domain-specific behavior is e.g. requesting information or placing a furniture into a trolley.

2.2 The Mapping Step

The mapping step is the second step in VR-WISE approach. In this step, the mapping from the conceptual level to the implementation level is specified. Similar to the specification step, the mapping is defined at two levels. The *Domain Mapping* defines the mappings from the concepts in the Domain Ontology to VR implementation primitives. The purpose of this mapping is to specify how a domain concept should be represented in the virtual world. This is a kind of default mapping that is also very useful for rapid prototyping. For example, in the shop domain, a bin could be mapped onto a cylinder. In this way, each instantiation in the World Specification of the bin concept can be represented as a cylinder.

Although instances may be of the same type (concept), they may in some cases require different representations. Therefore, the *World Mapping* allows defining mappings for instances to VR implementation primitives. In this way the default mappings, specified in the Domain Mapping, can be overwritten.

The mapping step is very important, as it is the step where the designer and the domain expert must agree on how to represent a concept inside the virtual world. For instance, in the virtual furniture shop, the concept “Lamp” from the domain ontology has several attributes. Among them, one attribute “Shape” and two other attributes dealing with the size namely an attribute called “Height” and an attribute called “Base”. The VR-expert can decide together with the domain expert to map the attribute “Shape” to a particular geometry to visually represent the concept “Lamp”. They can also decide to map the “Height” attribute to the height of the bounding box containing the geometry of the lamp and the attribute “Base” to the width of the bounding box.

2.3 The Generation Step

The generation step generates the actual code for the virtual world specified in the Specification Step using the mappings defined in the Mapping Step, i.e. the conceptual specifications given by means of the Domain Ontology and the World Specification are converted into a working application by means of the mappings given by the Domain Mapping and the World Mapping. We have developed a Tool called “OntoBasis “ to support our approach. OntoWorld is written in C++ and uses DAM+OIL (Pellens et al, 2005), (http://www.daml.org/2000/12/daml+oil.daml) for the ontologies.

OntoWorld generates two files. The first file contains the 3D scene structure with its objects in VRML format (Hartman and Wernecke, 1998). The second file contains the semantic information generated in an MPEG-7 format. MPEG-7 is used, as it is an ISO/IEC standard supported by the industry (Salembier and Smith, 2001) and it is readable by human and machine (Ucelii et al, 2005). It also has a tree representation and therefore, it can easily be used to make a one-to-one mapping between the 3D scene structure generated by our tool and its semantic information.

We have chosen to generate two files (one containing the 3D structure of the virtual world and one containing the semantic information) as it provides some advantages for the maintenance of the VR application. For instance, in our example, the MPEG-7 needs only to be updated if the prices of products of the VR furniture shop have changed. Furthermore, if the user wants to provide the semantic information in different languages, only this file needs to be updated (Ucelii et al, 2005).

3 An Application: A Semantic Search Engine

To illustrate how these two files (MPEG-7 and VRML/X3D) can be used, we have developed an applet which reads the MPEG-7 and communicates with the VR plug-in namely Cortona (http://www.parallelgraphics.com/products/cortona/) used to display the virtual world. For our example, the virtual furniture shop is generated in VRML format. All of this will be contained inside a normal html page which can then be loaded via any web browser. Furthermore, to illustrate the advantages of the semantic annotations, a search engine has also been developed and attached to the applet to allow the end user to formulate queries in order to find objects quickly.

In the furniture shop, by means of the search engine, the end-user can find any furniture belonging to the VR furniture shop using different kind of queries. For instance, he can look for furniture that belongs to the living room section. In that case, the search engine will display the different products that you can find in the living room like a “sofa” or a “table”. The user can refine his query and ask for an expensive sofa made of leather. As a result the user will directly be transported to the particular products (see figure 2a). If the user walks in the shop, he can also directly find information on any furniture by pointing at it (see figure 2b). The user can see that the chair is capable of rotating and it will rotate on itself by clicking on it.

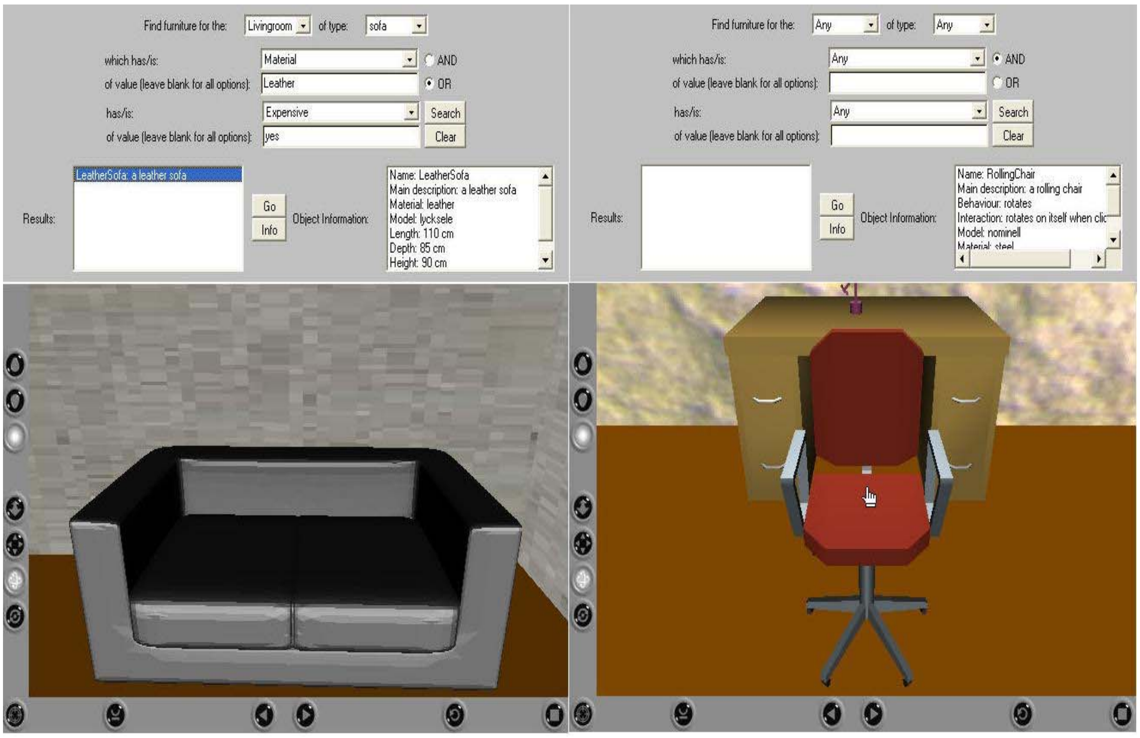

Figure 2a. User asks for an expensive sofa (left). Figure 2b. Users points at the chair and all the information related to the chair is being displayed inside the applet (right).

4 Discussion

Currently, the approach still has some limitations. For instance, the current implementation of OntoWorld does not yet allow to semantically describing complete scenes. It is also limited to static virtual worlds. More works need to be done in terms of how the semantic information can be updated when we have to deal with dynamic virtual world in which the objects populating the virtual world may be moved to different locations in the virtual world or deleted or new objects may be created dynamically.

5 Conclusion

The work presented in this paper describes an approach for designing VR applications using high-level modeling concepts and ontologies. In this way, the design of the VR application is more domain-oriented and more intuitive for the designer. This approach allows for a more active participation of a domain expert during the design phase. In addition, the designer can add rich semantic information during the design process of a VR application. This semantic information will enhance the usability of the VR application. For instance, the semantic information can be exploited by a simple search engine to allow more powerful and domain-oriented queries. Also inside the virtual world domain specific information can be provided to the end-user (e.g. by clicking on an object).

6 References

Bille, W., De Troyer, O., Kleinermann, F., Pellens B. and Romero, R. (2004): Using Ontologies to build Virtual Worlds for the Web, In *Proceedings of the IADIS International WWW/Internet 2004 Conference*, Madrid, Spain, 683-690

Cavazza, M., Palmer, I. (2000): High-level Interpretation in Virtual Environments, *Applied Artificial Intelligence*,14(1), 125-144

De Troyer, O., Bille, W., Romero, R. and Stuer, P. (2003): On Generating Virtual Worlds from Domain Ontologies, In *Proceedings of the 9th International Conference on Multimedia Modeling*, Taipei, Taiwan, 279-294

Gruber, T.R. (1993): translation approach to portable ontologies, *Journal of Knowledge Acquisition, 5(2)*, 199-220.

Guarino, N. and Giaretta, P. (1995): Ontologies and knowledge bases: towards a terminological clarification, Towards Very Large Knowledge Bases: *Knowledge Building Knowledge Sharing*, IOS Press, Amsterdam, 25-32

Hartman, J. and Wernecke, J. (1998): *The VRML 2.0 Handbook*, Addisson-Wesley Publishing

Ibanez, J. (2004): *An intelligent guide for virtual environments able to answer fuzzy queries and tell stories from her own perspective*, PhD thesis, University of Murcia, Murcia, Spain

Otto, K. (2005): Semantic Virtual Environments, In *Proceedings of the 14th international World Wide Web conference*, ACM press, Chiba, Japan, 1036-1037

Pellens, B., De Troyer, O., Bille, W. and Kleinermann, F. (2005): Conceptual Modeling of Object Behavior in a Virtual Environment, In *Proceedings of Virtual Concept*, Springer-Verlag, Biarritz, France, 93-94

Salembier, P.E. and Smith, J. R. (2001): MPEG-7 Multimedia Description Schemes, *IEEE transactions on circuits and systems for video technology*, 11(6), 748-759

Ucelii, G., De Amicis, R., Conti, G. (2005): Shape Semantics and Content management for industrial Design and virtual Styling, *Workshop towards Semantic Virtual Environments' (SVE)*, 16-18 March, Villars, Switzerland, 127-137

http://www.daml.org/2000/12/daml+oil.daml. Accessed 10 June. 2005

http://www.parallelgraphics.com/products/cortona/. Accessed 10 June. 2005