# Ontology of a scene based on Java 3D architecture.

Holman Bolívar Barón[1], Rubén González Crespo[2] Oscar Sanjuán Martínez[3]

*[1]Facultad de Ingeniería, Universidad Católica, Bogotá, Colombia*
*[2]Departamento lenguajes y sistemas informáticos, UPSAM, Madrid, Spain.*
*3Departamento de Informática, UNIOVI, Oviedo, Spain*

***Abstract* – The present article seeks to make an approach to the class hierarchy of a scene built with the architecture Java 3D, to develop an ontology of a scene as from the semantic essential components for the semantic structuring of the Web3D**

**Java was selected because the language recommended by the W3C Consortium for the Development of the Web3D oriented applications as from X3D standard is Xj3D which composition of their Schemas is based the architecture of Java3D**

**In first instance identifies the domain and scope of the ontology, defining classes and subclasses that comprise from Java3D architecture and the essential elements of a scene, as its point of origin, the field of rotation, translation The limitation of the scene and the definition of shaders, then define the slots that are declared in RDF as a framework for describing the properties of the classes established from identifying the domain and range of each class, then develops composition of the OWL ontology on SWOOP**

**Finally, be perform instantiations of the ontology building for a Iconosphere object as from class expressions defined.**

***Keywords: Ontology, Java3D, Web3D, X3D.***

## I. Introduction

THE term web 3D makes reference to the programming language, standards, files formats and any kind of technology that can be used to the construction of virtual environments of immersion in the web [1]. The construction of those environments is not easy and requires an artist team, illustrators, designers and programmers for its construction, turning this process into an independent labor of the required technology to make it real [3]. To turn this artisanal labor into an industrial one , it is necessary, first of all, to identify the semantic structure of the 3D objects that constitute the immersion environments, to make the searching, recycling, development and the assemble of new environments, easier, faster and richer.

Taking in account that the majority of 3D environments is characterized for its objects, which are built for geometrical low level elements, such as polygonal net or NURBS surfaces [6] ; the authors make the semantic association, sharing the perception of the visitors toward the objects which compound the environment, unfortunately even when the users Identify the relationships and the semantic associations of the 3D objects, they cannot infer high level information descriptions, stored in the 3D, and they cannot make the manufacture process of new environments. Besides nowadays the molding practices do not stimulate the semantic connotation of the 3D objects, some of them specify limited opportunities, such as the manager of layers or the tagging of objects; which is not constantly used for the standard storage of the high level structures and it is presented to other people as an additional working molding result [15]. The lack of a high level standard for the environment elements description is an inconvenient for the advanced usage of 3D world description, which require the semantic knowledge of those elements. The knowledge usage possibilities, are first of all the searching engines, which process formulated requests in a natural language making reference to the high level characteristics (for instance, finding buildings that have inside of them, patios surrounded by columns) in second place the extraction of semantic objects from big files to facilitate the exam and the automatic creation of high level libraries which can be used to create different 3d environments, knowing that the usage of high level description is not only useful in the 3d virtual environments but includes everything which is related with the mixed reality. (For example the high level description of a virtual model in a real map environment, can help to the navigation in a real or mixed scene when summarizing the virtual and real elements of it) [15].

The current research proposes to develop the semantic composition, starting in the hierarchy of the tridimensional visualization based on Java 3D architecture, used in the construction of 3D objects, having in account the architecture of the geometrical composition that is proposed in the X3D standard, for this reason a revision of the syntactic point of view is made, also a semantic one based on the principal programming languages for virtual immersion environments in the web, such as VRML and O3D, to unify a hierarchy of the 3D objects construction. For the development of the semantic composition, we apply the methodology competency questions; this one allows the construction of ontologies about specific domains, putting limits to the domain, defining the classes and sub classes, the relationships between them and the production parting from the SWOOP tools, generating RDF and OWL files [13].

## II. 3D E INTERNET

There are a variety of 3D applications development oriented internet especially in the entertainment, which require many developers and special processing specifications that are not common to most people, these many people can be made by communities interested in create 3D environments for use either as a business, an educational or entertainment [1], as with SecondLife, developed for the Linden Lab laboratory [17], that provides the infrastructure for processing and storage that this community needs. Development in Second Life begins with structures pre-established because the interactivity and usability are subject to the proposed platform by Linden Lab, with low-level geometric structures without any semantic description [15], impossible to identify the properties of the virtual environment and the possibility of building manufacturing agents for distributed development [7]. The construction of an ontology of a scene is needed as a starting point for developing integrated services Web3D distributed manufacturing agents for complex and rich virtual worlds [18].

## III. WEB 3D STANDARS

There are several standards used in the Web 3D, the main ones: VRML (Virtual Reality Modelling Language) and X3D (Extensible 3D) [6]. VRML is a file format that allows the creation of interactive objects and worlds in three-dimensional [20]. X3D is an open standard XML, a 3D file format that allows the creation and transmission of 3D data between different applications, especially in network [22]. The ISO 19775 Extensible 3D specification provides a application programming language independent interface (API), allowing access to a range of services and functions, For integration with a programming language. X3D provides abstract interfaces that depending on the syntax are inserted into an established language. One of the languages that allow for such insertion is Java, which generates the specification Xj3D. ISO / IEC 19777 specifies a layer for the Java programming language [23]. One of the main differences between VRML/X3D and Xj3D, conceptually, is that the latter is defined as a script programming language of low level 3D scenes [10]. The creation of three dimensional objects and Java elements requires not only the formation of 3D elements, but also the definition of all aspects of display and control capabilities [16]. To create the simplest scenario, the Java code is well above that required in VRML/X3D, but on the other hand the control of different elements in the system is superior and more natural in Java [9]. You can also use Java as file viewer VRML/X3D, using one of the loaders of VRML/X3D developed for Java. The main advantages are the ability to execute on different platforms and avoid the need to install a specific plugin for each browser [11].

## IV. EXTENSIBLE 3D LANGUAGE BINDINGS JAVA

This specification provides a set of independent implementations of the classes and interfaces that represent the possible interactions with the X3D scene through UPS. The file shows the specifications of the application must remain hidden to the user. If classes are declared abstract, it is expected that the classes that inherit from the browser they use according to the needs [23]. A communications session based on Java depends on the type of driver set: applet or component. The difference is given by the way it obtains relating to browser, once obtained may not differ from services using a request time in which it generates the Java code you want to access the X3D browser [16]. An X3D browser is set by creating a new instance of the component class X3D CreateBrowser that can be used to control the properties of the browser. For example, a property can be used to request a Swing based component instead of AWT, or a software off-screen rendering by establishing and running the browser is also able to add additional properties. [23]. Xj3D uses Java3D framework and adapts according to the specification of developing an X3D XML Schema [21], that can create your own data types and namespaces, Because X3D Shaders specifies an architecture without to point in the operation and integration of these. Because framework Xj3D be used java3D architecture, this is selected for the specification of the scene [10].

## V. ARCHITECTURE OF A JAVA3D SCENE

Every Java3D scenario begins with a virtual universe that will be the container for all the graphics on the scene; this consists of locating objects that establish the relevant positions on the scene. Each universe will be a totally independententity; therefore an object won’t be able to exist in two virtual universes [16]. The locating objects are instances of the class Locale and are the node of the graphic branches in which the scene will be divided into. They consist of BranchGroup and are in charge of defining the origin point in the Cartesian space from doublé precision coordinates. There will be one for each universe. From the locating object which is the trunk of a tree there are groups of branches, transformations and visualization spreading out.

The compiler at the time of making a render follows the classes’ hierarchy, which, if not respected, will generate class recognition errors. The class Locale has a subclass called Node and most of the objects appearing on the scene are at the same time Node subclasses; therefore it is established that Node puts together the branches, the transformations and the visualization. From these groups we get graphic objects that will allow us to define the shape of the objects, its colour and texture composition and the lighting needed to be visualized; these objects will be established as Shape. [5]. Figure 1 presents the diagram of the hierarchy of the classes of a scene made with Java3D. In the Java3D programs you have to program in stages each piece of the scene’s diagram and then connect each stage with each other to make the final program. The virtual universe is the universo that hides behind the screen. When programming in three dimensions what we have to achieve is to project a shaft from the object behind the screen to the observer's eye. [8].

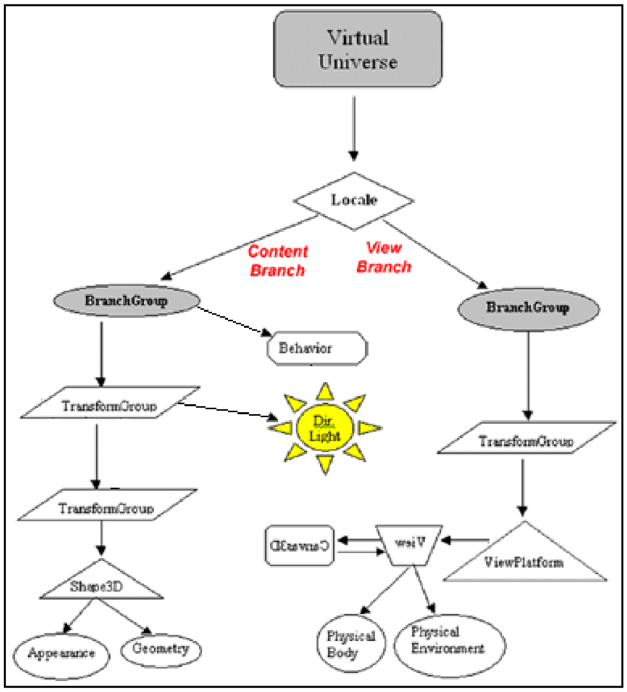


Figure 1. Diagram of a Graphic Scene

## VI. Domain and Reach

With the development of this ontology is to make a first approach to the conceptualization of a 3D scene from Java3D architecture, identifying the resources essential to the definition of the scene and the properties or slots of these resources [14]. With the semantic conceptualization of a scene is looking to establish class structures of the objects composing the scene, to facilitate assembly and reuse of these objects in the construction of more complex scenes with the help of agents manufacturing [18]. The ontology should answer questions such as:

- What is a scene?
- What are the essential components of a scene?
- What are the relationships between these components?
- What is the limiting factor in the scene?
- Where is the origin of the scene?
- How it is given the shifting of the elements in the scene?
- How it is given the rotation of the elements in the scene?
- How the scene is displayed?
- What is the range of illumination of the scene?

## VII. Classes and Subclasses

Bearing in mind the construction of a simple scenario in Java3D, nine classes have been identified that allow us to create a virtual universe, locate the origin, put together transformations, branches and object visualization components that will be made from meshes of triangles with shadows and lighting properties.

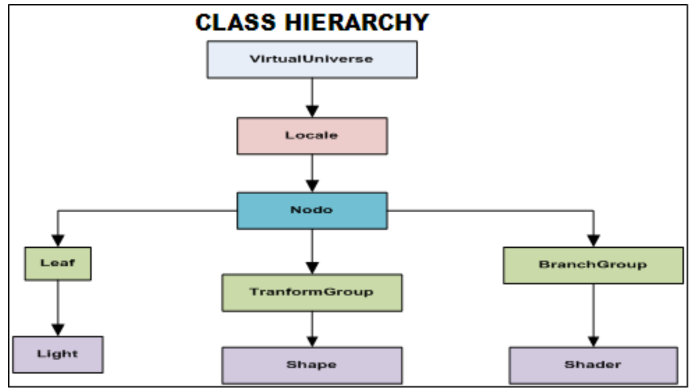


Figure 2. Diagram of a Graphic Scene

Figure 2 shows the hierarchy of classes used and the directions of the arrows allow us to infer the inheritance relations between them; the necessary classes to build a simple scene in Java3D are:

a. VirtualUniverse: It is all the other ones' superclass as it is the container of all the graphics in the scene.
b. Locale: It defines the Cartesian origin of the
c. Euclidian space.
d. Node: It allows the definition of the representations of the scene
e. Leaf: It allows the specification of shapes, lighting and behavior.
f. TransformGroup: It places the objects in space and allows its rotation and scalability.
g. BranchGroup: It is in charge of the visualization of the scene..
h. Light: It identifies a specific type of lighting.
i. Shape: Construye formas 3D.
j. Shape: It builds 3D shapes.
k. Shader: It defines color and shadows of the objects created with Shape

For the edition of the ontology in OWL language (Ontology Web Language), which was selected because it was W3C's recommendation and the fact it allows several functionalities such as DAME and OIL, as it is and RDFS model's extension, they allow the redefinition of properties and the definition of classes through Boolean restrictions and operations, so the editor SWOOP beta 2.3 edition was used [2].

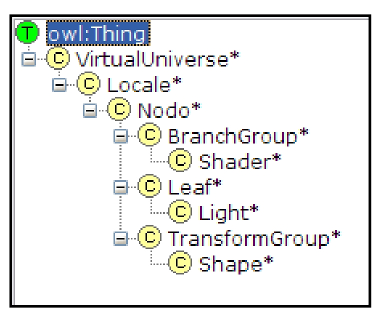


Figure 3. Classes and Subclasses

Figure 3 shows clases and subclases of tree in SWOOP editor. The description languages' function is to represent data and there is a frame for the description of RDF resources (Resource Description Framework) which is one of the XML vocabularies and it is made up based on the following rules [12]:

- A resource is any thing that may have an URI, this includes all the web sites, all the individual elements of each XML document y many more.
- A property is a resource that has a name and that can be used to identify responsibilities or other resources' actions.
- A sentence consists of a combination of a resource, a property and a value. These parts are known as subject, predicate and object of the sentence. The use of RDF allows the independence between resources, the exchange of information, and the scalability due to its simplicity. [4].

Figure 4 shows the RDF description of the Class VirtualUniverse:

```
<rdf:RDF xml:base="&EGJ3D;#"
         xmlns:owl="&owl;"
         xmlns:rdf="&rdf;"
         xmlns:rdfs="&rdfs;">
  <owl:Class rdf:about="#VirtualUniverse">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Locale"/>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#defineUniverso"/
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#Locale"/>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>

  <owl:AnnotationProperty rdf:about="&rdfs;comment"/>
</rdf:RDF>
```

Figure 4. RDF of the Class VirtualUniverse

## VIII. Properties Definition

Properties allow the establishment of the classes' responsibilities and the relations between them, which have a range and an application domain [11]. The ontology properties where identified from the classes responsibilities and then it was establish their range and domain so the relation between the different classes could be identified. The properties established where:

i. **defineUniverse**: Property responsible for the class VirtualUniverse and corresponds to the graphic scenario's creation, but this needs other elements such as nodes, the Locale, objects' geometry, the light and the visual appearance. The domain and range of this property is the class VirtualUniverse.

ii. **especifyCartesianOrigin**: Property responsible for the class and consists of the identification of the origin of the similar universe. The domain and range of this property is the class Locale

iii. **defineUtilityScene**: An object in a scene needs light to be visualized, a shape and geometric composition, an appearance, and certain rotation around an axis possibilities or movement from one point to another. There are several classes in charge of these actions, like the class Shape, which defines the geometric shape or the class Shader in charge of the appearance, but all of these belong to the class Node which is the domain and range of this property

iv. **DefineLightingRange**: In order to visualize an object in a scene first you have to establish the light from the scene that is getting to each of the points of the projection plane. The best way to answer this question is to follow a straight line from this point in the projection plane and the focal point until it impacts a representable surface in the scene; at this point the light that should impact this point is established. The properties of the surface and the light's incidence angle establish what amount of light should be reflected by the incident vision angle; this is the reason why it is necessary to define the range of lighting with the class Light. The domain and range of this property is the class Leaf.

v. **visualize**: The light's impact over a surface and the subsequent re-radiation by diffusion phenomenon can be very fuzzy , for example, the re-radiation in every direction in an isotropic way. This means the camera will see the same amount of light from that surface point regardless the incident vision angle [8]. Given that the amount of light that impacts a surface depends on the angle of incidence of the light. If most of the light hitting a surface is reflected in a fuzzy way, the surface will have a matt appearance. This is why during the visualization process it is necessary to establish the way in which the object will radiate its environment affecting the rest of the objects in the scene. The domain and range of this property is the class BranchGroup.

vi. **moves**: This property allows the recalculation of all the positions of all the vertexes of a geometric object from a relocation of the centroid in the Euclidean space. The domain and range of this property is the class TransformGroup.

vii. **rotates**: This property performs turns of the object around the axis of the centroid specifying an axis in a single direction, x, y or z, or a diagonal established. The domain and range of this property is the class TranformGroup.

```
<rdf:RDF xml:base="&EGJ3D;#"
         xmlns:owl="&owl;"
         xmlns:rdf="&rdf;"
         xmlns:rdfs="&rdfs;">
  <owl:ObjectProperty rdf:about="#defineRangoIluminación">
    <rdfs:domain>
      <owl:Class rdf:about="#Leaf"/>
    </rdfs:domain>
    <rdfs:range rdf:resource="#Leaf"/>
  </owl:ObjectProperty>
</rdf:RDF>
```

Figure 5. RDF of the Propertie defineLightingRange

 *ISSN - 1989-1660*

Figure 5 shows the RDF description of the property defineLightingRange.

## IX. INSTANCES CREATION

The instantiation is the creation of individuals from the assignment of values to the attributes of a class therefore each individual will have certain particular properties defined in the class where he belongs. For each class an individual was defined which has certain particular characteristics opposite to the domain range of the properties responsible for the classes:

- **Screens:** Instance of the class VirtualUniverse that is a simple graphic container.
- **Axis:** Instance of the class Locale. Screen locator placing the Y plane horizontal, the X plane perpendicular to the screen and the Z plane vertical.
- **Layers:** Instance of the class Node. Allows the visualization of the scene in perspective.
- **Leaf:** Instance of the class Lamp, which allows the establishment of the focal axis of the light.
- **Pipeline:** Instance of the class BranchGroup, assigns a determined color to the faces of the geometric object**.**
- **Scale:** Instance of the class TranformGroup that modifies the size of the geometric figure.
- **Icoesphere:** Instancia de la clase Shape y crea un Icosaedro que tiene 20 caras todas formadas por triángulos equiláteros.
- **Fresnel:** Instance of the class Shader; defines how reflective the material is depending on the angle between the normal surface and the observation direction.
- **Hemi:** Instance of the class Light; projects light evenly through a hemisphere that surrounds the scene.

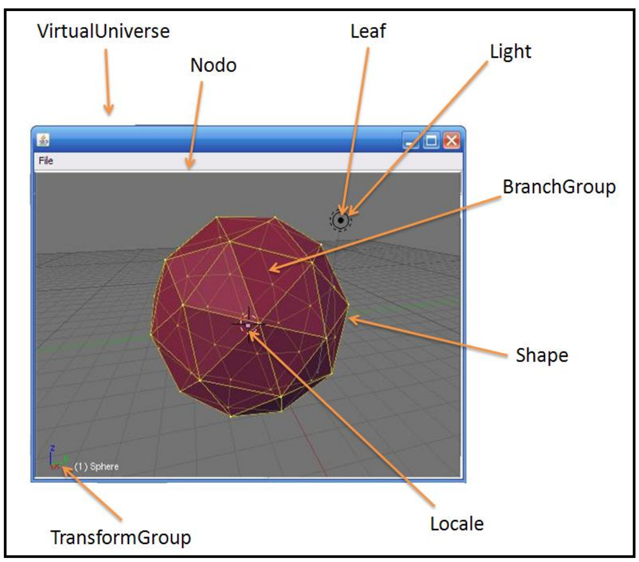


Figure 6. Instantiation Example

## X. CLASS EXPRESSIONS

The class expressions allow us to establish the relations between the classes and the properties so by means of inference the relations between individuals can be established, the following class expressions were created:

$$VirtualUniverse \equiv (\exists Locale.defineUniverso) \cap Locale$$

$$Locale \equiv (\exists Nodo.especificaOrigenCartesiano) \cap Nodo$$

$$\equiv (\exists BranchGroup.defineUtilidadEscena) \cap BranchGroup \cap TransormGroup \cap Leaf$$

$$Leaf \equiv (\exists Light.ilumina) \cap Light$$

$$BranchGroup \equiv (\exists Shaders.visualiza) \cap Shaders$$

$$TransforGroup \equiv (\exists Shape.trasladar) \cap Shape$$

```
<rdf:RDF xml:base="&EGJ3D;"
         xmlns:EGJ3D="&EGJ3D;"
         xmlns:owl="&owl;"
         xmlns:rdf="&rdf;"
         xmlns:rdfs="&rdfs;">
  <EGJ3D:Shape rdf:about="#Icosphere">
    <rdfs:comment xml:lang="en"></rdfs:comment>
  </EGJ3D:Shape>
  <owl:Class rdf:about="#Shape"/>
  <owl:AnnotationProperty rdf:about="&rdfs;comment"/>
</rdf:RDF>
```

Figure 7. Instantiation Example

## XI. CONCLUSION

The construction of an ontology for the definition of tridimensional spaces will allow the Web3d to standardize the development of scenarios and the creation of manufacture agents that will make easier the modeling and texturing processes. The biggest difficulty in the ontology's development lies on the definition of abstract classes which allow the instantiation of non visual individuals, which determine graphic behaviors. The identification of the properties from the responsibilities of the classes generates different fields of similar domains bringing all the classes closer to a general equivalence state. All classes are equivalent with each other, for example, the class Light is equivalent to the class VirtualUniverse.

## References


[1] ALPCAN. Tansu, BAUCKHAGE. Christian, KOTSOVINOS. Evangelos (2007). Towards 3D Internet: Why, What, and How?. Deutsche Telekom Laboratories Ernst-Reuter-Platz 7, 10587 Berlin, Germany. http://www.tansu.alpcan.org/papers/Alpcan-3D_Internet.pdf

[2] ANTONIOU. Grigoris, HARMELEN. Frank Van. (2004). Web Ontology Languaje: OWL. Department of Computer Science, University of Crete and Department of AI, Vrije UNiversiteit Amsterdam.

[3] BERTINI, Marco. Del BIMBO, Alberto. (2005). Automatic video annotation using ontologies extended with visual information. Proceedings of the 13th annual ACM international conference on Multimedia. (pp. 395 - 398). http://portal.acm.org/citation.cfm?id=1101149.1101235&coll=ACM&dl=ACM&CFID=40134319&CFTOKEN=68049444

[4] DOAN, Anhai. MADHAVAN, Jayant. (2002). Learning to map between ontologies on the semantic web. Proceedings of the 11th international conference on World Wide Web. ACM. (pp. 662 - 673).

[5] http://portal.acm.org/citation.cfm?id=511446.511532&coll=ACM&dl=ACM&CFID=40138157&CFTOKEN=10917136

[6] ESCRIBANO. Manuel. (1995). Programación de Gráficos 3D. Addison Wesley, Delaware, 228p.

[7] GEER, David.(2005).Making 3D Technology Universal Phed by the IEEE Computer Society, 38, 08, (pp. 23-25).

[8] http://ieeexplore.ieee.org/iel5/2/32069/01492260.pdf

[9] GIBBINS, Nicholas. HARRIS, Stephen. (2003). Agent-based semantic web services. Proceedings of the 12th international conference on World Wide Web. ACM. (pp. 710 - 717).

[10] http://portal.acm.org/citation.cfm?id=775152.775251&coll=ACM&dl=ACM&CFID=41273239&CFTOKEN=77032621

[11] [8] HEAM, Donald. BAKER, Pauline. (2004). Computer Graphics with OpenGL. Published by Pearson Education. Indiana University. 895 p. http://dlist.sir.arizona.edu/1897/01/12_Zhang.pdf

[12] HUDSON. Alan, BAUCH. (2003). An introduction to the Xj3D toolkit. Proceedings of the eighth international conference on 3D Web technology. ACM. (pp. 190).

[13] http://portal.acm.org/citation.cfm?id=636593.636627&coll=ACM&dl=ACM&CFID=40134319&CFTOKEN=68049444

[14] KRAUSE. Dirk, (2008). Serious Games – The State of the Game, The relationship between virtual worlds and Web 3D. Proceedings Pixelpark white paper. (pp. 18).

[15] http://www.pixelpark.de/de/pixelpark/_ressourcen/attachments/publikationen/080529_White_Paper_Serious_Games_English_final.pdf

[16] MOUNTINHO, Gonzalo. PARISI, Tony. (2004). X3D programmable shaders. Proceedings of the ninth international conference on 3D Web technology. ACM. (pp. 99 - 108).

[17] http://portal.acm.org/citation.cfm?id=985040.985055&coll=ACM&dl=ACM&CFID=40134319&CFTOKEN=68049444

[18] NOY. Natalya, MCGUINNESS. Deborah. (2004). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University.

[19] OMG. (2005). Ontology Definition Metamodel. Third Revised Submission to OMG/RFP. http://www.omg.org/docs/ad/05-08-01.pdf

[20] PARISI. Salvatore, BAUCH. Jochen, BERSSENBRÜGGE Jan. (2007). Using Ontology to create 3D Aniations for Training Purposes. Deartment of Computer Integrated Manufacturing, Heinz Nixdorf Institute – University of Padrborn http://www.sersc.org/journals/IJSEIA/vol1_no1_2007/IJSEIA-2007-01-01-05.pdf

[21] PITTARELLO, Fabio. De FAVERI, Alessandro. (2006). Semantic description of 3D environments: a proposal based on web standards. Proceedings of the eleventh international conference on 3D web technology. ACM. (pp. 85-95).

[22] http://portal.acm.org/citation.cfm?id=1122591.1122603&coll=ACM&dl=ACM&CFID=41266025&CFTOKEN=53829823

[23] PLATDEPADUA. Joan. (2003). Programación en 3D con Java 3D. Alfaomega. México 303 p.

[24] PURBRICK, Jim. LENTCZNER, Mark. (2007). Second life: the world's biggest programming environment. Conference on Object Oriented Programming Systems Languages and Applications. ACM. (pp. 720).

[25] http://portal.acm.org/citation.cfm?id=1297846.1297849&jmp=abstract&coll=ACM&dl=ACM&CFID=40134319&CFTOKEN=68049444#abstract

[26] TRAUM, David. RICKEL, Jeff. (2002). Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. ACM. (pp. 766-773).

[27] http://portal.acm.org/citation.cfm?id=544862.544922&coll=ACM&dl=ACM&CFID=41266025&CFTOKEN=53829823

[28] ZHANG. Jane. (2007). Ontology and the Semantic Web. Proceedings of the North American Symposium on Knowledge Organization. Harvard University.

[29] ISO/IEC 14772-1:1997. (1997). The Virtual Reality Modeling Language (VRML) Functional specification and UTF-8 encoding. Web3D Consortium International Electrotechnical Commmission. Cambridge, Massachusetts.

[30] http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/

[31] ISO/IEC FCD 19776-1 Ed. 2. (2001). XML Schema. Web3D Consortium International Electrotechnical Commmission.Cambridge, Massachusetts.

[32] http://www.web3d.org/x3d/specifications/ISO-IEC-FCD-19776-1.2-X3DEncodings-XML/index.html

[33] ISO/IEC 14772-2:2004. (2004). The Virtual Reality Modeling Language (VRML) External authoring interface (EAI). International Web3D Consortium Electrotechnical Commmission. Cambridge, Massachusetts. http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772- VRML97/#ISO_IEC_14772Part2

[34] ISO/IEC 19775-1:2008. (2008). Extensible 3D (X3D) — Part 2: Scene access interface (SAI). International Web3D Consortium Electrotechnical Commmission. Cambridge, Massachusetts.

[35] http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/index.html

[36] Xj3D as a Java3D Loader. (2006). International Web3D Consortium. http://www.xj3d.org/tutorials/j3d_loader.html



**Holman Diego Bolívar** born in 1979, graduated as Computer Engineer from Catholic University in Bogotá D.C., Colombia in 2007, Currently studying PhD in Informatics Engineering at the Salamanca University, Director of Department of Informatics and Telecommunications of Faculty of Engineering of Catholic University in Bogotá D.C, Colombia.

**Dr. Ruben González Crespo** is a research manager in accessibility and usability in web environments for people with severe disabilities BS in Computer Management from the Polithecnical University of Madrid and Computer Engineering from the Pontifical University of Salamanca. Master in Project Management from the same university. PhD from the Pontifical University of Salamanca in 2008 and PhD Extraordinary Award given by the UPSA where is professor at the School of Computing (Madrid Campus) since 2005. Accredited teachers in all categories by the Agency for Quality Assessment of the Community of Madrid. Currently is the Director of the Master in Project Management, is professor of doctoral programs organized by the University and Master's programs in GIS, Software Engineering and Security.

**Dr. Oscar SanJuán Martínez:** He is Ph.D. from the Pontifical University of Salamanca in Computer Engineering. His research interest includes Object-Oriented technology, Web Engineering, Software Agents, Modeling Software with BPM, DSL and MDA. Electronics HNC from the University of Wales, graduate in International Studies from the SEI. He has translated and contributed to the review of multiple reference books in the field of Software Engineering. He has published over 80 articles in journals and national conferences and international prestige. In his teaching, he has taught over 30 seminars and conferences in Europe and Latin America on Intelligent Agents, Evolutionary systems, Biologically-inspired computing, Interactive Software, Multimedia, Games and Virtual Reality. In his business, he was Director of the Office of R & D of the Pontifical University of Salamanca, Coordinator of the Master in developing video game and entertainment software and technology consultant for "Vector Information Technologies" Spanish company dedicated to developing highly skilled computer projects in the area of Internet and Mobile Systems, Ubiquity and GIS. Currently works at the University of Oviedo where he developed his research in the field of Model Driven Development, intelligent systems, bio-inspired and interactive, accessibility, emerging systems and the future of the Internet (Smart-Objects and Objects-Net).