Analele Ştiinţifice ale Universităţii "Alexandru Ioan Cuza" din Iaşi
Informatică, Tomul XV, 2004, 70–80

Scientific Annals of the "Alexandru Ioan Cuza" University of Iaşi
Computer Science Section, Tome XV, 2004, 70–80

The Use of Semantic Web Technologies in the Context of Virtual Environments

Sabin Buraga   Ştefan Tanasă
Faculty of Computer Science
"A. I. Cuza" University of Iaşi, Romania
{busaco,stanasa}@infoiasi.ro

Abstract

The paper proposes a metadata model and a simple ontological model in order to specify certain semantic information about the tridimensional virtual worlds. As metadata language, *XFiles* language is used. The simple ontology defined in this paper is written in OWL.

1 Introduction

In this paper we propose an ontological model to be used in the context of tridimensional virtual worlds. This approach could be helpful for detecting relations established between elements from same environment or between different virtual worlds available on Internet. Also, the established connections between the objects of the 3D system are very important in the virtual world analysis phase in order to improve the realism of the environment.

For the 3D virtual worlds system we'll associate certain metadata statements written in RDF (Resource Description Framework) [5] and each main concept of the environment (such as a geometric 2D object or a non-visual object) will be expressed by OWL (Web Ontology Language) [4] constructs.

Our proposal can be viewed as a first step towards a semantic model for tridimensional virtual worlds environment.

2 Tridimensional Virtual Worlds

In this section, we'll briefly present the core concepts of the tridimensional virtual worlds, such as virtual reality, scene graph or object. Also, we'll give in section 2.3 a classification of the actual 3D virtual worlds.

2.1 Preamble

Virtual reality is a user-interface paradigm in which the user feels immersed in a computer-generated space.

A *virtual world* is a virtual reality system that allow multiple users to interact in the same space [1]. In this paper we refer to virtual worlds that consists of a common tridimensional space with a graphically representation. The ordinary Internet/Web chats are virtual worlds, but they have not a common tridimensional space.

Some of the important issues concerning the development of a 3D virtual worlds system are [7]:

- a distributed architecture needs to be supported;
- objects need to persist over time;
- end users should be able to easily extend the system;
- end users should be able to make changes to the environment while the system is already running;
- the system must be secure.

2.2 Objects

Each 3D virtual world consists of an hierarchy of objects (see figure 1).

There are child–parent relations and for that reason there are no cycles. This hierarchy forms an acyclic graph called *scene graph*. Unlike DOM (Document Object Model) [9] representation of XML documents where a child node has just one parent, the scene graph can have children with multiple parents. Moreover, the scene graph has not the restriction of one root node (the acyclic graph is not always a tree). Each node (object) from the scene graph may have children nodes and fields with values. The type of the field is given by the set of the all possible values.

Some objects of the environment may have a graphical representation. This representation may be geometric, in a 2 or 3-dimensional space. The 3D objects are *simple* (e.g., box, sphere, cone etc.) or *complex* (e.g. extrusion, NURBS). Of course, for a certain user that use the environment, a graphical object may be visible or not.

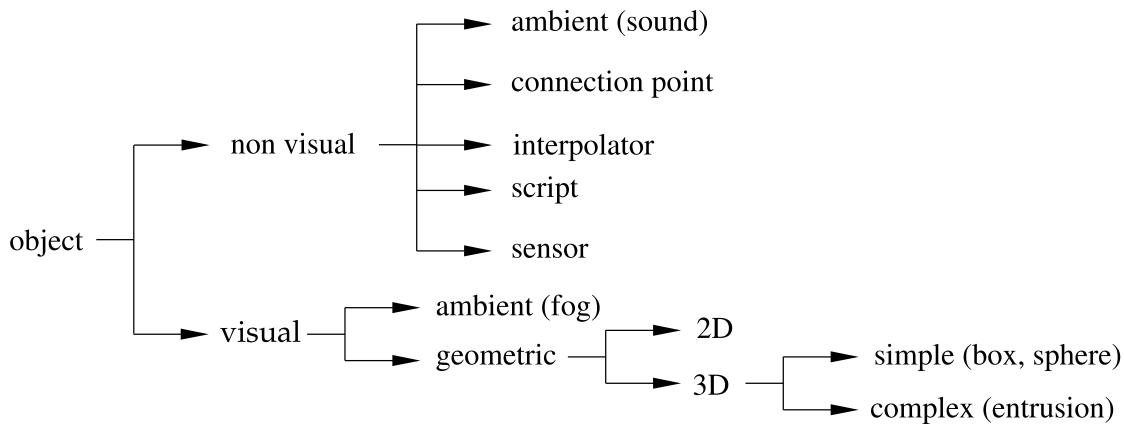

Figure 1: Object hierarchy

An object can be *dynamic* or *static*. An object is dynamic if it has at least one subordinate dynamic object or at least one field has a variable value. If an object is not dynamic then it is static.

An object is *interactive* if there is an user that can action directly to it or to a subordinate node. Otherwise the object is *non-interactive*.

Not all interactive nodes are dynamic. For example, there may be a hyperlink node that allows interactions from users but it doesn't change its fields or subordinate nodes.

2.3 Virtual World Classification

A virtual world is a static one if the scene graph contains static nodes and dynamic otherwise. Also, a virtual world is interactive if there are interactive objects.

If more then a user can take place to the virtual environment and to percept the presence of the others, then the virtual world is a *multiuser* one. Users can interact one with each other or with the objects from environment.

The virtual world is *distributed* if parts of it are located at different places in the local network or Internet. In some environments users can bring own objects, such as an umbrella, a notebook, or a laptop.

A special case is when the users' computers are involved in the execution of the virtual world application. In this situation the virtual world is distributed from the execution point of view.

The cooperative work in a 3D virtual worlds system implies joint editing of shared objects, while collaborative additionally allows truly concurrent editing[1].

Frequently, a virtual world is cooperative if some objects supports interactions from many users but not simultaneously. The collaborative environments contains objects that allow concurent interactions.

3 Associating Metadata to Virtual World Objects

One first step towards an ontological model for tridimensional virtual worlds enviroment is to associate metadata to certain objects of this environment. To accomplish this goal, we'll use RDF (Resource Description Framework) [5] statements.

3.1 Resource Description Framework

3.1.1 Caracterization

RDF is a standardized basis for processing metadata. The used metadata format should allow us to reason about data. The RDF is intended to be used to capture and express the conceptual structure of information offered in the Web. Particularly, we'll use RDF to express the conceptual structure of a tridimensional virtual environment.

The RDF assertions can be considered as a data model for describing machine processable semantics of data to build the infrastructure for Berners-Lee's *Semantic Web* [3, 9].

3.1.2 RDF Model

RDF consists of a model for the representation of named properties and property values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute–value pairs. RDF properties also represent relationships between resources and therefore a RDF model can resemble an entity–relationship diagram.

To facilitate the definition of metadata, RDF is based on *classes.* A collection of classes, typically designed for a specific purpose or domain, is called a *schema.* Through the sharability of schemas, RDF supports the reusability of metadata definitions. The RDF schemas may themselves be written in RDF.

The basic model of RDF consists of three object types:

resources All objects being described by RDF expressions are called *resources* and they are always named by Uniform Resource Identifiers (URI) [9] plus optional anchor identifiers. Using URI schemas (i.e. `http`, `ftp`, or `file` schemas), every type of resource can be uniformly identified.

properties A *property* is a specific aspect, characteristic, attribute, or relation to describe a resource, as stated in [5]. Each property has a specific meaning, defines its permitted values, the type of resources it can specify, and its relationship with other properties (via RDF Schema).

statements A specific resource together with a named property, plus the value of that property for that resource is an RDF *statement*. These three individual parts of a statement are called, respectively, the *subject*, the *predicate*, and the *object*. The object of a statement (e.g., the property value) can be another resource or a literal.

RDF also specifies three types of container objects:

- *Bag* (an unordered list of resources or literals),
- *Sequence* (an ordered list of resources or literals),
- *Alternative* (a list of resources or literals that represent alternatives for the single value of a property).

The collections can be used instead of *Description* elements. The containers may be defined by a URI pattern. RDF can also be used to make statements about other RDF statements (higher-order statements).

The RDF data model provides an abstract, conceptual framework for defining and using metadata. The concrete RDF syntax is based on XML.

3.2 3D Object Metadata

Using *XFiles* [2] language, we'll associate to each object of the 3D virtual environment certain metadata, such as owner, type, spatial relation with other objects of the environment, alternatives of representation (to establish different levels of detail in the case of visible onjects) and others.

3.2.1 *XFiles* Language

Initially, the elements of the *XFiles* language was designed to be used to specify different metadata (using RDF statements) about the components of a distributed file system or about the relationship between these components.

A short description of these elements is following (for details, see [2]).

The root element of an *XFiles* document is the *Properties* element. This element may contain, in any order, the following child elements:

- the *Type* element reflects the file type: ordinary, directory, pipe, symbolic or hard link, character or block device, or socket, on Unix-like systems; the *mime* attribute specifies the MIME (Multipurpose Internet Mail Extensions) type for a file (i.e. `text/html`, `image/gif` or `model/vrml`);

- the *Location* element represents the IP address of the host on which a certain file resides; the *dns* attribute is used to specify the Domain Name System (DNS) entry for the given IP address;

- the *Auth* element specifies the authentification method to access a given file;

- the *Owner* element includes the information about the owner of a file: login name, password, group, real identity etc.; it is possible to have multiple owners for a single given file;

- the *Size* element specifies the actual file size; *max* attribute denotes the maximum permitted size for a file;

- the *Permissions* element reflects the set of file permissions;

- the *Timestamp* element gives the possibility to track the access, modification or status-change time of a specific file;

- the *Version* element can be used in a Concurrent Versions System (CVS) environment, for versioning purposes;

- the *Parse* element denotes the application(s) used to process a file (e.g. file editors, compilers, viewers etc.); the *params* attribute can be used to pass additional options to a program.

3.2.2 *XFiles* – as Metadata language

We propose to use *XFiles* constructs in order to specify certain metadata about different objects from a 3D virtual worlds system.

- the *Type* element can reflect not only the file type (e.g. VRML world), but the type of the object (e.g. 2D or 3D, visible or non-visible etc.).
- the *Location* element can contain details about the physical localization (for example, the IP address of the user's host) or the logical localization (inside/outside a certain world) of an object.
- the *Auth* and *Permissions* elements can help to restrict the access to different objects/worlds or to prevent malicious code to be executed within the environment. This will assure the verification of certain conditions imposed at the security level of the system.
- the *Owner* element will be used to know the actual owner of an object (for example, to track the ownership changes for certain objects, such as an umbrella);
- the *Timestamp* element can give interesting details about the evolution in time of an object.

Example

To associate the global metadata to a collection of 3D objects (a table and a glass), we can generate the following RDF document:

```
<rdf:RDF>
  <rdf:Bag rdf:ID="objects">
    <rdf:li rdf:resource="#table" />
    <rdf:li rdf:resource="#glass" />
  </rdf:Bag>

  <rdf:Description rdf:about="#objects">
    <xf:Properties>
      <!-- no authentication requirements -->
      <xf:Auth>none</xf:Auth>
      <!-- ownership information -->
      <xf:Owner>
```

```
        <rdf:Description rdf:about="#environment">
          <!-- other constructs... -->
        </rdf:Description>
      </xf:Owner>
      <!-- public access -->
      <xf:Permissions>
        <xf:Permission>public</xf:Permission>
      </xf:Permissions>
    </xf:Properties>
  </rdf:Description>
</rdf:RDF>
```

4 Simple 3D Object Ontology

An *ontology* is the set of terms that describe the concepts and relationships that represent an area of knowledge.

Generally, an ontology must define the following:

- the classes or element types that compose the area of knowledge;
- the relationship among those;
- the attributes (properties) of the classes and relationships.

There are several kinds of ontologies, for instance [6]:

- *taxonomic* – which consists in a simple classification of the elements in a hierarchical scheme;
- *metadata* – using metadata, additional information can be associated to each element of the knowledge area, which allows for more complex inter-relations.

4.1 Our Proposal

Following the classification presented in section 2.3, we propose a simple ontology that can be used to model the 3D virtual worlds. Using this ontology, we can model the knowledge about the entities from virtual reality domain.

The general (uppest) class of the proposed ontology is *Object*. Derived classes are *Visual* and Non-visual, using *isVisible* relation.

The *Non-visual* class has three sub-classes: *Sensor*, *Script* and *Interpolator* (these classes correspond to the usual types of non-visual objects of a 3D environment).

The *Visual* class is composed by *Ambient* and *Geometric* sub-classes. The *Geometric* class will contain all visible 2D and 3D objects, using two other sub-classes: *2DObject* and *3DObject*.

The properties of each proposed classes will be the general properties of the corresponding nodes that model each object – at the description level – in VRML (Virtual Reality Modeling Language) or X3D languages [8].

4.2 Examples

In this example, we use OWL (Web Ontology Language) [4] constructs to denote *Non-visual* and *Visual* classes:

```
<owl:Class rdf:ID="Non-visual">
   <rdfs:subClassOf rdf:resource="#Object" />
   <owl:disjointWith rdf:resource="#Visual" />
</owl:Class>

<owl:Class rdf:ID="Visual">
   <rdfs:subClassOf rdf:resource="#Object" />
   <owl:disjointWith rdf:resource="#Non-visual" />
</owl:Class>
```

Also, we specify the *storedOn* object-property to denote the fact: an object is stored on a Web server (an instance of the *Site* class):

```
<owl:ObjectProperty rdf:ID="storedOn">
   <rdfs:domain rdf:resource="#Object" />
   <rdfs:range rdf:resource="#Site" />
</owl:Class>
```

Each visual object will be stored by a least one Web site:

```
<owl:Class rdf:ID="Visual">
   <rdf:subClassOf rdf:resource="#Object" />
   <rdf:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="#storedOn">
```

```
        <owl:minCardinality
           rdf:datatype="&xsd;nonNegativeInteger">
           1
        </owl:minCardinality>
     </owl:Restriction>
  </rdf:subClassOf>
</owl:Class>
```

5 Conclusion

In this paper, we proposed a metadata model and a simple ontological model in order to specify certain semantic information about the tridimensional virtual worlds.

In section 2, after a short description of virtual worlds, we gave a classification of the objects involved in such a system and a taxonomy of virtual worlds.

We continued with the specification of the metadata level. For this, our proposal used *XFiles* language, presented in section 3.2.1. In this context, *XFiles* constructs can be considered as containters for metadata information associated to certain objects of the 3D system. To effectively associate the metadata, we used RDF statements.

Next section presented a simple ontology in order to model the 3D virtual worlds. We defined different classes, following the 3D object classification. Section 4.2 gave some examples of classes and properties modeled in OWL. Of course, the proposed ontology will be enriched with more complex constructs. Another approach is to use different other ontological models that can be helpful to specify, for example, user interactions.

Our proposal can be viewed as a first step towards a semantic model of tridimensional virtual worlds, in order to place virtual reality applications in the context of semantic Web. This approach could open interesting directions of research in the *semantic virtual reality* domain.

References

[1] W. Broll, “Interacting in Distributed Collaborative Virtual Environments”, in *The Proceedings of the IEEE VRAIS’95*, IEEE Computer Society Press, 1995, pp. 148-155.

[2] S. C. Buraga, "A Model for Accessing Resources of the Distributed File Systems", în D. Grigoraş *et. al* (eds.), *Advanced Environments, Tools, and Applications for Cluster Computing. NATO Advanced Research Workshop, IWCC 2001 – revised papers*, Lecture Notes in Computer Science – LNCS 2326, Springer-Verlag, Berlin, 2002, pag. 224–230.

[3] J. Davies, D. Fensel, F. van Harmelen (eds.), *Towards the Semantic Web*, John Wiley & Sons, England, 2003.

[4] M. Dean, G .Schreiber (eds.), *OWL Web Ontology Language Reference*, W3C Recommendation, Boston, 2004:
`http://www.w3.org/TR/owl-ref/`.

[5] F. Manola, E. Miller (eds.), *RDF Primer*, W3C Recommendation, Boston, 2004:
`http://www.w3.org/TR/rdf-primer/`.

[6] M. Miranda, "Ontology for Virtual Worlds", *Technical Report*, Universidad Politecnica de Madrid, 2003.

[7] S. Singhal, M. Zyda, *Networked Virtual Environments: Design and Implementation*, SIGGRAPH Series, ACM Press, 2000.

[8] * * *, *Web3D Consortium*,
`http://www.web3d.org/`

[9] * * *, *World Wide Web Consortium*,
`http://www.w3.org/`