



**HAL**  
open science

# An Ontology for Semantic Representation of an Urban Virtual Environment

Kenza Harkouken, Nicolas Sabouret, Jean-Yves Donnart

► **To cite this version:**

Kenza Harkouken, Nicolas Sabouret, Jean-Yves Donnart. An Ontology for Semantic Representation of an Urban Virtual Environment. The International Conference on Artificial Intelligence, Jul 2013, Las Vegas, United States. hal-00903983

**HAL Id: hal-00903983**

**<https://hal.science/hal-00903983>**

Submitted on 13 Nov 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Ontology for Semantic Representation of an Urban Virtual Environment

K. Harkouken Saiah<sup>1</sup>, N. Sabouret<sup>2</sup>, and J-Y. Donnart<sup>3</sup>

<sup>1</sup>LIP6, UPMC, Paris, 75005, France

<sup>2</sup>LIMSI, CNRS, Orsay, 91403, France

<sup>3</sup>THALES Training & Simulation / Cergy-Pontoise, 95525, France

**Abstract**—We present in this paper a model of semantic representation of a dynamic virtual environment. Our model is embodied into a simulation architecture of a virtual city. The objective is to enable agents to make action decision consistent with the semantic state of their environment. The idea is to represent the services offered by the environment into an ontology of services and deduce the available object that has the best quality of service at any given time of the simulation. For this, we defined a layer of the ontology containing the general concepts of representation of the environment. This representation allows us to apply a unified model of reasoning to infer elements of knowledge useful for decision-making agents.

**Keywords:** Knowledge Representation and Reasoning, Semantic Virtual Environments, Ontology, Urban Simulation

## 1. Introduction

Knowledge representation for virtual environments (VE) became an important issue with increasing interest in the last decade [1] [2]. Indeed, the agents operating in VE need semantic informations on it to adapt their behavior to identifiable elements of that environment and making a choice of action [3]. Semantic Virtual Environments (SVE) [4] have been proposed to address this lack of semantic informations (environmental knowledge) necessary to support intelligent interactions between agents and their environment. In fact, SVE provide a coherent representation of (i) the simulated world and the behavior of its entities, (ii) the interactions and tasks that users and agents can perform in the environment, and (iii) elements of knowledge that the agents can use to make decisions [5].

The work carried out for the representation of VE are divided into two categories corresponding to two levels of semantics.

The first category of these works aims at an explicit representation of the contents of the environment to reproduce the scenes. Therefore, the agents have direct access to the definitions of entities of the environment. These representations are mainly geometric and topological information of the environment. Several approaches have been proposed for this representation. Most of them offer an

ontological representation [6] containing different types of environmental information for different types of application.

Otto [4] uses domain ontologies to create virtual environments in a platform he called "SeVE". The goal is to find a representation of the environment independent from the software description to create virtual objects that can be reused in several virtual environments. This reuse is possible by adding information relevant to the tasks of an object in the field of virtual environments generation. Following the same principle, [7], [8] propose to generate a virtual world from an ontology. It contains a description of the objects and their properties defined by domain experts.

Kalogerakis et al. [9] have defined an ontology "X3DOntology" associated with entities defined in description files of 3D virtual worlds. This provide a semantic representation of the scene to infer its content. We can find the same approach in [10] but at a higher level, making the correspondence between the objects of the virtual environment and the concepts of the ontology.

These works give agents direct access to descriptions of VE entities. However, this definition of the world doesn't allow them to deduce information for decision support to identify (i) the object that allows them to perform an action, (ii) what actions can be performed on an object (iii) the relationship between two objects, (iv) and the relationship between actions.

The second category of works on semantic representation of VE has been made to strengthen the agent-object interactions that include not only information related to the geometry of the scene, but also knowledge about the possible interactions with the objects in the environment [11] (what can be done? how this can be done? why and how it should be done?).

In [12] and [13], the authors have extended the principle of using ontologies to represent SVE incorporating the semantic description related to the function of objects so that agents can design their own animation procedures like path planning.

Kallmann & Thalmann [14] have proposed *Smart Object* whose idea is to include in the description of the object all the required information to describe how to interact with it. They distinguish information on: (i) the properties of the object (semantic and physical), (ii) how to interact with

the object (actions, positions, gestures), (iii) the behavior of the object in response to an action and (iv) the behavior of actors to achieve interaction. Following the same philosophy, Badawi [15] proposes Synoptic Objects *STAR FISH*. The idea is to define a minimum set of primitive actions which are used to build complex actions.

In a lower level of representation [2] they developed a meta-model to represent the semantic of the VE and its structural properties, geometrical, topological, the behavior of agents and interactions between agents and users.

These works propose a semantic representation of VE which offers data for decision-making of agents. This give them the ability to interact with the VE and avoid having aberrant behaviors (that never occur in real life). But these solutions do not give agents the elements of knowledge to make *the best choice* of action for a given situation. They can provide the utility and function of an object of the environment but agents should make the connection with their actions plan to decide if it suits them or not.

Instead, we propose in this paper a method that avoids agents to find which element of the environment will allow them to carry out their plan. The idea is to provide agents with a sorted list of objects according to their relevance regarding their actions plans. Our goal is not to influence the decisions of agents or decide for them, but we aim to provide them with knowledge they need to make a decision consistent with the context. The goal of our approach is to decentralize and capitalize calculations made by agents by strengthening communication links between the decision-making of the agents and the representation of their environment. Our choice is justified by the complexity of the calculations for the decision and the large number of agents contained in the SVE.

Published works with similar objectives, propose a representation of the environment that integrates the description of possible actions that agents can perform. In this case, the interactions between the elements of the simulated world and the agents are managed by their environment and are related to what is defined in it. However, the heterogeneity of SVE (different categories of its elements) increases the size of the representation and affects the calculations cost to be performed by agents.

Our first contribution is to overcome this problem by proposing a unified representation model of the environment which allows us to do a generic calcul of decision support elements (measure of relevance between the action of the agent and function given by an element of the environment). The idea is to have simple calculations and applicable to all elements of the environment by treating uniformly requests of agents. Our study focuses on modeling the semantics of a virtual urban environment which has the distinction of being complex and heterogeneous, ie, it is composed of several types of objects with different features. The objective is to propose a model to represent the semantic of the

environment allowing us to apply a unified treatment for all its components.

Our second contribution is to represent the types of objects that make up the environment by services (services offered by objects of the environment), where each service of the environment will be evaluated according to the type of object that proposed it. We were inspired by the techniques of semantic web [16] [17] by defining the elements of the environment in a *Ontology of Services*. We consider that each object type offers one or more *services* with some *quality*. The Quality of Service is derived from the unified reasoning we have applied on our representation model and it will be used by agents to make their decision.

In the remainder of this article, we present the architecture on which we are working, and then we describe the representation model we proposed. Finally, we discuss the first results of the implementation of this model and perspectives of our work.

## 2. Architecture

Our representation module of the environment is part of a platform of the project *Terra Dynamica*<sup>1</sup>, which is a simulation project of a virtual dynamical city. The architecture of the project consists of several modules interacting with each other (semantic representation of the environment, decision-making, affective, path-finding and patrol).

As shown in figure 1, the environment semantic representation module interacts with the decision-making module of the agents. These interactions are summarized in an exchange of requests / responses between the two modules. The decision-making module queries the semantic representation module so that it offers him the environment entities enabling to define an action plan for the agents or to choose the best element of the environment allowing them to make an action. For the latter case, we distinguish two situations (i) agents that want to achieve immediate action (which they consider very important) and (ii) agents that want to perform a task plan in an opportunistic manner (when a opportunistic situation occurs).

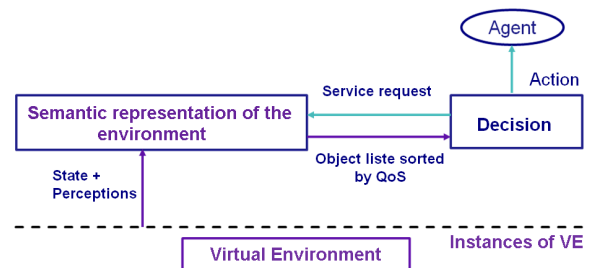


Fig. 1: System architecture: interactions between semantic module representation of the VE and the decision-making module of the agents.

<sup>1</sup><http://www.terradynamica.com/>

We therefore considered three types of interactions between the semantic representation module and the decision-making module :

- **Interrogative mode:** during the planning phase the semantic representation module is requested to provide information on the environment to make a plan of action. The semantic module responds to these requests by providing the elements of the environment allowing to achieve the action plan in line with the context of the environment (Query: what allows me to eat, Answer: a restaurant, sandwich, etc.).
- **Reactive mode:** it occurs during the carrying out of the agents plan. In this case, the agents know what element of the environment they want to use (defined during the planning step). When they are preparing to use a service provided by this element, a request is submitted to the semantic module to return the quality of service in relation with the context on one hand and action on the other hand.
- **Pro-active mode:** agents can submit an application of interest for a given service. The semantic module takes account for propose the elements of the environment providing the service when they arise.

We also distinguish two types of queries sent by the decision-making modules of the agents:

- Request to obtain a given service (for interrogative mode and pro-active mode),
- Request on the consistency to use an element of the environment for obtaining a given service (for reactive mode).

We will see in section 4 how these queries are implemented.

### 3. Model

Our representation model of the environment is based on a two-level ontology that provides support for a generic mechanism for reasoning about the dynamic and functioning of the simulated environment (see Fig. 2).

#### 3.1 Ontology of Services

In our model, an ontology is a pair  $\mathcal{O} = \langle \mathcal{C}, \mathcal{R} \rangle$ , where  $\mathcal{C}$  is the set of concepts and  $\mathcal{R}$  is the set of binary relations. The ontology of services is not limited to relations of hierarchy (*isa*) or meronymy (*partof*), but we also defined specific relationships with our goal of representation.

The generic level of our ontology of services contains general knowledge of environmental concepts  $\mathcal{C}$  which consists of three subsets  $\mathcal{C} = S \cup O \cup C_r$  where:

- $S$  the set of all services,
- $O$  the set of all types of objects in the environment,
- $C_r$  collects the evaluation criteria of the quality of service rendered by an object type.

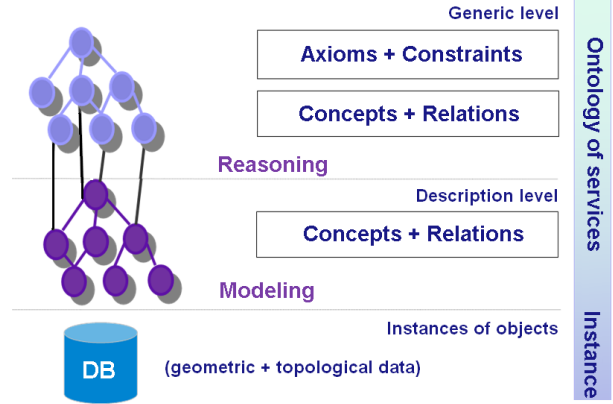


Fig. 2: Representation model of the environment: Ontology of services

#### Definition of object types

An object type is an element of the environment providing one or more services. Each object type provides a service with a certain quality (rendering service).

#### Definition of services

A service is a tuple  $\langle name, C_r(s) \subset C_r, O(s) \subset O \rangle$  such that  $name$  is the service name,  $C_r(s)$  the set of criteria for evaluating service  $s$  and  $O(s)$  the set of object types providing this service.

We have defined the relationship  $offers(o, s) \in O \times S$  between services and object types. It allows us to say that one object type  $o$  offers a service  $s \in S$  such that  $o \in O(s)$ . For example, the service *Eat* can be offered by several types of objects  $O(Eat) = \{Restaurant, Fast Food, Cafeteria, etc\}$ , this service will be evaluated according to the criteria  $C_r(Eat) = \{Cost, Time, Quantity, Quality, Ambiance\}$ .

We have defined two hierarchy levels of service  $S_a$  and  $S_b$ , as we can see in Fig. 3.

- Behavior service  $S_b$  : set of services allowing to have a behavior and to respond to a motivation,
- Action service  $S_a$  : set of services allowing to make an action.

Such as  $S = S_b \cup S_a$  and  $S_b \cap S_a = \phi$ .

This two types of services are related by the relation  $Implements(s, s') \in S_a \times S_b$ , which means that a behavior service can be obtained (implemented) by one or more action service. For example, the service  $s = Eat \in S_b$  is a behavior service which can be achieved by different ways (implemented with different action services)  $s_1 = have a snack$ ,  $s_2 = Nibble$ , etc, with  $s_1, s_2 \in S_a$ .

For each behavior service there is at least one action service that implements it  $\forall s \in S_b, \exists s' \in S_a$  such that  $implements(s', s)$ , where a service action is related to a single service behavior  $\forall s'' \in S_a, \exists! s \in S_b$  such that  $implements(s'', s)$ .

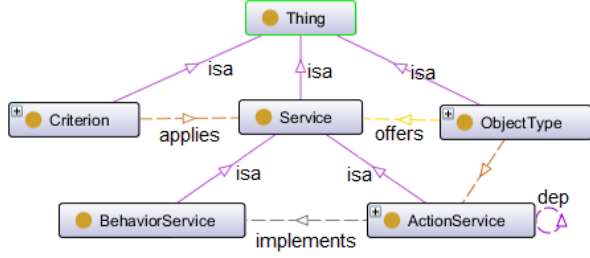


Fig. 3: General concepts of the ontology of services.

Note that the objects types only offer actions services  $\forall o \in O, \forall s \in S, o \in O(s) \Rightarrow s \in S_a$ .

We have defined a non-hierarchical relation *dependency*  $dep(s, s')$  between two services  $s$  and  $s'$  proposed by the same object type, such as,  $dep(s, s') \in 2^S$ , with  $o \in O(s)$  and  $o \in O(s')$ .

For example, the service *Drink* proposed by the *Restaurant* depends on the service *eat* proposed by the same object *Restaurant* because we can't use the service *Drink* of the restaurant if we don't intend to *eat*.

We use the dependency relationship between the services in calculating the quality of service (section 3.2.1).

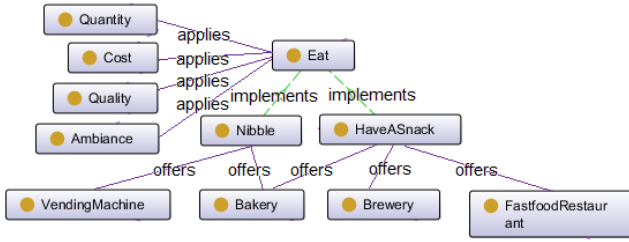


Fig. 4: Ontology of services : example of service "Eat"

### Definition of criteria for evaluating services $C_r$

Services are evaluated according to a set of criteria. A criterion can be shared by multiple services, such as *Cost*, or it can be specific to a service, such as *quality of food* which is reserved for catering services.

Let  $C_r$  the set of criteria. Each service  $s \in S$  will be assessed according to a subset of  $C_r$  which we denote  $C_r(s) \subset C_r$ .

The set of criteria  $C_r(s)$  of an action service  $s \in S_a$  is a subset of criteria  $C_r(s')$  of behavior service  $s' \in S_b$  it implements ( $\forall s \in S_a, s' \in S_b, implements(s, s') \Rightarrow C_r(s) \subseteq C_r(s')$ ).

We have defined a non-hierarchical relation *applies* between services and criteria, such that  $applies(c, s) \in C_r \times S$ , which means that the criterion  $c$  is *applied* to evaluate the service  $s$  (see Fig. 4).

We have defined for each criterion  $c \in C_r$  a set of values denoted  $V(c)$ . This set contains all possible values of a given

criterion  $c$ . We have for example the values  $V(cost) = \{expensive, not\ cher, free\}$ .

## 3.2 Quality of Service

Quality of Service  $QoS$  is calculated thanks to the following parameters:

- **Effectiveness**, to assess whether an environmental element meets the needs of the agent and allows it to perform its task.
- **Difficulty**, calculate the induced cost by additional constraints to access the desired service.
- **Proximity**, measure the accessibility of a service based on the abundance of elements in the environment that offer the desired service.

Note that the  $QoS$  is independent of agents preferences (we have set one type of calculation for any object type or service).

We note that the decisional module of agents take into account other information about the environment and the state of the simulation in addition to the  $QoS$  for making a decision (geometric and topological data extracted from the DB, distance from an object, the presence of a queue, etc.).

### 3.2.1 Parameters for calculating the QoS

#### Effectiveness

The effectiveness of an object type  $o \in O$  offering a service  $s \in S_a$  searched by an agent is the ratio between the number of criteria satisfied with the service provided by the object type and the total number of evaluation criteria of service  $s$ .

To do this, we define the following sets:

Let  $QC(o, s) \subseteq C_r(s)$  the set of criteria that apply to assess the service  $s$  proposed by an object type  $o \in O$  such that  $s \in S_a, o \in O(s)$  and  $\exists offers(o, s)$ . For example, the service  $s = have\ a\ snack$  proposed by the object type  $o_1 = Vending\ machine$  have the set criteria as  $QC(o_1, s) = \{Cost, Time, Quality, Quantity\}$  and the type of object  $o_2 = Fast\ food\ restaurant$  offering the same service  $s$  have the set criteria as  $QC(o_2, s) = \{Cost, Time, Quality, Quantity, Ambiance\}$ .

We have defined a set of values on the criteria of the service  $s$  we noted  $V_q(s, c) \subseteq V(c)$  such that  $c \in C_r(s)$ . The set  $V_q(s, c)$  contains all possible values that can take a service  $s$  for one of its criteria  $c$ . For example, the criterion  $c = Quantity$  for the service  $s = have\ a\ snack$  will have a value in the set  $\{Medium, Small\}$ .

We have defined a set  $EQ(s)$  (*Estimated Quality*) containing all possible combinations of pairs (criterion, value) for a service  $s$  with  $EQ(s) = \{(c, v) \in C_r(s) \times V_q(s, c)\}$ . In the previous example,  $EQ(have\ a\ snack)$  contains all  $\{(Quantity, Small), (Quantity, Medium)\}$ .

Let  $EQ(o, s)$  the set of effects expected by the use of an object type  $o \in O$  to obtain a given service  $s \in S_a$  with  $EQ(o, s) = \{(c, v)/c \in QC(o, s) \wedge v \in V_q(s, c)\}$ .



For example, the service  $s = \textit{have a snack}$  proposed by the object  $o = \textit{Vending machine}$  will result in  $EQ(o, s) = \{(Cost, Not\ expensive), (Time, Speed), (Quantity, Small)\}$ .

Efficiency  $Eff(o, s)$  of the object type  $o$  relative to the requested service  $s$  is therefore calculated as follows:

$$Eff(o, s) = \frac{|EQ(s) \cap EQ(o, s)|_2}{|QC(o, s)|}$$

### Difficulty

The difficulty parameter is based on additional conditions to be added to obtain a service. We consider that an object type offers a service with a difficulty when it depends on another service offered by the same object type. The difficulty  $Diff(o, s)$  of the service  $s$  offered by the object type  $o$  and having a relationship of dependence ( $dep(s, s')$ ) with another service  $s'$  is the rate of criteria in addition to consider (criteria of  $s'$ ) for obtain service  $s$ .

$$Diff(o, s) = \frac{|C_r(s')| - |QC(o, s) \cap QC(o, s')|}{|C_r(s')|}$$

The difficulty is not taken into account in the case where the agent had planned an action corresponding to the service  $s'$  in its original plan. In this case, the solution to have both services will be considered an optimized solution.

### Proximity

We measured the proximity of a service based on the attendance of the object type that offer this service. Our goal is to enable agents to tell the difference between one object type they can find very often in the environment, making it more accessible and less difficult to find, and an object type not very common in the environment and may be the only way for getting given service.

Let  $o \in O$  an object type providing the service sought  $s$ . We defined the set  $O_p(o)$  containing the instances of objects with the object type  $o$ . Let  $O_s(s)$  the set of instances of the object types  $o \in O$  offering the service  $s$ .

The proximity  $Prox(o, s)$  of an object type  $o$  offering the service  $s$  will be calculated as the ratio between the number of its instances, and the number of instances of object types that offer same service  $s$ :

$$Prox(o, s) = \frac{|O_p(o)|}{|O_s(s)|}$$

#### 3.2.2 calculating the QoS

The quality of service offered by an object type of the environment takes into account the three parameters we mentioned above, with a variant on the weight of their importance. For example, an agent may prefer to have an object type which provides a service with maximum efficiency

regardless of the means to do so (difficulty parameter) or difficult to find (proximity parameter).

We calculated the  $QoS$  by a weighted sum of three parameters as follows:

$$QoS(o, s) = \frac{p_1 Eff(o, s) + p_2 Diff(o, s) + p_3 Prox(o, s)}{\sum_i^3 p_i}$$

In the implementation of our model (next section), we assumed that all parameters have the same importance weight of ( $p_1 = p_2 = p_3 = 1$ ).

## 4. Implementation and first results

We have implemented our model in the platform development of the *Terra Dynamica* project (see Fig. 5), developed in C++. We built our ontology using the Protégé<sup>3</sup>. Access to the ontology is done by sending SPARQL queries<sup>4</sup>. These requests are processed according to the three modes of interaction defined above (interrogative, reactive, pro-active).

Requests sent by the decisional module contain the name of the service desired by the agent. These are transformed into SPARQL queries to return a set of object types offering this service. The principle of the research of object types offering a service is to explore the hierarchy of downwards services if we have a behavior service as an input, this will allow us to define the set of services that implements it (knowing that an object type is connected with an action service). Once all the services that can satisfy the agent are found, we will ask our ontology to infer all object types that allow us to have these services.

All object types that returned from querying our ontology will be sorted according to the  $QoS$  we have calculated for each object type. Finally, the decision module selects an instance of one of these object types with the best  $QoS$ .



Fig. 5: Screenshot of the *Terra Dynamica* simulator.

The first results show that the actors of the simulation exhibit a good reactivity to the context. We have seen

<sup>2</sup>We denote by  $|X|$  the number of elements of the set  $X$

<sup>3</sup><http://protege.stanford.edu/>

<sup>4</sup><http://www.w3.org/TR/rdf-sparql-query/>

actors who were heading breweries and bakeries to satisfy their hunger motivation (interrogative mode). We can also see actors throw something in the garbage and continue their activity (pro-active mode). And finally, we found that actors used the objects services when it was possible for the reactive mode (no queue on the same object while another object type is available a little further).

## 5. Conclusion

We presented in this paper a model for representing the semantic of a complex and heterogeneous virtual environment (virtual city). We propose an ontology of services which defines the semantic of the elements of the environment through the services they offer. Our model of representation allows us to deduce in a unified manner the quality of service (*QoS*) provided by an object type of environment. This will be used for decision support of the agents. Our goal is to allow agents to make the best choice among all object types in the environment that enable them to achieve their action. To do this, we proposed a generic level in the ontology to represent the definition of concepts that allow us to make our calculation of *QoS*. The semantic of the environment is used at key moments in the evolution of agents (during the planning and execution of the plan) to accompany the agent in these behaviors by allowing it to react to what happens in its environment in a consistent manner and opportunistic.

We have implemented our model in the simulation platform project where we got the first results. The numerical evaluation of the results is not the purpose of this article but we plan to evaluate our model on its performance especially for a large number of heterogeneous elements of the environment and a large number of agents.

In the near future, we are considering the addition of a fourth parameter to assess the *QoS* that defines the availability of the element of environment. As it is now, the agents observe the availability of an object in the environment when they can see it. But this can be avoided by taking into account knowledge of the environment that will be used to filter out inconsistent results (eg, the banking services offered by the bank are unavailable after 6pm).

## Acknowledgment

This research received support from the TerraDynamica Project (FUI8) funded by the City of Paris, the Local Councils of Val d'Oise, Seine-Saint-Denis, Yvelines, the Regional Councils of Ile de France and Aquitaine and the French Ministry of Economy, Finances and Industry, Directorate for Competitiveness of Industry and Services.

## References

- [1] R. Aylett and M. Cavazza, "Intelligent virtual environments: a state-of-the-art report," in *Eurographics 2001, STAR Reports volume*, 2001, pp. 87–109.
- [2] P. Chevaillier, T.-h. Trinh, M. Barange, J. Soler, P. De Loor, and R. Querrec, "Semantic modelling of virtual environments using mascot," in *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems SEARIS, IEEE VR 2011*, 2011.
- [3] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. D. Kraker, "The role of semantics in games and simulations," *Comput. Entertain.*, vol. 6, pp. 1–35, 2008.
- [4] K. Otto, "Towards semantic virtual environments," in *Workshop Towards Semantic Virtual Environments*, 2005.
- [5] K. M. Stanney, *Handbook of virtual environments: design, implementation, and applications*, K. M. Stanney, Ed. Lawrence Erlbaum Associates, 2002.
- [6] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in *Formal ontology in conceptual analysis and knowledge representation*. Kluwer Academic Publishers, 1993.
- [7] B. Pellens, O. D. Troyer, W. Bille, and F. Kleineremann, "Conceptual modeling of object behavior in a virtual environment," V. C. 2005, Ed. Springer-Verlag, 2005.
- [8] O. D. Troyer, W. Bille, R. Romero, and P. Stuer, "On generating virtual worlds from domain ontologies," in *Proceedings of the 9th International Conference on Multi-Media Modeling*, 2003.
- [9] E. Kalogerakis, S. Christodoulakis, and N. Moutoutzis, "Coupling ontologies with graphics content for knowledge driven visualization," in *Proceedings of the IEEE conference on Virtual Reality*. IEEE Computer Society, 2006, pp. 43–50.
- [10] R. C. Vanacken Lode, De Boeck Joan and C. Karin, "Using relations between concepts during interaction modelling for virtual environments," vol. Software Engineering and architectures for Realtime Interactive Systems, 2008.
- [11] L. Edward, K. Amokrane, D. Lourdeaux, and J.-P. A. Barthès, "Knowledge representation - an ontology for managing a virtual environment," in *ICAART*, 2010.
- [12] F. Grimaldo, M. Lozano, F. Barber, and G. Viguera, "Simulating socially intelligent agents in semantic virtual environments," *Knowl. Eng. Rev.*, vol. 23, no. 4, pp. 369–388, 2008.
- [13] Y.-L. Chu and T.-Y. Li, "Using pluggable procedures and ontology to realize semantic virtual environments 2.0," 2008.
- [14] M. Kallmann and D. Thalmann, "Direct 3d interaction with smart objects," in *Proceedings of ACM VRST'99*, 1999.
- [15] S. D. Marwan Badawi, "Autonomous agents interacting with their virtual environment through synoptic objects," in *The 17th Conference on Computer Animation and Social Agents*, 2004.
- [16] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, pp. 34–43, 2001.
- [17] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "Owl-s: Semantic markup for web services," in *W3C*, ser. <http://www.w3.org/Submission/OWL-S/>, 2004.