

IEEE

Multimedia

<http://computer.org>

OCTOBER-DECEMBER 2002



IEEE
COMPUTER
SOCIETY

IEEE

The Active3D-Build: A Web-Based Civil Engineering Platform

Christophe Cruz,
Christophe
Nicolle, and Marc
Neveu
Bourgogne
University, France

Computer programs for building design, analysis, and maintenance typically 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.

We developed the Active3D-Build (<http://www.active3d.net>) to improve civil engineering project management. This Web-based platform lets geographically dispersed project participants—from architects to electricians to plumbers—directly use and exchange project documents in a centralized virtual environment using a simple Web browser. A 3D visualization lets participants move around in the building being designed and obtain information about the objects that compose it.

The Active3D-Lab is the software development arm of the Archimede Group (<http://www.archimede-groupe.com/>), a civil engineering and facility management firm. The Active3D-Lab team includes three administrative staff members, eight developers, and three University of Bourgogne researchers. We're already marketing this product, and several enterprises use it in their civil engineering projects.

The problem

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 (http://www.iai-international.org/iai_international/) proposed a standard that specifies object representations for construction projects. Industry foundation classes (IFCs) 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.

Our platform derives from research on how to represent computer graphics modeling semantics within a relational database.¹ Semantic modeling involves two levels. The first entails creating one database for the IFC and another for 3D scene compositions. The second level associates the IFC semantics with 3D database elements. To permit this association, we represent all information (IFC, 3D scenes, and management information) as Extensible Modeling Language (XML) trees.

Civil engineering project composition

A civil engineering project consists of phases codified by the profession, including APS (Avant Projet Sommaire, preparatory project summary), APD (Avant Projet Détaillé, preparatory project detailed), DET (Direction d'Exécution de Travaux, direction of work's execution), and so on. Every phase generates files, sets of specific-scale plans, and text documents that the architect validates with the client, administrative authorities, or financial partners before moving to the next phase. Figure 1 represents these phases in a tree view that displays ramifications according to project participants' rights.

Only the architect can manipulate the entire tree. For example, the architect generally validates the APS phase by obtaining planning permission from the client. At this level, administrative authorities can visualize and print plans contained in the previous phase (*Etudes Préliminaires* in Figure 1) at the 1/200 scale and, upon completing their work, indicate the planning permission electronically in the tree's APS node. All



Figure 1. The Active3D-Build interface, showing project phases and documents.

authorized participants then receive an automatically generated email indicating the revised document's availability.

Next, in the APD 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 software such as AutoDesk's Architectural Desktop or Nemetschek's Allplan 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.

The Active3D-Build architecture

We used XML to develop Active3D-Build. Derived from Standard Generalized Markup Language (SGML), XML has become a universal language for representing information. Its tree structure permits easy manipulation of information contained in its nodes, known as tags. Extensible Stylesheet Language (XSL), an XML language that translates source data into appropriate data display formats, permits this manipulation.

We've studied two XML schemas, XMLifc and X3D (<http://www.web3d.org/x3d.html>), from which we generated a database creation script and a set of Java classes using XSL. The Java classes serve to insert, extract, and manipulate information contained in the associated databases. To create a database, we first define mapping rules from a schema to an entity-relationship schema, then

translate this schema into a relational schema. Finally, at the physical level, we optimize the database structure.² The X3D schema defines 103 tags to represent 3D scenes. We've defined 103 tables corresponding to the X3D tags and some other tables representing connections between tags.

The Active3D-Build architecture consists of three distinct XML-based layers (Figure 2). The data layer stores information generated during the civil engineering project. This layer also stores XML documents in the specific relational databases. The broadcast layer provides the distributed communication support of information sharing through the Internet, while the behavioral layer provides information-processing mechanisms. This layer consists of a set of Java classes that manipulate

several XML streams between databases.

The data layer includes three relational databases: management, IFC, and X3D. The management database contains information about participants (such as roles, rights, and actions in the project) and documents exchanged during the project (creation date, release date, information updates, deletion date, and so on). The IFC and X3D databases store information usually kept in ASCII files. Our team developed an analysis method to convert these files into a relational format (primary keys, simple and mono-valued attributes, foreign keys, and so on).

These databases use more than 500 tables to store management, IFC, and X3D data. Using databases instead of files increases data manipulability and avoids file size problems when transferring files through the network. The IFC database is generated from the IAI's XMLifc language, a translation of the XML schema containing IFC file production rules corresponding to the STEP (standard for the exchange of product model data) specification (<http://www.nist.gov/sc4/www/stepdocs.htm>). The X3D database is generated from the Web3D Consortium's X3D schema, the XML translation of the Virtual Reality Modeling Language (VRML).

The behavioral layer consists of three Java packages. The first, the management interface module (MI in Figure 2), permits manipulation of management information. We defined a Java class for

each table of the management database's relational schema. The database (extraction process) or user-inserted data (insertion process) can initialize Java objects. The second Java package, the IFC interface module (II in Figure 2), permits IFC manipulation. The third package, the scene authoring interface module (SAI in Figure 2), permits X3D scene manipulation. We developed our own SAI module using the most recent X3D schema (<http://www.web3d.org/TaskGroups/x3d/sai/SceneAccessInterface.html>).

We used a client-server structure for the broadcast layer, using a Java applet (on the client side) and a Java server. This architecture opens a secure object stream between the server and the client. Both the client and the server use a common Active3D interface that regroups the three interfaces (MI, II, and SAI). They can directly exchange information with the shared objects through this object stream.

Development environment and evaluation

We're running the Active3D-Build on a Dell dual-processor server (Pentium 3, 500 MHz, 256 Mbytes 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) to facilitate development and Active3D-Build porting to other operating systems. We're currently developing the 3D visualization module in Java

3D using the JBuilder 5 Enterprise Java development environment. We use Microstar's Sax and Sherry's Dom parsers for all XML streams.

We evaluated the X3D database's insertion and extraction performance using VRML source files. We used no database or network optimizations. Table 1 (next page) shows the results of the tests that measured

- the time needed to convert a VRML file into a X3D document, create the corresponding insertion script, and execute this script in the database; and
- the time needed to extract data and build a X3D document from the database.

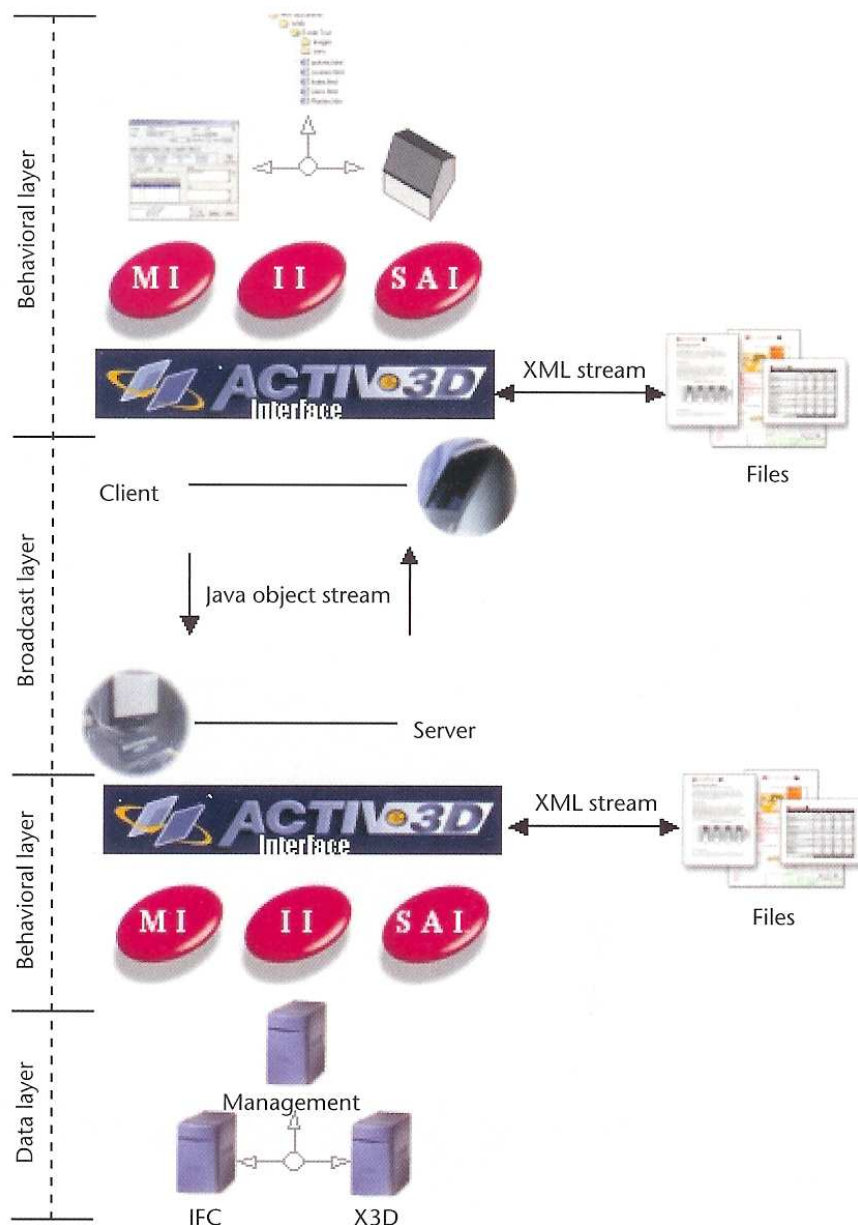


Figure 2. The Active3D-Build architecture.

These results show that processing time depends on two criteria: file size and scene structure. For example, although scene 4 contains fewer tags than scene 3, it's larger because a set of points rather than primitives (cube, sphere, and so on) describes the graphic elements, and some elements possess more than 15,000 characters.

Future work

We plan to implement file optimization by detection of primitives, network optimization by adding a cache zone on the server to reduce client delay, and database optimization by switching off the Oracle optimizer and using the Oracle cache system to extract data. We're also developing another platform, the Active3D-

Facilities, which will enable facilities management using a Web-based 3D environment. **MM**

References

1. C. Cruz, C. Nicolle, and M. Neveu, "Active3D: Interrogation de Scènes 3D en SQL" (Active3D: Interrogation of 3D Scenes in SQL), *Proc. Medianet, Accès Intelligent aux Documents Multimédia sur l'Internet* (Intelligent Access to Multimedia Documents on the Internet, in French), Hermes Science, Paris, 2002.
2. C. Nicolle and K. Yétongnon, "XML Enabled Meta-modeling and Tools for Cooperative Information Systems," *Proc. 2nd Int'l Conf. Electronic Commerce and Web Technologies (EC-Web)*, Lecture Notes in Computer Science 2115, Springer, Berlin, 2001.

Table 1. Performance of the X3D database.

Scene	File Size (Kbytes)	Insertion Script Creation (seconds)	Insertion Script Execution (seconds)	X3D Scene Extraction (seconds)
1	315	1	6	15
2	1962	4	25	80
3	4039	5	30	140
4	8767	40	25	125

Readers may contact Christophe Cruz at Active3D-Lab, 21 Ave. Albert Camus, BP 66606, 21066 Dijon CEDEX, France, email c.cruz@active3d.net.

Readers may contact editor Tiziana Catarci at the Dept. of Information Systems, Univ. of Rome "La Sapienza," Via Salara 113, 00198 Rome, Italy, email catarci@dis.uniroma1.it.

IEEE Annals of the History of Computing

January-March

Contributions to Computing

Learn how yesterday's contributions have shaped today's computing world. Featured articles cover topics such as the first assembly-line production of a digital computer, a manager's account of working at Computer Sciences Corporation in the mid-1960s, and searching for tractable ways of reasoning about programs.

April-June

Evolution of Digital Computers

Digital computers have evolved into powerful workstations, but that wasn't always the case. Read about one man's foray into digital computing as well as the making of the MCM/70 micro-computer. This issue also features articles on the glory days of Datamation and the history of the sector, an analog calculating instrument.

2003 Editorial Calendar

July-September

Historical Reconstructions

This special issue is devoted to recording and recounting efforts to preserve computing practice through physical reconstruction, restoration, and simulation. In addition to providing accounts of such projects through substantive articles, the issue will serve as a digest of projects and initiatives as well as societies and organizations active in these areas.

October-September

Women in Computing

Since the days of Ada Lovelace, women have played an important role in the history of computing, and this role has received increasing attention in recent years. Scholarship in this area has begun to move beyond simply demonstrating women's presence in the history of computing to considering how computing and gender constructs have shaped one another over time.

<http://computer.org/annals>