An ontology-based approach for IVE+VA*

Francisco Grimaldo, Fernando Barber, and Miguel Lozano

Computer Science Department, University of Valencia,
Dr. Moliner 50, (Burjassot) Valencia, Spain
{francisco.grimaldo, fernando.barber, miguel.lozano}@uv.es

Abstract. This paper proposes the use of ontologies to provide intelligent virtual environments with semantical information, which can be of great value to the virtual agents inhabiting the world. On the one hand, this approach allows us to define complex objects and to introduce semantic levels of detail that help the agents when sensing complex scenes. On the other hand, the object taxonomy established by the ontology of the world is also used to define general and reusable activities for the roles involved in the simulation. Finally, in order to verify the approach presented, we have developed an application example dealing with a virtual bar, where different actors are able to sense as well as interact efficiently with complex objects (e.g. cabinets, shelves, ...).

1 Introduction

Game narrative [11] and storytelling systems [5] have productively applied planning techniques to generate dynamic and interactive stories. Although these approaches are well known as *knowledge intensive* [5], there is not a common formalism to manage the knowledge associated to the environment. Instead, all that knowledge is generally assigned only to the actors plans. Nevertheless, regardless of the nature of the application (e.g. educational, entertainment, training, etc), the definition of a semantic knowledge base would benefit production, visualization and interaction within Intelligent Virtual Environments (IVE) [1]. Scene graphs are insufficient to specify knowledge associated to virtual entities beyond simple relationships between objects. The need for richer expressiveness has led to Semantic Virtual Environments (SVE) that take into account not only the spatial structure but also high-level semantics.

Animated worlds populated by autonomous agents should also move in this direction in order to increase agent/object interaction. Even though planning based agents are usually able to perform tasks on well-known objects within the IVE, they can hardly reuse their skills. For instance, a real barman has the general ability to serve drinks to customers. However, a corresponding virtual barman would normally define one operator for each of the tasks that the role can carry out. Hence, an agent might be able to serve an orange juice and

* Supported by the Spanish MEC under grants TIN2006-15516-C04-04 and Consolider Ingenio 2010 CSD2006-00046.

a glass of wine while not having any idea about how to serve a cup of tea. This happens because virtual agents do not actually have general but particular understanding about the situations they face. Semantics can enrich planning in knowledge-rich application domains [8], however, SVE systems have normally focused on the construction of the environment [14][10] instead of dealing with interaction performed by virtual agents inhabiting the world.

In this paper we present a SVE approach which uses ontologies to enhance both agent-object interaction and manageability of complex objects such as containers (e.g. a tray with glasses, a cabinet containing dishes, shelves with bottles on top, etc). The next section expounds the main approaches related to the usage of ontologies in virtual worlds. Section 3 introduces the architecture proposed for our SVE and section 4 deals with the ontologies employed. We present an interactive Semantic Layer that controls sensing and actuation during simulation in section 5. Section 6 explains how plan-based virtual agents can use semantics to enhance their operativity. Finally, section 7 shows an application example in order to illustrate the goals of the proposal.

2 Related Work

The use of ontologies as a means to develop virtual worlds is becoming familiar within the graphics community. In VR-Wise [15] a non virtual reality specialist can design a virtual environment using a set of DAML+OIL ontologies. Afterwards, the system can map concepts of a world domain onto VRML graphical representations that can be used to render the scene. The SeVEn platform [14] supports SVE based on W3C Resource Description Framework (RDF). Behind this approach is the idea of developing system-independent software that can be reused over several virtual environments. Objects are categorized using an explicit type field that is passed to the SVE software. Depending on this field, task-relevant information about an object can be extracted. The semantic model presented in [10] defines the abstract concept of virtual entity. It includes the geometry and user interface of objects in a virtual scene and it is mainly used to visualize and interact with the items in different contexts/devices. Additionally, a preliminary version of an ontology for virtual humans is also introduced. This XML-based ontology is aimed at modelling and animating the human body as well as at the interaction of virtual humans with Smart Objects [12]. Interaction is then governed by the environment since all the information relative to it is stored in the objects themselves.

Several research efforts deal with the retrieval and management of the information represented in the ontology. Synoptic Objects in STARFISH [2] create an informed environment [7] which allow autonomous agents to interact with the objects with a certain freedom. The agent's decision cycle is not treated, though. Another interesting use of semantics is shown in [16], where the separation of actions and consequences brings interoperativity and reusability to virtual entities. The authors, however, do not propose a formalized ontology that can be used to build and manage virtual worlds.

From the agents community, "Kwnowlege in the World" Doyle's idea [6] proposed annotated environments as a way to allow believable agents to act across different worlds. A semantically enhanced approach attain the integration of environmental concepts into the agent decision-making [4]. These semantic concepts are supplied from an ontology-based cognitive middle layer between agent minds and the environment. Actions are also represented in this layer through causal rules whose effect is turning the target object into an instance of another concept. Although this schema allows inference using a situated backward-search planning algorithm it has several expressiveness limitations. For instance, the effect of an action is restricted to one single object and always implies placing the affected object as an instance of another different qualitative concept. Sometimes, though, actions can affect multiple objects or change an object property without modifying any concept. An example could be *to fill up a glass using a bottle of wine*. In such an action the quantity of liquid contained in both objects should be modified but they should remain instances of the same concepts.

3 An ontology-based Semantic Virtual Environment

In this paper we present a SVE approach that uses ontologies as an appropriate basis to animate virtual environments inhabited by intelligent virtual agents. Figure 1 shows the architecture of our multi-agent framework, which can be divided into several parts:

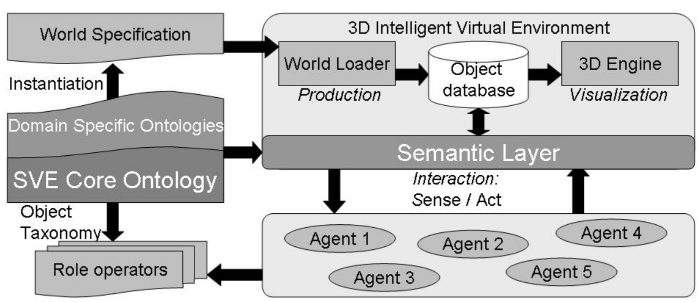

Fig. 1. Ontology-based SVE architecture

– Ontologies define the world knowledge base. We distinguish two levels of representation: the *SVE Core Ontology* is a unique base ontology suitable for all virtual environments which can be extended by different *Domain Specific Ontologies* in order to model application-specific knowledge (see figure 2). Then, environments can be constructed by instantiating the classes of these ontologies in a *World Specification* file. For example, in section 7 we will create a virtual bar with a large of objects (e.g. bottles, glasses, ...).

– The *Semantic Layer* is the interface between the agent and the world (i.e. the sense/act manager). When sensing complex objects (e.g. a tray with 20 glasses, spoons, etc.), it uses the ontology to reduce the information flow (as section 5 shows). Besides, this layer is in charge of executing the actions requested by the agents as well as maintaining their semantic effects.
– Planning based agents receive sensorial information from the *Semantic Layer* and calculate the appropriate sequence of actions in order to achieve their goals. In this context, the object taxonomy defined in the ontology is also used to generalize operators, which, in turn, will increase both the number of planned interactions and the interoperativity of the agents across different scenarios (see section 6).

Finally, a *3D Engine* can extract graphical information from the object database thus performing visualization (which is out of the scope of this paper).

4 SVE Core Ontology and Domain Specific Ontologies

Ontologies in our SVE define the abstract world model, that is, the hierarchy of classes as well as their properties and possible interrelationships. We have adopted the Web Ontology Language (OWL) [18] to implement them. In particular, the *OWL DL* sublanguage, as it provides maximum expressiveness without losing computational completeness.

The *SVE Core Ontology* defines the basic classes needed to create any virtual environment and allows us to use inheritance to define the object taxonomy. According to this, figure 2 shows a fragment of the core ontology[1] focused on the classes inherited to represent different *Container* objects. Here, containers of uncountable substances (e.g. a pot with salt) are distinguished from those that hold countable elements. Among them, *ServiceContainers* provide services of entities with no associated visual model (e.g. a bottle of whisky) while *ObjectContainers* contain graphical objects. Moreover, *HeterogeneousObjectContainers* such as a cabinet with different items (e.g. books, bottles, etc) are differentiated from *HomogeneousObjectContainers*, which only contain undistinguishable objects (e.g. a shelf with clean glasses). Figure 2 also shows the properties shared by all *Containers*, both quantitative (e.g. *location* or *height*) and qualitative/semantic (e.g. *empty* or *full*). The core ontology also defines the set of possible relations for *MovableObjects*: currently, *in*, *on* and *pickedBy* (see figure 3).

The *SVE Core Ontology* provides a high-level classification, however, certain application domains contain objects with special traits. Then, a *Domain Specific Ontology* can be used to extend the core ontology and to represent new classes in a particular scenario. This way, in the virtual bar example (see section 7) we can define classes such as *Bottle*, which inherits from *MovableObject* and *ServiceContainer* (see figure 2).

[1] Protege [17] has been used to develop the ontologies.

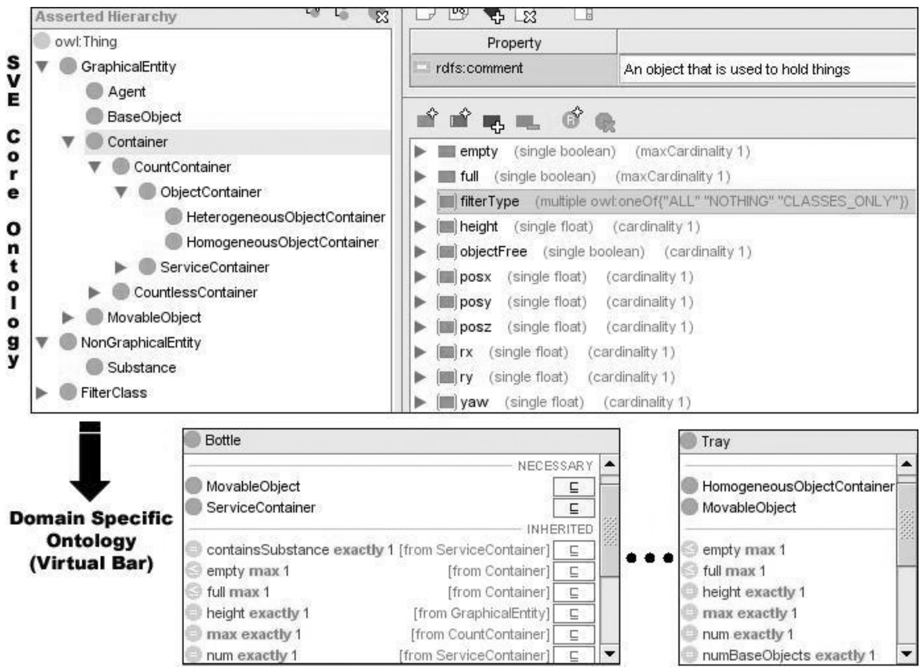

Fig. 2. SVE Core Ontology and Virtual Bar Ontology

5 Interactive Semantic Layer

The *Semantic Layer* in our SVE uses the ontology to properly manage agent-object interaction during simulation. It is mainly concerned with two issues: sensing and actuation.

Knowledge-rich environments can be hard to sense for interactive planning agents (e.g. a shelf in a bookstore contains too much information for any planning search state). To properly balance expressiveness and manageability, this layer uses a tree to hierarchically extract the scene state (see figure 3). Links in this tree represent sensorial dependencies between objects which are modelled through *senseDepends* relations. *SenseDepends* has been modelled as an OWL superproperty defined in the *SVE Core Ontology*. The range associated to this superproperty is the superclass *FilterClass*, which filters the information according to the value of the property *filterType*. This property is configured initially and it can be changed dynamically. For instance, relation *in* relates objects with containers, so, each *Container* can manage the information that is sensed about the contained objects. Currently, we have implemented three kinds of *FilterClass* filters: *ALL*, *NOTHING* and *CLASSES-ONLY* (see figure 2). Any information can pass through the *ALL* filter whereas the *NOTHING* filter blocks everything. On the other hand, an object with the *CLASSES-ONLY* filter only publishes the classes of their subordinates. These filters allow the definition of semantic levels of detail in complex objects, for instance, a cabinet can be modelled as an *HeterogeneousObjectContainer* that only publishes the classes of the objects inside while it is closed. Additionally, the *Semantic Layer* implements an specific sensorial behavior for *HomogeneousObjectContainers*. Within them, objects are supposed to be undistinguishable, thus, this kind of containers can only publish the information about a reduced number of k interactive objects [2]. This way, interaction with these objects is guaranteed at anytime while the amount of information is lowered. In the example of the virtual bar this behavior will be applied to a shelf with one hundred clean glasses (see section 7).

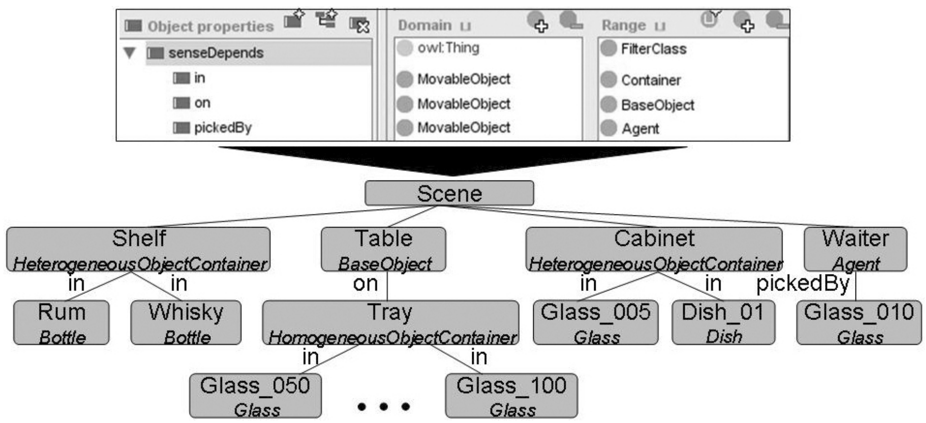

Fig. 3. Sensorial dependency tree

The *Semantic Layer* also executes the actions requested by the agents. It supports the procedures that check action success/failure and apply the effects to the objects involved. Thus, actions can change object relations, quantitative properties and also semantic properties. In our virtual bar, for example, when a waiter picks one glass from a shelf the action also decreases the number of glasses contained in the shelf and changes the *empty* property to true if the shelf contains nothing.

6 Actors operativity

Goal oriented virtual actors need an action scheme to plan their tasks and to achieve their goals. Planning based agents are normally inspired in the STRIPS (ADL/PDDL) action language, where activities are represented using a classic state model with a conjunction of grounded literals. This approach has been used to model knowledge intensive 3D environments, such as storytelling domains ([5]) and other task-oriented 3D agents ([13][9]). In these contexts, however, the usual

[2] Where k is also a dynamically configurable object property.

representation of the actions suffers from different drawbacks when an actor, or a group, is interested in reusing their operators with different objects. Thus, their activity should be reviewed so it incorporates semantic information from the ontology, which represents the abstract world model for the actors.

In this paper, we use the object taxonomy to define general and reusable operators. Figure 4 shows the definition of two frequent actions for a waiter in the virtual bar. On the left, *PickFromContainer* interacts with any movable entity that is placed in any object container. The parameters section contains the objects involved, and the *preconditions*, *add* and *delete* lists follow normal STRIPS assumptions. The operator *ServeFromContainer*, coded on the right, can be used to pour any *Substance* contained in a *ServiceContainer* into a *CountlessContainer*. Therefore, it allows serving both ices from an ice container and a donut from a tray, providing they are instances of the class *ServiceContainer*. Here, *not* expressions are also permitted to represent negative preconditions over the current search state.

```
operator PickFromContainer:
    parameters:
      ?object - MovableObject
      ?container - ObjectContainer
    preconditions:
      ?object in ?container
    add:
      ?object pickedBy ?me
    delete:
      ?object in ?container
```

```
operator ServeFromContainer:
    parameters:
      ?glass - CountlessContainer
      ?container - ServiceContainer
      ?substance - Substance
    preconditions:
      ?glass pickedBy ?me
      not(?glass full)
      not(?container empty)
      ?container containsSubstance ?substance
    add:
      ?glass containsSubstance ?substance
    delete:
      ?glass empty
```

Fig. 4. General and reusable operators

This schema, which basically introduces variables and types referred to the hierarchy of classes, avoids the definition of an operator for each object, which finally reflects in a higher degree of interaction. Furthermore, agent interoperability between different scenarios is enhanced since they will be able to manage any object of the world provided that it is an instance of a class defined in the ontology.

7 Application example

In order to test the techniques previously explained we have developed an application example dealing with a virtual bar. Figure 5 shows the constructed scenario, where different waiters attend orders made by customers. We have used the ontologies introduced in section 4 to properly classify the interactive objects present in the world. For instance, the class *ServiceContainer* has been used to model a coffee machine, a tray with donuts and also an ice container. Similarly, several objects are represented as instances of the class *HeterogeneousObject-Container*, among them *shelf-B*, a shelf with alcoholic bottles on top. A domain specific ontology has been implemented that defines common objects in a bar such as *Bottles*, *Trays*, *Dishes* and so on.

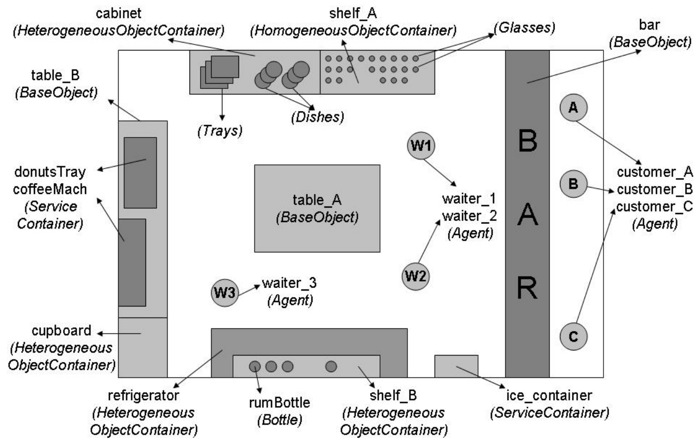

Fig. 5. Virtual bar scenario

The class of an object is used by the *Semantic Layer* to enhance sensorization. For example, we have modelled a shelf with one hundred clean glasses (*shelf-A*) as an instance of the class *HomegeneousObjectContainer*. A full representation of it would easily inundate any search state, however, its sensorial behavior summarizes the necessary information during the simulation. Hence, although it informs about the total number of glasses on top, it only gives a full description of a reduced number of glasses (in our scenario, three available glasses) [3]. *HeterogeneousObjectContainers* such as the *refrigerator*, the *cabinet* or the *cupboard* have been configured with the *CLASSES-ONLY* filter type to only publish the classes of the objects inside while they are closed. Thus, we guarantee reasonable size states for the intelligent virtual agents involved.

Agents within this scenario can play one of two possible roles: waiters or customers. Each role has its own set of operators, used to plan the agent's activity. In our virtual bar, we have employed heuristic search plan-based agents that interleave sense, plan and act to animate its behavior in dynamic environments ([13]). Figure 6 shows the set of operators available to waiters. Parameters match the objects of the environment according to their classes. Thus, agents can generalize their operators and reuse them to satisfy different goals (e.g. serving breakfast or spirits to a customer). For example, the operator *ServeFromCon-*

[3] This number is configured as an object property when the shelf is created.

tainer is used in five different contexts: serving a donut onto a dish, pouring coffee into a cup, putting ices into a glass and pouring rum or coke into a glass.

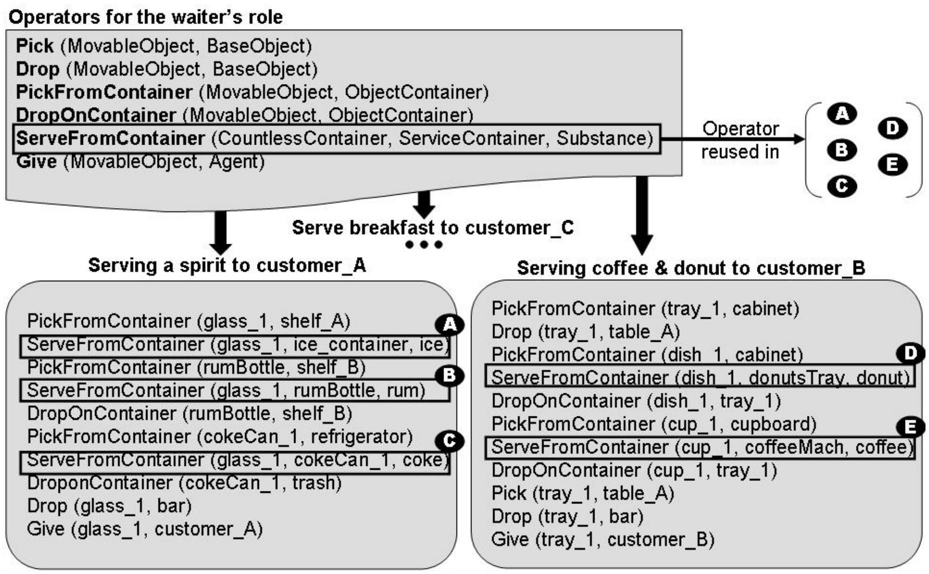

Fig. 6. Example of operator reusability

8 Conclusions

Semantics can be very useful in knowledge intensive environments such as games or storytelling systems. In this paper, we have presented a Semantic Virtual Environment that uses ontologies as an adequate model to handle the knowledge of the world, which can be necessary for planning based agents.

Firstly, the sensorial dependency tree [4] introduced in the *Semantic Layer* helps the agents to manage knowledge-rich environments, which are normally hard to sense for them.

It is also important to realize that the object taxonomy extracted from the ontology allows the actors to reuse their operators in different contexts. This will finally help us to create autonomous agents with a higher degree of interaction, as occurs with the waiter role previously explained. This example demonstrates how an agent can reuse or instantiate the same operator in different plans to obtain different desired effects depending on the objects involved.

The approach presented above is part of the work in progress in order to develop cooperative roles for virtual agents within Intelligent Virtual Environments.

[4] Similar to a semantic network.

References

1. Aylett, R. and Luck, M. Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments. *Applied Artificial Intelligence*, 2000.
2. Badawi, M. and Donikian, S. Autonomous Agents Interacting With Their Virtual Environment Through Synoptic Objects In *Proceedings of the 17th conference on Computer Animation and Social Agents (CASA2004)*, 2004.
3. Cavazza, M., Charles, F. and Mead, S.J. AI-based Animation for Interactive Storytelling. In *Proceedings of Computer Animation*. IEEE Computer Society Press. Pages: 113–120. 2001.
4. Chang, P. H-M., Chien, Y-H., Kao, E. C-C. and Soo, V-W. A knowledge-based scenario framework to support intelligent planning characters. In *Proc. of the 5th International Workshop of Intelligent Virtual Agents (IVA05)*. Springer LNCS. Vol: 3661. Pages:134–145. September 2005.
5. Charles, F., Lozano, M., Mead, S.J., Bisquerra, A.F. and Cavazza, M. Planning Formalisms and Authoring in Interactive Storytelling. In *1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.
6. Doyle, P. Believability through context using "knowledge in the world" to create intelligent characters. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems(AAMAS02)*. ACM Press. Pages: 342–349. July 2002.
7. Farenc, N., Boulic, R. and Thalmann, D. An informed environment dedicated to the simulation of virtual humans in urban context In *Eurographics 1999* Volume 18. 1999.
8. Gil, Y. Description Logics and Planning. In *AI Magazine*. 2005.
9. Grimaldo, F., Lozano, M., Barber., F. and Orduña J.M. Integrating social skills in task-oriented 3D IVA. In *Proceedings of the 5th International Working Conference on Intelligent Virtual Agents*. Springer-Verlag LNAI. September 2005.
10. Gutierrez, M. A. Semantic Virtual Environments. *PhD thesis*. April 2006.
11. Harris, J. and Young, M. Proactive Mediation in Plan-Based Narrative Environments. In *IVA'05*. Springer LNCS. Vol: 3661. 2005.
12. Kallmann, M. and Thalmann, D. Direct 3d interaction with smart objects. In *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*. ACM Press. Pages: 124–130. 1999.
13. Lozano, M., Grimaldo, F. and Barber, F. Integrating miniMin-HSP agents in a dynamic simulation framework. In *Proceedings of the 3rd Hellenic Conference on Artificial Intelligence*. Springer-Verlag LNAI, May 2004.
14. Otto, K. A. Towards semantic virtual environments. In *Workshop Towards Semantic Virtual Environments (SVE'05)*. Pages: 47–56. 2005.
15. Pellens, B., Bille, W., De Troyer O. and Kleinermann, F. VR-wise: A conceptual modelling approach for virtual environments. In *Methods and Tools for Virtual Reality (MeTo-VR 2005) workshop*, 2005.
16. Soto, M. and Allongue S. Modeling methods for reusable and interoperable virtual entities in multimedia virtual worlds. *Multimedia Tools and Appliclations* Volume 16. Pages:161–177. 2002.
17. Stanford Medical Informatics. Protege. *Available at http://protege.stanford.edu/*, 2006.
18. W3C. OWL web ontology language guide.
Available at http://www.w3.org/TR/owl-guide/, February 2004.