Managing IFC in Civil Engineering Projects

Renaud Vanlande
Active3D-Lab
21 Avenue Albert Camus
BP 66606, 21066 Dijon, France
+(00) 380 539 595
r.vanlande@active3d.fr

Christophe Cruz
Active3D-Lab
21 Avenue Albert Camus
BP 66606, 21066 Dijon, France
+(00) 380 539 595
c.cruz@active3d.fr

Christophe Nicolle
Le2i – UMR CNRS 5158
Université de Bourgogne
BP 47870 – 21078 Dijon, France
Telephone number, incl. country code
cnicolle@u-bourgogne.fr

ABSTRACT

The "Industrial Foundation Classes" (IFC) is an ISO norm to define all components of a building in a civil engineering project. IFC files are textual files whose size can reach 100 megabytes. Several IFC files can coexist on the same civil engineering project. Due to their size, their handling and sharing is a complex task. In this paper, we present an approach to automatically identify business objects in the IFC files and simplify their visualization and manipulation on the Internet. We construct an IFC Viewer which transforms the IFC file into a XML IFC tree manipulated through the 3D visualization of the building. The IFC Viewer composed a web-based platform called ACTIVe3D BUILD SERVER. This platform lets geographically dispersed project participants—from architects to electricians to plumbers—directly use and exchange project documents in a centralized virtual environment during the life cycle of a civil engineering project.

Keywords
IFC, 3D, XML, Semantic, DBMS

1. INTRODUCTION

In civil engineering projects, dedicated computer programs can't exchange data directly, even when the same team uses them. Buildings therefore take longer to design and build and cost more to construct and operate. Information sharing should be the starting point in applying information technology to building design, construction, and use. Information sharing requires a software environment in which computer programs can exchange data automatically regardless of software and data location. Toward this goal, the International Alliance of Interoperability proposed a standard called IFC that specifies object representations for construction projects [4]. Industry foundation classes (IFC) include object specifications, or *classes*, and provide a useful structure for data sharing among applications. A door IFC, for example, isn't just a simple collection of lines and geometric primitives recognized as a door; it is "aware" that it's a door and has a door's attributes. However, many construction project teams don't exploit the IFC. No central IFC database exists, nor do tools for IFC analysis and comparison or visualization during construction.

The goal of our research is a technology by which IFC files are analyzed and directly transformed both in a tree and in a 3D visualization. This transformation is an important step in the manipulation of the IFC on the net by final user. The transformation reduce the size problem of the IFC text files where the description of one building floor can reach 15 megabytes size.

To address this problem, a research team in computer science from the LE2Ì laboratory at the University of Bourgogne, in collaboration with the "Groupe Archimen", carried out research on the handling of languages derived from XML, in particular IFC and X3D languages. The "Groupe Archimen" is a civil engineering and facility management firm. The results of this collaboration are the creation of a private company called ACTIVe3D-LAB (http://www.active3d.net) which employed ten people (three administrative staff members, eight developers, and five University of Bourgogne researchers including two PhD students) and the development of the ACTIVe3D BUILD SERVER to improve civil engineering project management.

Currently, 87 people use the application daily and manage ten projects. The commercial launching of the ACTIVe3D BUILD SERVER will take place at the beginning of November, during the BATIMAT'2003 show in Paris.

The rest of this paper is organized as follows. Section 2 describes our approach to the problem. Section 3 explains our methodology. Section 4 described our implementation.

2. APPROACH

A civil engineering project consists of phases codified by the profession (Primary studies, pre-project summary, pre-project detailed, project, and so on). Every phase generates a set of specific-scale plans, or text documents that the architect validates with the client, administrative authorities, or financial partners before moving to the next phase. Only the architect can manipulate all the phase of a project. For example, the architect generally validates the "pre-project summary" phase by obtaining planning permission from administrative authorities. At this level, administrative authorities can visualize and print plans contained in the previous phase ("Primary studies") at the 1/200 scale and, upon completing their work, indicate the planning permission electronically. All authorized participants then receive an automatically generated email indicating the revised document's availability. Next, in the "pre-project summary" phase, the architect must define the space structures' real measurements at the 1/50 scale. At this level, other team members can consult the plans and use specific building software to specify the electrical network, plumbing, and so on. Once the plans are completed, the architect can correct them and reevaluate project costs as needed. At each step of the project, plans are created, exchanged and modified by all members of a project. All the exchanges are based on the IFC files which are the numerical translation of the architect plan.

The "Industrial Foundation Classes" (IFC) is an ISO norm to define all components of a building in a civil engineering project. IFC files are textual files whose size can reach 100 megabytes. Several IFC files can coexist on the same civil engineering project. Due to their size, their handling and sharing is a complex task. An example of IFC file structure is given in script 1. This file described a building with more than 111000 business objects (one lines per object).

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('ArchiCAD generated IFC file.'), '2;1');
FILE_NAME ('Karlstr.IFC', '2002-06-19T15:48:48', ('Architect'), ('Building Designer Office'), 'PreProc - IFC Toolbox Version 2.x (00/11/07)', 'Windows System', 'The authorising person.');
FILE_SCHEMA (('IFC2X_FINAL'));
ENDSEC;
DATA;
#1 = IFCORGANIZATION ('GS', 'Graphisoft', 'Graphisoft', $, $);
#3 = IFCPERSON ($, 'Undefined', $, $, $, $, $, $);
#4 = IFCORGANIZATION ($, 'OrganizationName', $, $, $);
#5 = IFCPERSONANDORGANIZATION (#3, #4, $);
#7 = IFCSIUNIT (*, .LENGTHUNIT., $, .METRE.);
…..
#111029 = IFCRELCONTAINEDINSPATIALSTRUCTURE
('25wKeWex98fQp5Pukf_Ilc', #6, 'BuildingStoryContainer',
'BuildingStoryContainer for Building Elelements', (#111007), #110989);
#111030 = IFCRELAGGREGATES ('216Bv$yJj3tQjFeDohe6fQ', #6, 'BuildingContainer', 'BuildingContainer for BuildigStories', #30, (#34, #16236, #29699, #56800, #62077, #67336, #72633, #91702, #110989));
#111031 = IFCRELAGGREGATES ('17XMUtNDr8FeFMtR6rOcy5', #6, 'SiteContainer', 'SiteContainer For Buildings', #28, (#30));
#111032 = IFCRELAGGREGATES ('0pMN8yq8vDRfwN_tnJREKC', #6, 'ProjectContainer', 'ProjectContainer for Sites', #26, (#28));
ENDSEC;
END-ISO-10303-21;
```

Script 1. Example of IFC file

Our goal is to simplify the manipulation of these IFC files by developing an IFC Viewer. The IFC Viewer composed a web-based platform called ACTIVe3D BUILD SERVER. This platform lets geographically dispersed project participants—from architects to electricians to plumbers—directly use and exchange project documents in a centralized virtual environment during the life cycle of a civil engineering project.

3. METHODOLOGY

IFC files are made of objects and connections between these objects. The attributes in the objects, describe the "business semantic" of the objects. The connections between objects are represented by "relationship elements". IFC Files are cyclic graphs. This structure makes them difficult to handle. In the sharing of IFC files, one of the most significant questions for the handling of the IFC is "where to cut in the graph?"

To address this problem we have developed a methodology based on graph analysis and tree classification. The first step is a loading of the IFC file in the computer memory. During this loading, each object and connection is analyzed and converted from the source file into acyclic graphs called contextual trees. This process is done using business rules. An example of a business rule is "*a door is in an opening element in a wall*". Resulting from this process, the main tree is the geometrical contextual tree which contains the topological relations between the various objects. Other contextual trees are built starting from the IFC files, such as the contextual tree of capacity defining the object composition (a building contains two floors, a floors contains beams, walls, and so on.) This step finishes when the trees replaced the IFC file in the memory model. The figure 1 displays a snapshot containing the view of a tree of capacity.

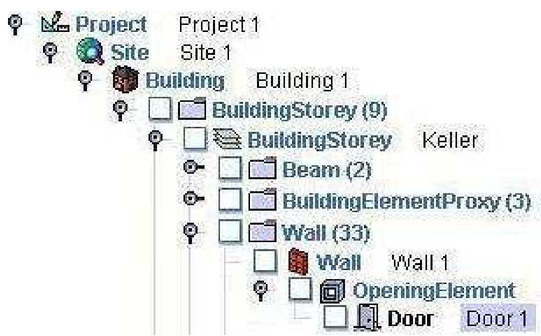

Figure 1. IFC Tree of capacity

The second step is dedicated to the 3D modeling [1, 2, 3, 5]. In this step, all the geometry defined in IFC trees is converted into triangular surface model [7]. During this conversion, the 3D objects are associates with the GID. The GID is the general identifier used to identify each business object of an IFC file. In the script 1, the GID of the IFCRELAGGREGATES object is #111032. This GID is used to link the 3D visualization with the information stored in the databases. All insertion of new data in any base is reference by a GID correspondent to an IFC object. All trees generated in the platform are XML trees [9]. These trees and the component elements are stored in a relational database and manipulated using the SQL. From this database and the GID, all types of information can be attached to the 3D visualization of business Object.

4. DEVELOPMENT

We're running the ACTIVe3D BUILD SERVER on a Dell dual-processor server (Pentium 4, 2.4 GHz, 512 Mbytes DDR RAM, 180 Gbytes disk space, Red Hat Linux 7.2 OS). An Oracle 9i relational database management system (RDBMS) manages the data layer. We developed the broadcast and behavioral layers in Java using Sun Microsystems' JDK 1.4.1) to facilitate development and ACTIVe3D BUILD SERVER porting to other operating systems [5]. We use Microstar's Sax and Sherry's Dom parsers for all XML streams. We're currently developing the 3D visualization module called the "IFC Viewer" in GL4JAVA using the NetBeans 3.4.1 development environment. The IFC Viewer represents both the IFC Tree and the 3D visualization of a Building. All objects composing the building are managed through the tree or dynamically using the 3D scenes, activating object by clicking on it.

The IFC Viewer was certified ISO/PAS 16739 in May 2003 by the IAI. It allows the user to see the complete description of the building with an IFC tree view. Moreover, it lets participants move around in the building being designed and obtain information about the objects that compose it. This 3D is "semantically" dynamic. The structuring of the scene is calculated in real time from the IFC database according to geometrical features (distance from the point of view, object hides...) and semantic features (rights users, type of user, stage of visualization of the project...).

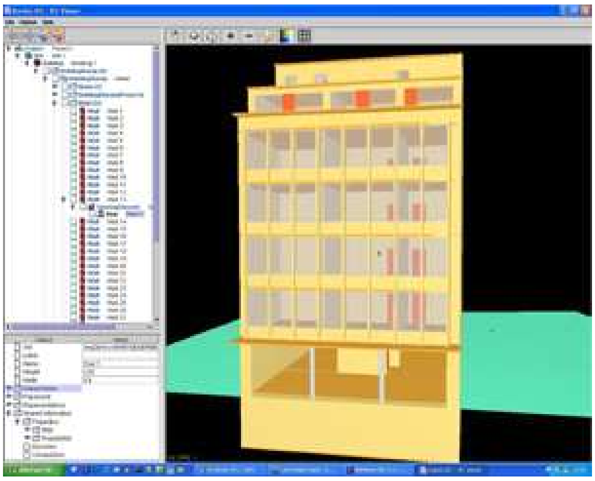

Figure 2. Snapshot of the IFC Viewer

The figure 2 presents a snapshot of the IFC Viewer. The entire platform is developed using web-services that are associated dynamically in a collaborative application at the user connection according to the user rights [6].

We evaluated the IFC loading, conversion and visualization performance. Table 1 shows the results of the tests that measured

File size		4.4 Mb	7 Mb	22 Mb
Active3D	Load file	12''	18''	1'9''
	Compute geometries	11''	1'23''	7'

Table 1. Measure of loading, converting and 3D visualization

5. ACKNOWLEDGMENTS

We thank the ANVAR and the ANRT for their support. We also thank many people at the Active3D-LAB, especially Olivier Gaudard, Florent Barth, Sylvain Bonandrini, Samuel Desgouilles and Sebastien Delune for their time and their cooperation.

6. REFERENCES

[1] Boon-Hee Kim, Jun Hwang, Young-Chan Kim, The design of high-level database access method in a Web-based 3D object authoring tool, The Fourth International Conference on Distributed Communities on the Web, 3-5 April 2002, Sydney, Australia

[2] Bowman, D., Davis, E., Badre, A., and Hodges, L. Maintaining Spatial Orientation during Travel in an Immersive Virtual Environment. Presence: Teleoperators and Virtual Environments, vol. 8, no.6, 1999, pp. 618-631

[3] Howard Abrams, Kent Watsen, Michael Zyda: Three-tiered interest management for large-scale virtual environments, VRST 1998, pp. 125-129

[4] International Alliance for Interoperability. Ifc, industry foundation class. http://www.iai-international.org/

[5] Kornel Szabo, Peter Stucki, Patrick Aschwanden, Thomas Ohler, Renato Pajarola, Peter Widmayer, A Virtual Reality Based System Environment for Intuitive Walk-Throughs and Exploration of Large-Scale Tourist Information. Proceedings of the Enter95 Conference, 1995, pp. 10-15

[6] Christophe Nicolle, Kokou Yétongnon, XML Based Toolkits for the Interoperability of Web Information Systems, Chapter book, Web-Enabled Systems Integration: Challenges X Practice, 2002

[7] Rémi Ronfard, Jarek Rossignac: Full-range Approximation of Triangulated Polyhedra. Computer Graphics Forum 15(3): 67-76 (1996)

[8] Alex Vakaloudis, Babis Theodoulidis: The Storage and Querying of 3D Objects for the Dynamic Composition of VRML Worlds. IV 1998: 267

[9] Extensible Markup Language (XML): http://www.w3.org/XML/