Intelligent Modelling of Virtual Worlds Using Domain Ontologies

Wesley Bille**, Bram Pellens, Frederic Kleinermann, and Olga De Troyer

Research Group WISE, Department of Computer Science, Vrije Universiteit Brussel,
Pleinlaan 2, B-1050 Brussel, Belgium
{Wesley.Bille,Bram.Pellens,Frederic.Kleinermann,Olga.DeTroyer}@vub.ac.be

Abstract. Today, the development of a VR application is still a long and difficult task as new features are being developed for Virtual Reality. There is a need to have better modelling tools which will help the designer to model the VR application much more from the domain expert point of view by involving him more in the design process. In this paper, a new approach is introduced for designing and developing a VR application where the domain expertise is used for generating it more easily. This approach uses ontologies as a way of grasping the knowledge of a domain, expressing the virtual world much more in terms of the end-user domain, generating the virtual world more easily and performing intelligent reasoning. Furthermore, ontologies are also used as a conceptual modelling tool allowing a non-VR-skilled person to model his VR application using the concepts of Virtual Reality more intuitively and more oriented towards his domain expertise.

1 Introduction

Today, Virtual Reality (VR) is used in different domains such as industry, military, medicine, teaching and entertainment [10]. The development of a VR application for such domains is still a long and difficult task as new features are being developed for VR such as complex behaviours, better collision detection and artificial intelligence (AI) concepts for making the virtual world more believable. In order to help the designer to create a VR application having all the features used in VR more easily, a number of software tools have been created during recent years. These software tools can be classified either as toolkits (like Java3D [9], VRML [2], Unreal [1]) or as authoring tools (like 3D studio max [14]). These tools are still not intuitive enough for a non-VR-skilled person to use them. Usually, one of these tools is not sufficient for generating a VR application having all the latest features available in VR and the designer often needs to know several of these software tools and how to use them together. For these reasons, the domain expert is often not the person who is developing the VR application, although he is there at the beginning of the project in order to explain his domain and what the VR application should do. As generally accepted in

---

** Research Assistant for the Fund for Scientific Research - Flanders

classical software engineering, it is important to ensure that the domain expert stays involved in the different stages of the development of a VR application in order to provide the necessary input and feedback. As new concepts are being developed for VR to model the reality better, there is a need to have tools which will help the designer to model the VR application much more from the domain expert point of view and to let him participate in the specification of the VR application in a more intuitive way using his own domain expertise and terminology.

In the recent years, some people have started to develop methods which separate the design process into a high level and a low level phase, helping the developer to design realistic interaction techniques [12],[6]. Others have developed methods which take a more user-centred approach to the specification, design, development and evaluation of a virtual environment [15],[11]. These approaches define methodologies for designing a VR application which is meeting the end-user requirements. However, they do not express the modelling of the VR application in terms of the domain of interest. Not only, it is important to incorporate the terminology of a domain, but it is also important to make sure that a domain expert with this terminology of the domain can fully exploit his domain expertise for designing his VR application. The use of ontologies in our approach gives us the advantage of having existing ontologies available for different domains. This allows us to grasp the knowledge from a certain domain. Our approach provides a way for a domain expert to design his VR application without having to know VR building primitives which are necessary for generating the code. Therefore, our approach uses an ontology as a way of

- *grasping the knowledge of a domain*
  The domain knowledge can be captured by a domain ontology. Such an ontology for the domain of the end-user can be either an existing one or a new one created explicitly.
- *expressing the virtual world much more in terms of the end-user domain*
  By using a domain ontology, the design of the virtual world can be expressed in terms of the end-user's domain. Therefore, it will be more intuitive for a domain expert to communicate about the design. For instance, the fact that an object is in front of another can be presented to him as: object A is "in front of" object B (using the proper domain terminology names for the object A and B) rather than giving him low level specifications such as positions for the primitive VR objects.
- *generating the virtual world more easily*
  From the knowledge given in the domain ontology, it is possible to derive a number of properties for an object. It becomes easier to generate (semi-) automatically the VR application. For instance, a cube generated by a language like the Virtual Reality Modeling Language (VRML) will need to have a width, a height, a depth and a position reference in the virtual world to be specified. All these parameters need to be specified explicitly. Using an ontology, it may be possible to derive that a particular object should be

modelled as a cube with the appropriate values for its parameters and be positioned at the centre of the virtual world.
– *making intelligent reasoning*
It is not easy to make intelligent reasoning from a scene graph or from APIs like Java3D. However, it is possible to do that by using an ontology. For example using ontologies, it is easier in a virtual world having hundreds of objects to find out quickly which objects are touching other objects rather than each time, looking at the scene graph to compute their respective distances for knowing which ones are touching.

Our approach can be seen as a more intelligent way of designing a VR application for two reasons. Firstly, the domain expert can design his VR application using his own domain expertise (terminology and concepts from his domain) without having to be a VR specialist. Secondly, the code generated for his VR application takes into account certain properties from his specification. The paper continues by explaining our approach in more details. An example is then used to illustrate our approach. Finally, the paper finishes by discussing the current stage of the work with its current limitations as well as further research.
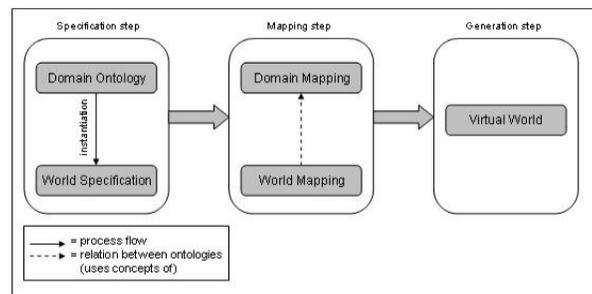
2 Overview of the approach

Fig. 1. Software Architecture

Today, ontologies are used in different domains (for more information about ontologies, see [3],[4]). Simply stated, an ontology can be seen as an abstraction of a computer-based lexicon, thesaurus, glossary or some other type of structured vocabulary, suitably extended with knowledge about a given domain. A domain ontology can be considered to be a representation of a domain conceptualisation describing possible concepts and relationships between these concepts. The information contained in such a domain ontology can be seen as a starting point for a conceptual description of a virtual world for a specific domain [4]. Our approach is divided into three stages namely a specification stage, a mapping stage and a generating stage (see Fig. 1).

2.1 Specification Stage

The aim of the specification stage is to allow the end-user to define his VR application at a conceptual level (free from any implementation or VR software library) using the domain terminology. This stage starts by either selecting an existing domain ontology or by creating one using the knowledge of the domain expert (or end-user). The domain ontology helps to express the virtual world in terms of the end-user domain. In order to realize this, a number of underlying ontologies are needed. Our software uses the DAML+OIL ontology language defined by the DAML Ontology. This ontology was already available. It was developed to be the normative reference on the precise syntax of the language constructs. In fact, it is the machine-readable RDF Schema definition [13] of DAML+OIL [8].

- **The VRML Ontology**
  VRML is used as the language for creating the VR application. An ontology for VRML is needed in order to be able to do the mapping stage. The VRML Ontology contains information like primitive object types that can be displayed using a VRML specification along with the necessary parameters for creating instances of these object types.
- **The VR Conceptual Modelling Ontology**
  Although VRML can be intuitive for someone knowing it, it may not be the case for someone who knows nothing about VRML or VR in general. As it is the purpose of our approach to allow such a person to describe his VR application at a conceptual level using his domain knowledge, another ontology called the "VR Conceptual Modelling Ontology" has been created. This ontology describes the conceptual modelling concepts that a non-VR-skilled person can use to specify the virtual world in an intuitive way. For instance, concepts like "InFrontOf", "Behind", "Touches", "InsideOf" are concepts used in this ontology and they do not exist explicitly in VRML. This ontology is currently being updated with new concepts for space/time relationships and behaviour between objects.
- **The Domain Ontology**
  The Domain Ontology describes the concepts (at a conceptual level) used in the domain along with their properties and relations between them. This is where the general background of the end-user domain is grasped. For the moment, this ontology has a fixed format and needs to be created for each new domain. However in the future, we would like to consider the possibility to use existing ontologies that describe the problem domain for which the virtual world needs to be created.
- **The World Specification**
  The World Specification describes the specific virtual world that someone wants to build at a conceptual level and in terms of the domain concepts. The concepts described in the Domain Ontology are in fact classes of objects in the domain of interest. While the Domain Ontology contains classes of objects, the World Specification contains instances of these concepts which are populating the virtual world.

2.2 Mapping Stage

Using the previous stage, the mapping stage defines the virtual world by mapping the concepts defined in the Domain Ontology onto concepts for VR building blocks (like VR primitives, constraints, and so on). A number of mappings are used during this stage.

– **Meta Mapping**
  The Meta Mapping describes how the DAML+OIL concepts are mapped onto the VRML concepts. A mapping is defined by means of a Source and a Target. The Source concept is defined as a DAML Thing (from the DAML Ontology). The Target concept is defined as a VR-ObjectType (from the VRML Ontology). Both are the most top-level concepts in their respectively ontologies. In this way, we are able to map every concept described using the DAML syntax onto concepts described by the VRML Ontology. For instance, the ConceptMapping is used to describe the mapping of domain concepts to VRML concepts. While the AttributeMapping is used to describe how the attributes of a domain concept should be mapped onto the attributes needed for the VRML concept. At this stage of our research this mapping is currently being extended in terms of mappings for constraints and connections.
– **The Domain Mapping**
  This mapping describes how the domain concepts should be represented by VR object types. Therefore, it is an instantiation of the Meta Mapping. It provides a kind of default mapping for the instances created for a domain concept. However, not all the instances have exactly the same properties. Therefore, another mapping called the World Mapping is introduced.
– **The World Mapping**
  The World Mapping allows to overwrite the default mapping specified for a concept in the Domain Mapping. For instance, a default colour may have been specified in the Domain Mapping for the concept of box and the world may contain a number of instances for this concept. However, one of these instances may need to have another colour than the one used by default. To overcome this problem, the World Mapping has been added so that instances of a concept can have other values than the default one.

2.3 Generating the VR application

Our tool transforms the Domain Ontology, the World Specification, the Domain Mapping and the World Mapping into a working virtual environment by generating VRML code. At this moment, the tool can also generate the code in a different format like the ODE format [5].

3 Results

In order to illustrate the overall approach, an example based on a bowling game is described where it has a ball, pins and a lane with two gutters (see fig. 2).
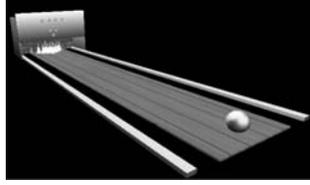
Fig. 2. Virtual Bowling Game

We will now explain how the end-user can come to this result using the proposed approach. Note that we have put the ontology content in an ORM-like [7] notation so that it is easier to understand by the reader.

3.1 Specification

The concepts appearing in a virtual bowling game are firstly described in the Domain Ontology. For instance, there is the concept of a bowling ball, a pin, a lane, a gutter and so on. Not only the concepts are described in the domain ontology, but also relations between the concepts. The designer will for example specify that one gutter is *left of* the lane and another one is *right of* the lane. Fig. 3 shows the description of the concept of a pin in an ORM-like notation and how intuitive relations can be used in the domain description. The World
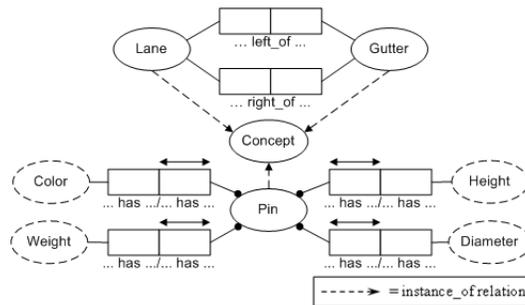

Fig. 3. Snapshot of the Domain Ontology in ORM-like notation

Specification is then specified. Here, the user will define the individual instances populating the world. Suppose that the complete bowling example is described as one object called *bowling lane* consisting of one lane, two gutters, a ball and ten pins, the user will instantiate it once. Suppose the virtual world to generate, is a bowling hall. The user may want to instantiate the bowling lane object three times. In the previous stage, we defined our complete world in terms of domain

knowledge. So far the user of our approach did not have to use VR terminology in order to specify the world. To be able to generate the virtual world, the user will now have to map the conceptual specification towards VRML concepts which are defined in our VRML Ontology. In the first phase of the mapping, which we call the Domain Mapping, the user will have to state the visual presentations for the concepts in the domain. This means that for example, the concept of a pin from the domain will be mapped onto the concept of a cone defined in the VRML Ontology by using the ConceptMapping from the Meta Mapping. In the same way, the concept of a bowling ball will be mapped onto the concept of a sphere. Furthermore, we will map the attribute "weight" from the pin concept onto the attribute "mass" from the cone concept. We will do this for all attributes using the AttributeMapping from the Meta Mapping. We should also take into account exceptions in our world. Suppose we map a pin onto a cone having a white color. If someone wants to have the closest cone having a green color, then we may want to make some exception. This is why the World Mapping has been introduced in order to be able to change some of the default mappings to more specific mappings for just one instance. In the World Mapping, the user can do this by means of the ConceptMapping and the AttributeMapping.

3.2 Generation

After we have performed the mappings, we are able to generate the code for a working virtual world. The result of this generation is shown in fig. 2.

4 Discussion and Conclusion

In this paper, a new approach for designing VR applications has been described. This approach allows to model a VR application at a conceptual level and in terms of concepts from the application domain. This is realized by using a Domain Ontology, which captures the domain knowledge. The Domain Ontology can be created by a domain expert or using an existing ontology. Using the knowledge in the Domain Ontology and a set of high level modelling primitives for modelling VR (especially developed for our approach and captured in the VR Conceptual Modelling Ontology), a non-VR-specialist can specify the virtual world by instantiating the concepts in the Domain Ontology. This results in an ontology that describes the virtual world one wants to realize. This ontology is called the World Specification. In a next step, the mapping from the conceptual level to the VR implementation environment is specified. For more complex virtual worlds, the VR-specialist may be needed to assist or perform this step. Next, the code for the VR application is generated. This approach is user-centred since the domain expert can be actively involved in the design and development of the VR application. This approach gives advantages as the developed VR application will better fit the requirements of the end-user or the customer; the development process can be shortened; fast prototyping is possible; the communication between the VR-specialist, the domain expert and the

end-user can be improved; and domain knowledge can be exploited. In addition, the use of ontologies allows to incorporate some intelligence and therefore, intelligent reasoning about the virtual world itself becomes possible.

At this moment, our prototype tool is still very basic and does not yet support all the latest features used in VR (like complex behaviours). Also detailed modelling of constraints is not yet included. These features are currently under investigation. Furthermore, the tool will be tested by a number of non-VR-specialists from companies with different domains to validate the approach and the different VR modelling concepts.

We believe that such an approach can open up the use of VR to a much broader community than it is nowadays since it is more domain oriented and it is easier and more intuitive for a non-VR-specialist to design his VR application.

References

1. http://www.unrealtournament.com, Accessed 16 January 2004
2. Carey, R., Bell, G.: The annotated VRML 2.0 Reference Manual. Addison Wesley (1997)
3. Guarino, N., Giaretta, P.: Ontologies and knowledge bases: towards a terminological clarification, In Towards Very Large Knowledge Bases: Knowledge Building Knowledge Sharing. ION Press. (1995) p25-32
4. Gruber, T.R.: A translation approach to portable ontologies. In: Knowledge Acquisition. Elsevier (1993) 5(2) p199-220
5. http://opende.sourceforge.net/ , Accessed 21 August 2003
6. James, S., Willans and Michael, D.: Harrison, A toolset supported approach for designing and testing virtual environment interaction techniques. International Journal of Human-Computer Studies. (2001) 55(2)p145-165
7. Halpin, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. 1st edition, Morgan Kaufmann Publishers (2001)
8. http://www.daml.org/2000/12/daml+oil.daml. Accessed 4 January. 2004
9. Sowizral, H., Rushforth, K. and Deering, M.: Java3D API Specification. Addison-Wesley. (2000)
10. Burdea, G.C. and Coiffet, P.: Virtual Reality Technology. Wiley-Interscience (2003)
11. Kim, G. J., Kang, K. C., Kim, H. and Lee, J.: Software Engineering of Virtual Worlds. Proceeding of the ACM Symposium on Virutal Reality and Technology. (1998)
12. Tanriverdi, V. and Jacob, R.J.K.:VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces. Proc. ACM VRST 2001 Symposium on Virtual Reality Software and Technology. ACM Press.
13. Lassila, O. and Swick, R.: Resource Description Framework: Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium, Cambridge. (1999)
14. Murdock, K. L.:3ds Max 5 Bible. Wiley Publishing, Inc. (2003)
15. Fencott, C.: Towards a Design Methodology for Virtual Environments. in proceedings of a workshop on user centered design for virtual environments, York, England. (1999)