# SEMANTIC MANAGEMENT OF INTELLIGENT MULTI-AGENTS SYSTEMS IN A 3D ENVIRONMENT

Florian Béhé[1,2], Christophe Nicolle[1], Stéphane Galland[2] and Abder Koukam[2]
[1] *Laboratoire Electronique, Informatique et Image - UMR CNRS 5258, IUT Dijon Auxerre, Univsersité de Bourgogne BP 17867 21078 Dijon Cedex, France*
[2] *Laboratoire Systèmes et Transports, Université de Technologie de Belfort-Montbéliard, 90010 Belfort Cedex, France*
*florian.behe@checksem.fr, cnicolle@u-bourgogne.fr, {stephane.galland, abder.koukam}@utbm.fr*

Keywords: Ontology, Intelligent Multi-Agent Systems, Knowledge Acquisition, Industry Foundation Classes

Abstract: This paper presents a new approach combining the 3D elements composing the environment of mobile agents with semantic descriptors from Building Information Models. Our proposal is based on the IFC standard, which is used in the field of Civil Engineering to build digital models of buildings during the design phase. The semantic of IFC objects composing the 3D environment is used to select and set up 3D objects and elements of simulation scenarios. The result of this process dynamically generates the input files for the JaSIM environment that performs the simulation. These files deserve the representation of the virtual environment in which the simulation is running. It is represented by two separate files: a COLLADA file for the geometry and a RDF file for its semantics. Both files are generated according to the data extracted and selected from an IFC file by the user.

## 1 INTRODUCTION

The construction of a building is organized into several steps, from conception to completion. This collaborative work requires the involvement of multiple stakeholders throughout the life cycle of the building. Many standards have been defined in each trade involved in this life cycle. However, this cooperation still faces problems of heterogeneity (Vanlande et al., 2008). To resolve the first levels of heterogeneity (syntactic, structural and schematic) a standard called Industry Foundation Classes (IFC) was imposed in the world of civil engineering. The IFC standard was created in the late 90s by the International Alliance for Interoperability (now called buildingSMART). Its goal is to build a common data model for all the actors in the building industry to resolve problems of heterogeneity. The IFC standard is the kernel of the new generation of BIM. Several versions of the IFC have been published, because of the gradual increase of the covered area. The heart of the model was stabilized in 2005 and received the ISO certification as ISO / PAS 16739: 2005. The current release is 2x3TC1. It is an ASCII file containing all the elements of the described building and may be displayed in 3D. The next version of the IFC, named IFC4, should see its final version published at the same time as the international standard ISO/IS16739[1]. IFCs enable the exchange of data, either in the form of geometries, but as objects and their structures (walls, doors, windows, stairs ...). IFC files contain a description of all objects in the buildings and their links. The format also describes more abstract concepts such as schedules, activities, places, organizations, construction cost, etc.

Gradually, publishers of CAD software and generally all software for civil engineering (structural analysis, air conditioning ...) develop translation functions of their proprietary language to the IFC standard (Cruz and Nicolle, 2008). Beyond a simple format for interoperability, the IFCs, which describe both the geometric representation of objects in the buildings and their semantics can be used to manage ontology aggregating all knowledge of the buildings (contractual document, pictures, dashboards...) (Cruz and Nicolle, 2005; Vanlande et al., 2003; Cruz and Nicolle, 2010). This semantic translation of building information can also be used to achieve new goals in qualifying the use of the building from the design phase. The combined use of 3D and semantics of IFC is perfectly suited to the construction of an intelligent multi-agent system to simulate the behavior of mobile entities in a 3D environment. This system helps to qualify the use of the

[1] The name of the ISO standard for IFC4

environment and especially to improve, using semantics, the existing processes in the field of multi-agent located in a 3D environment. This paper discusses an ongoing research on the design of a multi-agent system based on a semantic indexing of IFC objects. This indexing process allows to build dynamically an informed environment (composed of 3D objects semantically indexed) and intelligent agents that can react to the environment according to a context of use.

The new section presents the existing works on using semantics in Multi-Agents Systems. Section 3 overviews the proposed the principle of the environment representation and the architecture of the semantic-based environment generation tool. Finally, Section 4 concludes this work and draws several perspectives.

## 2 BACKGROUND

This paper is located in the domain of the simulation of buildings flows with situated multi-agent systems. A multi-agent system (MAS) is a system composed of multiple interacting intelligent software agents (Ferber, 1995). Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. A multi-agent system is situated when the agents are immersed inside an environment. In the domain of buildings simulation, an agent is assumed to be a pedestrian, or any object that owns an autonomous decision-making process. The environment is then everything that is not an agent in the buildings.

Three different points of view may be adopted to study the notion of environment in situated MAS (Weyns et al., 2007): (i) the part of the system which is outside the community of the agents; (ii) the medium for coordination among these agents; or (iii) the running infrastructure or platform. Weyns et al. distinguish between the physical environment and the communication environment (Weyns et al., 2006). The physical environment provides the laws, rules, constraints and policies that govern and support the physical existence of agents and the other entities. In the rest of this paper, only this aspect of the environment is taken.

Several problems may be solved to properly implement an environment: its topological and geometrical description, its dynamics, and the meanings of each object and zones in the environment. The two first points are addressed by the JaSIM environment model (Galland et al., 2009). The last point is addressed by both the integration of semantics in the environment and modification of the agent's algorithms that uses them. Most of the approaches found in the literature are based on the tagging process of the environment.

- **Tagging is often used as a kind of semantic**
  The concept of tagging consists in placing some tags in the environment to inform agents on various subjects. Basically, tags are considered as objects placed in the scene, but they do not have a physical presence. They are invisible for the viewer of the scene and are only being seen by the agents. Our proposal includes to describe through tags the usage of some places, i.e. where an agent can sit, pass through, etc. Lugrin and Cavazza (Lugrin and Cavazza, 2007) places various tags on a single object. These tags are linked between and can also represent information. For example, a glass will be represented by a geometry and two tags. The geometry will deserve its representation and also dimensions in the simulated world. The first tag is a "containing" tag that will notify agents that the object on which this tag is applied can contain some things. The second tag is an "opening" tag that will represent the fact that the object is opened, and then the inside of this object is accessible. These two tags are linked together to represent the fact that if an agent interact with the "opening" tag, that will affect the state of the "containing" one. Finally, a last link is made between the "opening" tag and the environment to represent the fact that the opening is accessible directly from the environment space.

  In this proposal, the evolution of an object is done by modifying the tags and links of this object. For example, if the glass is clogged, the link between the opening tag and the environment is deleted and this tag is no more accessible from the environment.

  Yersin et al. (Yersin et al., 2005) propose to implement a navigation graph with the help of tagging. This approach consists in covering the maximum amount of navigable areas with a minimum number of discs covering these areas and not overlapping any obstacle. These discs overlap themselves and form a navigation — center to center — graph in which agents are sure to do not collide with any obstacle. Moreover, these discs have labels to define the name of the zone in which they are located. These labels are very useful to select a target without knowing its position in the environment.

- **Using roles in addition of tagging**
  In a similar way, De Paiva et al. (de Paiva et al., 2005) propose to put labels on areas in order to associate a name with a position. But in opposite

to the previous works, the agents can only go to a location according to its name and not its position. In addition, the authors propose to assign roles to the agents and make them evolving in the environment according to their roles and the current simulation time. For example, a kid is at "school" at 11am and a working adult is at "office" at the same time. This approach is useful to only have agents that are in a given place, i.e. an agent which is supposed to be a student is not in the headquarters of a company, except if his behavior needs it.

- **Misc.**
  Gutierrez et al. (Gutierrez et al., 2005) propose to use semantic to describe the interactions among the agents and the environment objects to use them with various physical devices (mouse, keyboard, etc.). This approach can be extended to describe, for example, the interactions of agents representing disabled people.

In this paper, the definition of the environment is extended with semantics and agent behaviors are adapted to use these semantic informations. Indeed with semantics, the result of the simulation of the individuals is not only based on the geometric features but also on the information embedded in the spatial and temporal contexts of the simulation.

# 3 PROPOSAL

This section presents the principle and architecture of our proposal, which generates the four JaSIM input files from an BIM/IFC file.

## 3.1 Previous JaSIM state

Simulation of autonomous entities in a complex urban system requires dedicated software models. JaSIM platform (Galland et al., 2009) integrates components which are required to simulation complex environments in 1D, 2D and 3D (in particular, particular it is used for simulation in virtual reality). This platform integrates several models to reproduce human visual perception in a virtual environment and endogenous behavior of this environment. Thus simulated entities can use the JaSIM platform to perceive and act in a situated system.

To run a simulation, JaSIM requires two kinds of input files. A third kind of file may be needed if the simulation has to be displayed.

- **SFG file:** A SFG file is a XML-based file that can be seen as the scenario description of the simulation. It will contain various informations about the simulation and the environment: definitions of the places, spawning areas for the agents, goals, way points, stochastic generation laws, etc. An example of SFG configuration file can be seen in Listing 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE simulation PUBLIC "-//set.utbm.fr//DTD JaSimConfigurationFile3d v7.0//EN" "/fr/utbm/set/jasim/controller/config/jasim-config-3d-7.0.dtd">
<simulation id="68552aab-e71a-44d4-b321-9d9ec9b937fa"
      name="DEMO-SIMULATION"
            date="2009-12-23"
            authors="GALLAND Stephane"
            version="0.1"
            dtdversion="7.0"
            description="Simulation of Pedestrians">

  <time type="step" unit="millisecond" timeStep="500"
      />

  <environment dimension="3d">
    <places>
      <place id="e19fd4d1-73f0-4283-82b0-0d950d53bb62"
          name="MainHall">
        <groundEnvironment id="f3977258-9d02-4b71-8201-
            b96d7e684e62" type="constant">
          <indoorGround minx="-128.66" miny="-118.23"
              maxx="128.66" maxy="118.23" z="0"
              semantic=""/>
        </groundEnvironment>
      </place>
    </places>
    <portals>
    </portals>
  </environment>

  <spawners>
    <spawner type="area"
        id="7a6d7aca-4621-4efa-97b3-a5c084aff20f"
        name="SPAWNER 1"
        x="-8.5209"
        y="-91.8171"
        z="0"
        width="10" height="10" startAngle="0" endAngle="
            6.283185308"
        place="e19fd4d1-73f0-4283-82b0-0d950d53bb62">
      <entity budget="20" agentType="fr.utbm.set.jasim.
          demos.pedestrians.holon.PedestrianHolon" >
        <frustums>
          <frustum type="sphere" eyePosition="1.8"
              farDistance="10" />
        </frustums>

        <generationLaw class="fr.utbm.set.jasim.spawn.
            ConstantSpawningLaw">
          <lawParam name="value" value= "2000"/>
        </generationLaw>
      </entity>
    </spawner>
  </spawners>
</simulation>
```

Listing 1: Example of a SFG file

- **Precomputed structures of the environment:** The position of all the static/immobile objects in the environment are precomputed and saved inside a file containing the corresponding serialized Java tree. This tree data structure permits to efficiently localize the objects. In the same way a second file may be provided for all the agents which may be spawned at the start of the simulation. In both trees, simple semantics can be associated to objects such as "door", "window", etc.
- **3D model file:** The last file supported by the JaSIM platform is optional and is only used when

the simulation is rendered in 3D. It contains all the geometries of the visible objects which may be rendered to the final used. The format of this file should correspond to a standard 3D file format such as COLLADA® or 3D Studio®. All the geometry provided by this file may corresponds to an perceivable object for the agents and previously described the Java serialized files.

## 3.2 JaSIM evolution

In previous version, JaSIM allowed to represent information in various files depending on the kind of data that needs to be stored. One of the problems that can be identified in this kind of representation is that some piece of information related to the environment are stored into the scenario file. For example, places, for the "place-portal" principle, are stored in the scenario file despite it is more related to the environment than the scenario.

One of the goals of the presented work is to split correctly scenario and environment information. The SFG file is thus preserved, but it only contains information about simulation's execution parameters, such as agents' types, spawning areas, etc. In addition to this file, a Resource Description File (RDF) file is introduced in order to manage environment's semantics. Its geometry is, for its part, supported by the COLLADA file. A reference to IFC elements is kept in the COLLADA structure in order to retrieve semantics of the objects in the RDF file.

## 3.3 Principle

IFC files contain a huge amount of information, more than needed for a multi-agent simulation. The IFC files contain a complete description of the building as illustrated by Figure 1.

```
1      ISO-10303-21;
2      HEADER;
3      FILE_DESCRIPTION(('IFC2X_KEOD2011'),'2;1');
4      FILE_NAME('C:\\Documents and Settings\\ MASTS2011.ifc','2011-05-04T13:35:28',(''),('
5      FILE SCHEMA(('IFC2X3'));
...
47296  #47702=IFCAXIS2PLACEMENT3D(#47701,$,$);
47297  #47703=IFCLOCALPLACEMENT(#13278,#47702);
47298  #47704=IFCWINDOW('2UFNUHdW5EN98F5gpuXGRb',#33,'fenetre 2v speciale 3:fenetre 2v sp
47299  #47705=IFCPROPERTYSINGLEVALUE('Reference',$,IFCLABEL('fenetre 2v speciale 3:fenet
47300  #47706=IFCPROPERTYSET('1Pt6oFbGv6OeJxUuZ5xjnW',#33,'Pset_WindowCommon',$,(#47705,#
47301  #47707=IFCRELDEFINESBYPROPERTIES('205QD8saD2KP3Sq$e8d8pn',#33,$,$,(#47704),#47706)
47302  #47708=IFCPROPERTYSINGLEVALUE('Retrait',$,IFCLENGTHMEASURE(0.1),$);
...
100834 #101492=IFCPRESENTATIONLAYERASSIGNMENT('A-ROOFOTM',$,(#68338,#75188),$);
100835 #101493=IFCPRESENTATIONLAYERASSIGNMENT('A-WALLCWM',$,(#11871,#11877,#11947,#1195
100836 #101494=IFCPRESENTATIONLAYERASSIGNMENT('A-WALLMBM',$,(#213,#335,#390,#417,#536,#5
100837 ENDSEC;
100838 END-ISO-10303-21;
100839
```

Figure 1: Screenshot of an IFC file. This ASCII file represents a complete description of a building in 100838 lines.

That is why it is required to perform a selection on what piece of information will be extracted from IFC files.

The work presented in this paper allows to generate automatically all files required by JaSIM to perform a simulation. It relies on a kernel-centered architecture (Section 3.4) which first implementation is described here. The developed software allows to generate the four files that are needed by the JaSIM's evolution.

The SFG file only contains informations about the simulation's scenario (agents' spawners, goals, etc.) as mentioned in Section 3.2. The COLLADA file will contain geometry information and, for each object, the Global Unique IDentifier (GUID) of the IFC objects that was used to its generation. The preprocessed structure that describes mobiles entities is empty since the presented approach only focuses on the environment part of MAS. Finally, a RDF file is generated containing semantics about extracted IFC objects. This semantics contains data directly extracted from the IFC file and also semantics or constraints added by the user. It is linked with the geometrical representation by using the GUID extracted from the IFC (the same as the one postponed in the COLLADA file).

As already mentioned in the beginning of this Section, IFC files contain a lot of information, and MAS do not need all of them. Our kernel can automatically filter these pieces of information relying on an ontology schema that will describe what is needed to be kept for a MAS and transform IFC objects into RDF elements. Moreover, our kernel also enables the user to export or not elements in the environment according to a specific context of use. Elements that are not selected to be exported will thus not be present neither in the COLLADA file nor in the RDF file.

Once this operation is done, the system extracts geometrical information from the IFC in order to display to the user the building in a semantic way (i.e. under a tree form that will correspond to the building structure: building, storeys, places, etc. as shown in Figure 2). The 3D representation of the geometry is also shown to the user. The GUID, stored both in RDF elements and geometry extracted from IFC file, allows to link the tree-representation and the 3D representation of the building.

The user can manipulate the RDF elements and to associate them to SFG concepts or to add constraints on certain elements or concepts. SFG concepts are, for most, positioned in a certain location in the environment. In this case, the kernel will only use geometrical information extracted from the IFC in order to place correctly the scenario's element in the environment. In some case, semantics is also used, but only to set SFG elements' parameters. For example, there are two kinds of agents' spawners: the first one

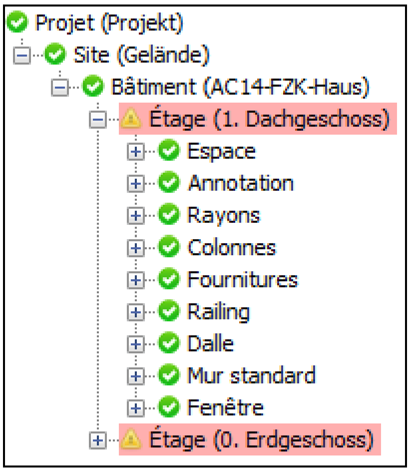


Figure 2: Example of a semantic representation of an IFC

spawns agents always on the same point in space and the second one spawns agent randomly positioned on a given surface. In this case, the semantics extracted from the IFC will be used to determine what kind an agents' spawner that needs to be used. For example, if the concept of agents' spawner is associated to an IfcSpace, the spawner will be set as an area spawner that will be able to spawn agents anywhere in the given space. In the same manner, if the concept is associated to an IfcDoor (for example the door that represents the main entrance) the spawner will only spawn agents on the exact position of this door.

In addition to performing associations to SFG concepts, the user can also add some semantic constraints to RDF elements. To illustrate this principle, we use the case of a simulation in which two kinds of agents can progress, respectively named type A and type B. The user can, for example, add constraints on some door to specify that this instance of a door can only be opened by type A agents. When the simulation will be executed, type A agents will thus see these doors as crossable elements while type B agents will see these doors as obstacles. Another example of constraint is that only type A agents will be allowed to lock or unlock certain instances of doors, type B agents' permissions to use the passages will thus evolve during the run of the simulation. In all these cases, the COLLADA file will contain the geometrical representation of the door, and the RDF file will contain the fact that it is a door (and all information that stem from it such as "ability to open it", etc.) and also constraints on these properties, such as the fact that the door can effectively by opened, but only by type A agents.

Finally, each action made by the user is stored by a profile management module. This module will retain, for each action, the semantic context and the action of the user. These profiles are used to propose to the user to try to perform automatically actions that the user could do to set a simulation.

## 3.4 Architecture

The main part of the proposed software is about the loading the IFC file (Figure 3). Two libraries are used. The first one is dedicated to load the IFC structure in memory according to their specifications. The second one loads the geometry in a directly usable shape structure and not in geometry description that need to be processed.

A central kernel was developed to manage and launch these two libraries, and link them to get the shape representation of a given IFC element from the specification-compliant memory representation. This kernel also manages the generation of the JaSIM scenario elements. It also allows communication with the Graphical User Interface (GUI) to update links and other data according to user actions.

This kernel is not dependent on the GUI and can thus be used with another GUI or other user interaction manner. The application structure is shown on Figure 4. As shown on mentioned Figure, the kernel loads the IFC file in order to extract IFC objects and geometry that are put respectively in the display tree and three-dimensional display. The developed GUI allows to interact with the IFC structure or selection and only calls kernel functionalities. It can thus be easily replaced by other user interaction methods.

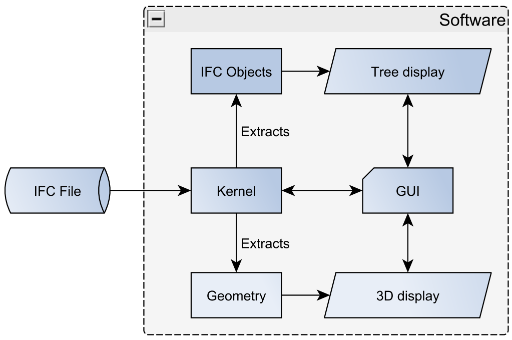


Figure 4: Kernel centered architecture of our proposal

To perform the JaSIM file generation, the kernel uses file generating modules shown in Figure 5. The COLLADA® export module has been developed in the scope of being reusable in any application. It takes as input a Java3D scene graph (built from the IFC files) and generate the COLLADA® schema that corresponds to the given structure. Each Java3D element can be associated to some extra data that allows us to keep a relation between the IFC objects

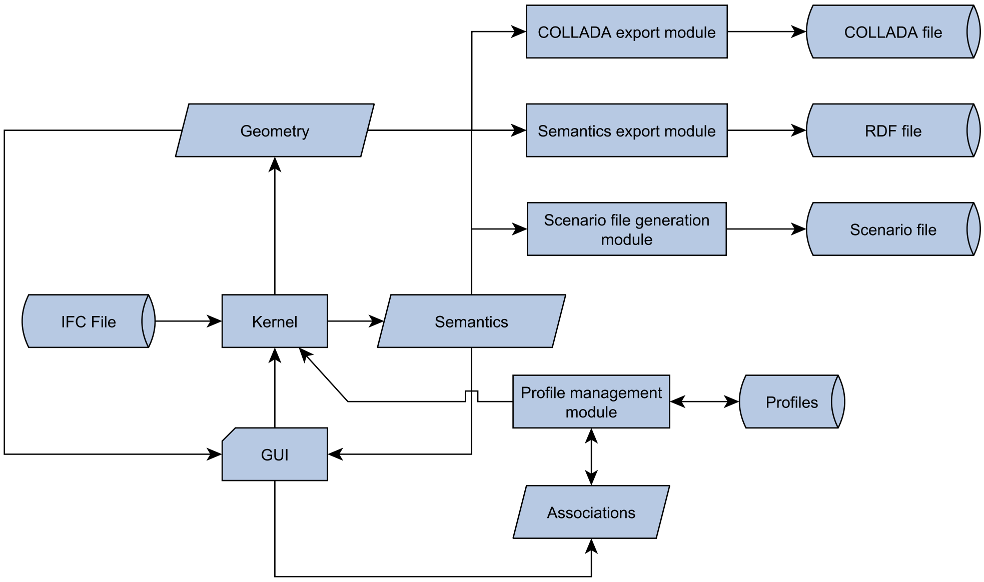


Figure 3: Global chart of our proposal

and the purely geometrical objects, i.e. GUID of IFC objects is postponed in the COLLADA structure. The SFG (JaSIM Scenario Configuration) generator module parses the IFC data structure of our kernel to retrieve associations to SFG concepts and uses IFC elements' description to get the geometrical position and then generate the SFG file.

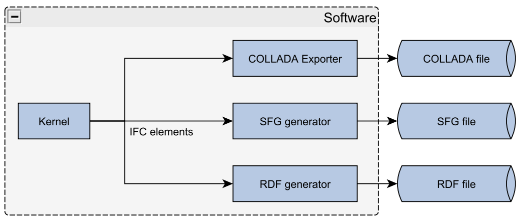


Figure 5: File generation process

Finally, profile management is done by the kernel. It updates the profile at each selection and re-use the profile on the IFC loading to perform associations. The profile update process is illustrated on Figure 6.

# 4 CONCLUSION & FUTURE WORKS

This paper discusses an ongoing research on the design of a multi-agent system based on a semantic indexing of IFC objects. This paper presents the use of IFC files to generate a MAS environment. These

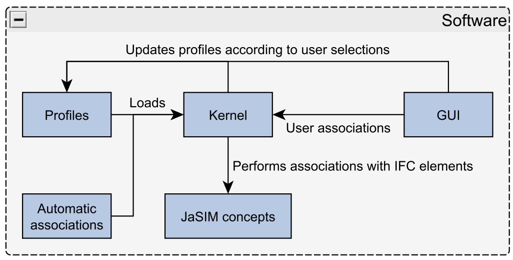


Figure 6: Profile management process

files are based on objects' description merging semantics and 3D geometries. Our work is done in order to test the viability of the usage of IFC files in MAS simulation domain.

Using IFC files as a starting point in MAS simulation is beneficial according to the quality of the data, which are always up-to-date. A simulation can be executed as soon as the building is designed to test the quality and the level of compliance of such design. Moreover, using BIM as a MAS input also enables to bring a high level of semantic information to the environment that can be used by the agents in turn. The focus of this paper is on the semantic analysis of IFC objects composing the 3D environment. It is used to select and set up 3D objects and elements of simulation scenarios. The result of this process generates semi-automatically and dynamically input files to the JaSIM environment that performs the simulation at the end.

Our next goal is to improve our software to get a better automation in the selection process. We are developing an ontology-based IFC. The first results of this work are the translation of the building objects into COLLADA and RDF files as depicted in Figure 7.

This ontology will be used to describe associations to support more parameters and build rules, including the management of the context. For example, in our environment, doors and windows are both opening elements and these elements can be seen by the agents as crossable elements to get out from a room. Nevertheless, is a window still a valid exit if this window is on the 6th floor? Rules will make it possible to perform a better classification of the elements and to apply several restrictions and filters on the associations. Finally, this architecture will help to apply fine associations and to improve the environment, or at least the simulation scenario.

## ACKNOWLEDGEMENTS

This work is funded by the region of Franche Comté and receive grants from a cooperative project between the Franche Comté and the Bourgogne. Thanks to the Checksem and SeT teams and Jordan Simonot for his help in the COLLADA export part of this project.

## REFERENCES

Cruz, C. and Nicolle, C. (2005). 3d reconstruction based on semantic information for architectural applications. Schriftenreihe Informations- und Messtechnick.

Cruz, C. and Nicolle, C. (2008). Cad software and interoperability. *Encyclopedia of Information Science and Technology Second Edition (8-Volume Set)*.

Cruz, C. and Nicolle, C. (2010). Active3d: Semantic and multimedia merging for facility management. *6th International Conference on Web Information Systems and Technologies*.

de Paiva, D., Vieira, R., and Musse, S. (2005). Ontology-based crowd simulation for normal life situations. In *Computer Graphics International 2005*, pages 221–226. IEEE.

Ferber, J. (1995). *Les Systèmes Multi-Agents : vers une intelligence collective*. InterEditions.

Galland, S., Gaud, N., Demange, J., and Koukam, A. (2009). Environment model for multiagent-based simulation of 3d urban systems. In *the 7th European Workshop on Multi-Agent Systems*, Ayia Napa, Cyprus.

Gutierrez, M., Thalmann, D., and Vexo, F. (2005). Semantic virtual environments with adaptive multimodal interfaces. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pages 277–283. IEEE.

Lugrin, J. and Cavazza, M. (2007). Making sense of virtual environments: action representation, grounding and common sense. In *Proceedings of the International Conference on Intelligent User Interfaces, Honolulu, Hawaii, USA*, pages 225–234.

Vanlande, R., Cruz, C., and Nicolle, C. (2003). Managing ifc for civil engineering projects. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 179–181. ACM.

Vanlande, R., Nicolle, C., and Cruz, C. (2008). IFC and building lifecycle management. *Automation in Construction*, 18(1):70–78.

Weyns, D., Ominici, A., and Odell, J. (2007). Environment as a first-class abstraction in multiagent systems. *Journal on Autonomous Agents and Multi-Agent Systems*, 14(1):5–30.

Weyns, D., Parunak, H., Michel, F., Holvoët, T., and Ferber, J. (2006). Environments for multiagent systems state-of-the-art and research challenges. In *Third International Workshop E4MAS*, volume 4389, pages 1–47. Springer.

Yersin, B., Maım, J., de Heras Ciechomski, P., Schertenleib, S., and Thalmann, D. (2005). Steering a virtual crowd based on a semantically augmented navigation graph. In *Proc. The First International Workshop on Crowd Simulation (V-CROWDS'05), Lausanne, Switzerland*, pages 169–178. Citeseer.

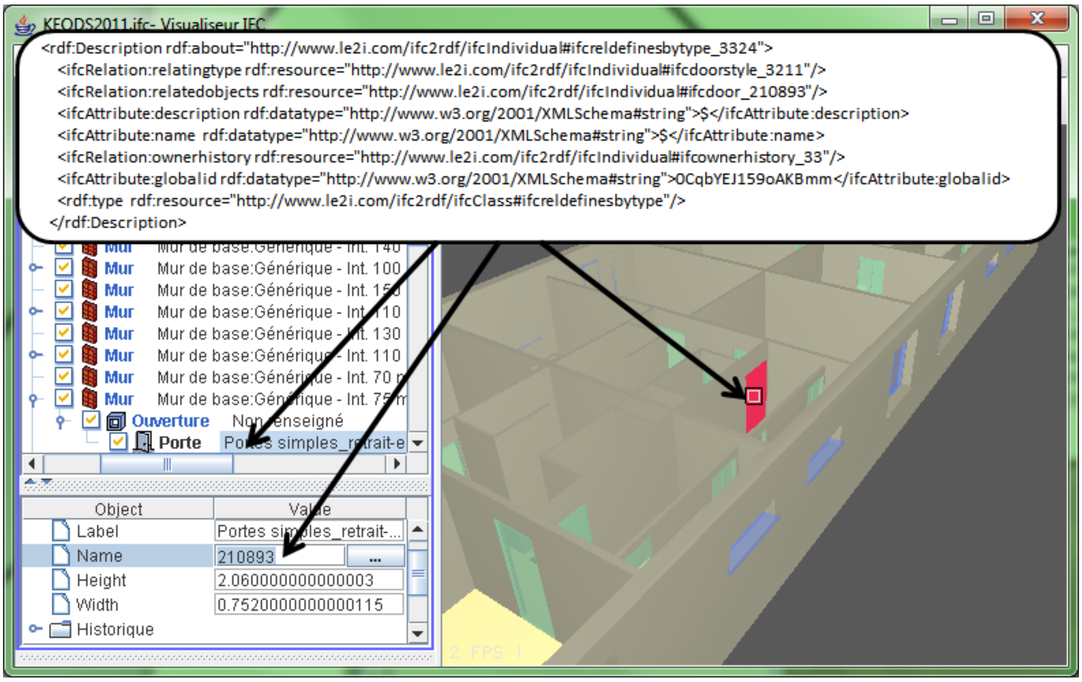


Figure 7: Example of RDF representation of IFC Object