

Modélisation sémantique pour la génération dynamique de scènes 3D Application aux projets architecturaux

Christophe Cruz, Renaud Vanlande, Christophe Nicolle

*Le2I – UMR CNRS 5158
Université de Bourgogne
B.P. 47870 – 21078 Dijon
France*

{c.cruz, r.vanlande}@active3d.net, cnicolle@u-bourgogne.fr

Résumé — Cet article présente une méthode de structuration, de manipulation et de visualisation des données sémantiques dans le cadre d'un projet de génie civil. Le projet ACTIVE3D a pour objectif la manipulation des documents IFC contenant l'information de conception des bâtiments. La taille des fichiers rend leur manipulation et leur transport difficile. Cet article décrit les processus utilisés pour la gestion sémantique d'arbre de la scène 3D. Une fois certaines contraintes résolues, la structure permet de visualiser les informations 3D du projet.

Mots Clés — Sémantique, 3D, Architecture, Interopérabilité.

1 INTRODUCTION

Les utilisateurs d'outils de conception assistée par ordinateur créent souvent des modèles géométriques 3D complexes de très grandes tailles. Il n'est pas rare que les plans d'un bateau, d'un avion ou d'une structure architecturale dépassent la taille du giga-octet. Le projet GigaWalk est un système de rendu permettant d'afficher des projets de CAO de plus de 10 millions de polygones. L'exemple le plus frappant est le projet de conception du tanker DoubleEagle constitué de plus de 4 giga-octets de données, soit 82 millions de triangles et 127 mille objets. Le taux de calculs est de 11 à 50 images par seconde, ce qui permet de naviguer en temps réel à travers la maquette numérique [1]. La conception basée sur la simulation de ces données ne peut avoir un apport bénéfique sans la possibilité de générer un affichage interactif à travers une visite virtuelle du modèle. Beaucoup de techniques d'optimisations et d'accélération pour l'affichage interactif ont été développées pour ce type de données. Celle-ci inclut les calculs de visibilité, la simplification d'objet et la représentation basée sur les images. Toutes ces techniques ont été combinées avec succès pour le rendu de données spécifiques qui incluent des modèles architecturaux [2] et dans les modèles urbains [3]. L'intérêt majeur des maquettes

numériques dans les métiers de la conception prend toute son ampleur lors des prises de décisions au cours des phases de conception. Les maquettes sont utilisées pour les applications dans le cadre de tests d'intégrations d'équipements, d'accessibilités et d'espace dans les domaines allant de l'aéronautique à l'automobile en passant par l'architecture. L'apparition de sites Web permettant un travail collaboratif sur les projets a amélioré la qualité des prototypages, du suivi du projet et des phases de tests [4]. Ces systèmes permettent aux architectes de prototyper un bâtiment et d'itérer des modifications avec les clients sur des détails [5]. Le projet CAVALCADE a pour objectif de faire avancer l'état de l'art dans le domaine des plateformes collaborative 3D grâce à une architecture distribuée permettant à des équipes distantes de collaborer sur la conception, les tests, la validation et l'échange de documents [6].

Néanmoins, ces plateformes collaboratives sur Internet ne permettent pas la manipulation géométrique d'une grande quantité de polygones en temps réel sans un temps de pré calcul prohibitif. Une voie pour résoudre ce problème est de structurer la scène 3D selon des critères sémantiques et non plus à partir des seuls critères géométriques.

Dans cet article, nous proposons une méthode qui permet d'associer de la sémantique métier aux objets représentés dans des modèles géométriques 3D complexes. Cette association fournit des arbres contextuels qui associe dynamiquement, à l'aide de règles, une connaissance métier à des groupes de polygones pour générer des objets métiers 3D. Le critère dynamique concerne la génération automatique de la scène 3D à partir des fichiers de CAO et la possibilité de manipuler chaque objet métier dans la scène et d'y associer des documents ou des fonctions spécifiques. Dans le cadre d'un projet de construction d'un bâtiment, il est très pratique de voir le plan en 3D généré à partir du fichier CAO et de pouvoir sélectionner une porte pour obtenir la facture et activer le service Web qui fournit une description du produit à partir du catalogue fournisseur.

L'utilisation des arbres contextuels limite la complexité du modèle géométrique ce qui permet son utilisation sur Internet dans des délais raisonnables. Cette méthode a été testée et implémentée dans l'application Active3D Build Server. Cette application est une plateforme collaborative qui permet la conduite de projets du bâtiment au travers d'échanges d'information via une interface 3D dynamique.

2 APPROCHE

Un projet d'ingénierie civil se compose de phases codifiées par la profession (études primaires, avant-projet sommaire, avant-projet détaillé, projet, et ainsi de suite). Chaque phase produit un ensemble de plans et de documents textes que l'architecte valide avec le client, avec les organismes administratifs, ou avec les associés financiers avant de commencer la phase suivante. Seul l'architecte peut manipuler toute la phase d'un projet. Par exemple, l'architecte valide généralement la phase récapitulative de l'avant-projet en obtenant le permis de construire des organismes administratifs. À ce niveau, les organismes administratifs peuvent visualiser et imprimer des plans contenus dans la phase précédente ("études primaires") à l'échelle de 1/200 et, lorsqu'ils ont terminé leurs travaux, ils indiquent le planning de construction par voie électronique. Tous les participants autorisés reçoivent alors un email automatiquement indiquant la disponibilité du document révisé. Après, dans la phase récapitulative de l'"avant-projet", l'architecte doit définir les espaces, et ceux en indiquant la taille à l'échelle de 1/50. À ce niveau, d'autres membres d'équipe peuvent consulter les plans et employer un logiciel spécifique de conception pour indiquer le réseau électrique, la tuyauterie, et ainsi de suite. Une fois que les plans sont accomplis, l'architecte peut les

corriger et réévaluer les coûts du projet si nécessaires. À chaque étape du projet, des plans sont créés, échangés et modifiés par tous les membres d'un projet. Tous les échanges sont basés sur les fichiers IFC [12] qui sont la traduction numérique du plan d'architecte. Les IFC "Industrial Foundation Classes" définie tous les composants d'un bâtiment d'un projet en ingénierie civil. Les fichiers IFC sont des fichiers textes qui peuvent atteindre la taille de 100Mo. Plusieurs fichiers IFC peuvent coexister sur un même projet. Due à la taille des fichiers, leurs manipulations et leurs transports restent une tâche difficile. Le Script 1 est un exemple de la structure d'un fichier IFC. Ce fichier est un bâtiment avec la définition de plus de 111000 objets métier (une ligne par objet).

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('ACTIVE3D generated IFC file.',
'2;1');
FILE_NAME ('active3D.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 ('A3D', 'Active3D', 'Active3D', $,
$);
#3 = IFCPERSON ($, 'Undefined', $, $, $, $, $);
#4 = IFCORGANIZATION ($, 'OrganizationName', $, $, $);
#5 = IFCPERSONANDORGANIZATION (#3, #4, $);
#7 = IFCSIUNIT (*, .LENGTHUNIT, $, .METRE.);
.....
#111029 = IFCRELCONTAINEDINSPATIALSTRUCTURE
('25wKeWex98fQp5Pukf_llc', #6, 'BuildingStoryContainer',
'BuildingStoryContainer for Building Elelements', (#111007),
#110989);
#111030 = IFCRELAGGREGATES
('216Bv$yJ3tQjFeDohe6fQ', #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. Exemple de fichier IFC

Le fichier IFC est un graphe cyclique. Chaque noeud du graphe possède des informations sémantiques partielles. Pour obtenir des informations sémantiques complètes sur un objet métier, il faut analyser plusieurs noeuds qui ne sont pas forcément directement liés. Cette structure rend l'analyse du fichier difficile. Pour résoudre ce problème, nous avons décomposé le graphe cyclique selon un modèle de représentation des IFC. Ce modèle nous permet de représenter les informations contenues

dans les fichiers IFC indépendamment de la structure du fichier initiale. Ensuite, nous avons développé un ensemble de règles qui nous permettent de générer des arbres contextuels au format XML. Ces arbres correspondent à une réorganisation de l'information en fonction de critères sémantiques. Le modèle et les règles sont présentés dans les sections suivantes.

3 MODELE DE REPRESENTATION DES IFC

La conception du modèle de représentation des IFC (appelé Modèle IFC-RM) est réalisée en fonction de plusieurs contraintes. Les deux principales sont 1/ la nécessité de conserver une interopérabilité avec les logiciels de CAO, et cela même pour l'échange partiel, 2/ la possibilité d'assurer la persistance des objets métiers définis au sein d'un SGBD relationnel. Pour répondre à ces contraintes, nous avons défini un ensemble de concepts permettant de représenter plus clairement la sémantique des fichiers IFC.

Le modèle que nous avons défini est basé sur les concepts des modèles orientés objets. En effet, les principaux composants des fichiers IFC sont des Classes, liées par des relations d'héritage. A titre d'exemple, la classe IFCBuildingElement possède un ensemble de classes filles composées de IFCWall, IFCBeam, IFCWindow, IFCDoor, etc. L'utilisation de l'héritage facilite la définition des classes, mais cette hiérarchie n'apparaît pas dans les instances de ce modèle. En effet, seules les classes IFCWall, IFCBeam, etc, apparaissent dans les fichiers IFC, mais la classe IFCBuildingElement a pour seul intérêt de les regrouper de manière sémantique.

Un élément par définition est une instance de classes IFC. Les éléments possèdent un identifiant unique appelé *GID*. Il existe trois types d'éléments : les éléments « sémantiques », les éléments « relations » et les éléments « ressources ». Les éléments « sémantiques » décrivent les objets physiques ou fonctionnels du domaine de la construction. La définition formelle d'un élément « sémantiques » est :

$$ES = (ts, Att_n[])$$

Où *ts* est le type de l'élément et *Att_n[]* une liste de références vers d'autres attributs.

Les éléments « relations » décrivent un lien entre plusieurs éléments sémantiques. Les cardinalités associées à ces relations sont de la forme (1..1) et (1..n). La cardinalité (1..1) est utilisée pour définir l'élément père qui est considéré comme un « conteneur ». La cardinalité (1..n) est utilisée pour définir des éléments fils. Ces éléments sont considérés comme des contenus. (cf. fig. 1). La définition formelle d'un élément « relation » est :

$$ER = (tr, ES_{père}, ES_n[], Att_m[])$$

Où, *tr* est le type de l'élément, *ES_{père}* est une référence vers un élément sémantique père, *ES_n[]* une liste de référence d'éléments sémantiques fils et *Att_m[]* une liste de références vers d'autres attributs.

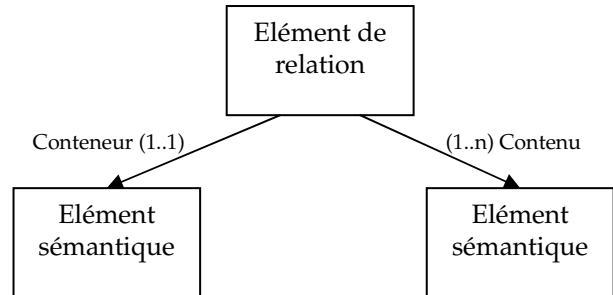


fig. 1 : Relation entre éléments du modèle IFC

Les éléments « sémantiques » et les éléments « relations » possèdent une arborescence d'attributs. Cette arborescence d'attributs est fixée par la grammaire IFC. La définition formelle d'un attribut est :

$$Att = (ta, v, Att_n[])$$

$$Att_n[] = (Att_1, Att_2, Att_3, \dots, Att_n)$$

Où, *ta* est le type sémantique de l'attribut *Att*, *v* est la valeur de *Att*, *Att_n[]* représente une liste de références vers d'autres attributs.

Les éléments « ressources » sont utilisés par tous les autres éléments pour décrire les propriétés du projet. Une ressource possède des attributs simples et d'autres ressources. Cette manière de définir les propriétés des éléments hiérarchise l'information (cf. fig. 2). A titre d'exemple, la ressource « représentation » d'un élément sémantique est constituée d'une ressource « placement » composée de trois attributs (Origine, Vecteur X et Vecteur Z), et d'une autre ressource « représentation » qui est de type « représentation géométrique ». Ces dernières peuvent être des boîtes englobantes, des modèles de surfaces, des représentations par les bords, ou des représentations mappées.

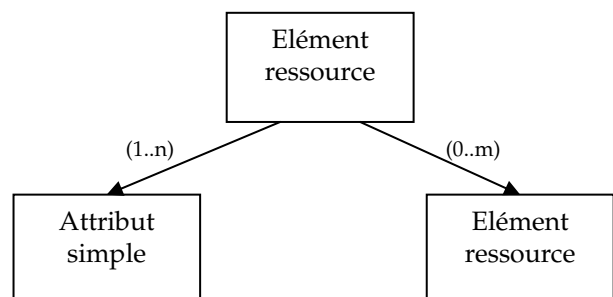


fig. 2 : Relation entre ressources du modèle IFC

Cette description succincte du modèle IFC-RM permet de montrer la structure cyclique des liens entre plusieurs éléments sémantiques. En effet, deux éléments sémantiques peuvent se référencer par deux types de relations différentes.

Une fois que chaque composant du fichier IFC est identifié en terme d'élément sémantique, d'éléments relation et d'éléments ressources, la deuxième étape peut commencer. Les arbres contextuels au format XML sont générés et leurs contenus sont placés dans un modèle mémoire IFC.

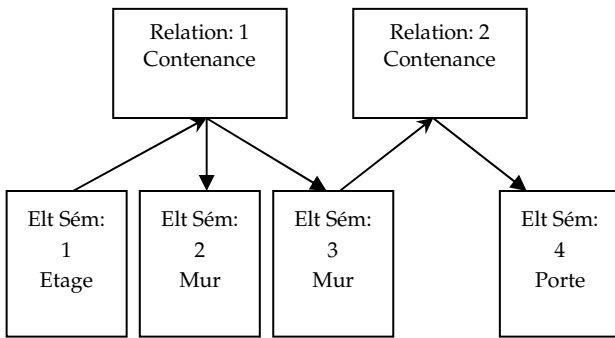


fig. 3 : Relation entre éléments du modèle IFC

4 GENERATIONS D'ARBRE CONTEXTUEL

Cette section décrit les règles qui sont employées pour la construction des arbres contextuels. Pour cela, nous utilisons la notion de contexte. Tous les éléments « relations » possèdent un type qui permet de distinguer les classes de relation. Toutes les relations qui sont de types identiques appartiennent à la même classe. Une classe de relation donnera les moyens de créer un contexte qui est une arborescence d'éléments sémantiques (cf. fig. 3).

Par exemple, le contexte de contenance va définir les éléments contenus par un élément père comme un bâtiment possède deux étages et que chaque étage possède des murs et certains murs possèdent des portes ainsi que des fenêtres. La classe de relation contenance permet de créer l'arbre contextuel des éléments qui correspond au contexte de contenance (cf. fig. 4).

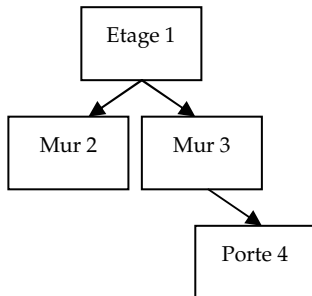


fig. 4 : Arbre contextuel de contenance

Pour Recanati [8] "La signification d'un mot comme le 'i' est une fonction qui nous porte d'un contexte d'expression à la valeur sémantique du mot dans ce contexte, où la valeur sémantique (référence de 'i') est ce que le mot contribue à la proposition exprimée par l'expression."

$$Y = f(X)$$

Où

$f()$ est le mot.

X est le contexte

Y la valeur sémantique du mot dans ce contexte

Par analogie, le mot $f_{is}()$ est l'élément sémantique de type 'ts' et Y est l'ensemble des valeurs d'attributs du mot pris en compte par le contexte. Le contexte $C = (TR[], TA[])$ est un ensemble de types éléments « relations » $TR[] = (tr_1, tr_3, \dots, tr_p)$ ainsi qu'une liste de types d'attributs $TA[] = (ta_3, ta_6, \dots, ta_q)$ pour filtrer les éléments « sémantiques ». La valeur sémantique Y peut donc appartenir à l'ensemble vide, même si le mot $f_{is}()$ existe dans le modèle mémoire IFC. La fonction du contexte apparaît quand nous nous occupons de quantificateurs logiques. La gamme et l'interprétation des quantificateurs dépendent du contexte. Par exemple, le quantificateur « tout » généralement ne fait pas appel à tous les objets, mais seulement à ceux d'un certain type dans le domaine particulier, déterminé par le contexte.

$$C = (TR[], TA[]) = ((tr_1, tr_3, \dots, tr_p), (ta_3, ta_6, \dots, ta_q))$$

$$Y = f_{is}(C)$$

$$Y = f_{is}((tr_1, tr_3, \dots, tr_p), (ta_3, ta_6, \dots, ta_q))$$

$$Y = (v_1, v_2, v_3, \dots, v_n)$$

où

$$Y = (\emptyset)$$

En langage naturel, Y est la liste des valeurs des attributs du mots $f_{is}()$ des attributs référencés par le contexte. De plus, le mot est sélectionné si et seulement si il est référencé par un élément « relations » appartenant à la liste des éléments de relation du contexte. $C = (TR[], TA[])$ est un contexte si l'arbre résultant des éléments « sémantiques » sélectionnés suivants les relations définie par la liste des éléments de relation du contexte est un arbre non cyclique. C'est-à-dire que tout élément « sémantique » de l'arbre ne peut être ancêtre et descendant de tout autre élément « sémantique » à la fois.

$$C = ((tr_1, tr_3, \dots, tr_p), (ta_3, ta_6, \dots, ta_q))$$

$$ES[] = (ES_1, ES_2, \dots, ES_n)$$

$$ES_1 = (ts_1, Att_1[])$$

$$ES_2 = (ts_2, Att_2[])$$

$$\dots$$

$$ES_n = (ts_n, Att_n[]) \\ Arb_{Sém} = (C, ES[])$$

Dans le cas présent, nous voulons afficher les informations géométriques dans un navigateur 3D, alors pour générer le graphe de scène, l'arbre contextuel géométrique est nécessaire.

$$C_{Géo} = (TR_{topologie}[], TA_{géométrie}[]) \\ ES_{tous}[] = (U ES_{projet}) \\ Arb_{Géo} = (C_{Géo}, ES_{tous}[])$$

Cet exemple a pour résultat lors de la création du graphe de scène la visualisation de tous les éléments géométriques du modèle IFC-RM, mais il est possible de restreindre la liste des éléments en spécifiant les types d'éléments sémantiques que le contexte doit prendre en compte.

$$ES_{mtd}[] = (ES_{mur}, ES_{tuyau}, ES_{dalle}) \\ Arb_{GéoMtd}(C_{Géo}, ES_{mtd}[])$$

Cet arbre sémantique a pour effet lors de la création du graphe de scène la visualisation de tous les éléments « sémantiques » de type mur, tuyau et dalle. Ce contexte géométrique correspond à celui d'un plombier, les éléments de charpente ou les éléments électriques n'apparaissent pas. Le contexte, dans le cadre d'un ciblage plus précis, peut affiner la sélection des éléments sémantiques. Lorsque l'on définit les types d'attributs, les éléments « sémantiques » qui ne les possèdent pas ne sont pas pris en compte et n'apparaissent donc pas dans l'arbre contextuel géométrique. A titre d'exemple, la recherche des éléments « sémantiques » de type mur doit se faire sur les murs possédant un enduit.

$$C_{GéoEnd} = (TR_{topologie}[], (TA_{géométrie}[], TA_{enduit})) \\ ES_{murs}[] = (U ES_{mur}) \\ Arb_{GéoEnd}(C_{GéoEnd}, ES_{murs}[])$$

Une fois l'arbre contextuel géométrique réalisé, la recherche d'une valeur précise d'un type d'enduit est réalisée par le parcours récursif de l'arbre. Il existe différentes manières de manipuler ces informations, une de celle-ci fait l'objet de la section suivante.

5 MANIPULATION DES ARBRES CONTEXTUELS

La conception d'un projet d'ingénierie civil évolue au cours du temps. Les différentes parties sont de plus en plus détaillées et contiennent de plus en plus d'informations. Cette évolution se traduit par une mise à jour du modèle mémoire IFC. Pour conserver une cohérence générale, ces mises à jour

doivent être cadrées par des règles d'insertions ou de suppressions. Un intervenant lorsqu'il désire mettre à jour la maquette numérique a deux possibilités. La première consiste à modifier les attributs d'un élément « sémantiques » ou un élément « relations ». L'intervention aura pour effet d'ajouter ou de supprimer des attributs de l'arbre hiérarchique des ressources de l'élément. La deuxième possibilité consiste à modifier un ou plusieurs arbres contextuels. Cette mise à jour de la hiérarchie des éléments du projet est réalisable de manière simple seulement si l'information se trouve sous la forme d'arbre. En effet, il est difficile pour un intervenant de travailler sur plusieurs vues du projet en même temps et de valider la cohérence de ceux-ci. Par conséquent, toute manipulation du modèle mémoire IFC est réalisée par l'intermédiaire des arbres contextuels. De plus, les arbres contextuels sont des références vers les éléments du modèle mémoire IFC donc l'ajout d'élément consiste à ajouter des références d'un modèle mémoire IFC source à un modèle mémoire IFC cible, limitant ainsi la copie de données.

Dans le cas de l'insertion, l'intervenant sélectionne un contexte de travail puis sélectionne un sous arbre du modèle mémoire IFC par rapport à un contexte choisi exemple un étage complet d'un bâtiment. Une fois cette étape effectuée, un processus se met en place pour sélectionner l'arbre contextuel géométrique, cet arbre est l'arbre par défaut qui permet de visualiser en 3D le modèle mémoire IFC. Parallèlement, un processus crée le même arbre contextuel pour le modèle mémoire IFC cible, et une fois cet arbre construit, ce processus recherche le nœud père du sous arbre à ajouter commun aux deux modèles mémoires IFC. En effet, pour garder la cohérence de la structure, il n'est possible d'ajouter un arbre que si le nœud père du modèle mémoire IFC source possède le même GID que le nœud père du modèle mémoire IFC du nœud cible. De cette façon, si les deux modèles-mémoires sont cohérents avant modification alors ils le resteront après modification. Dans le cas de la suppression, l'intervenant sélectionne aussi l'arbre contextuel, car la suppression de certaines informations peut rompre un arbre contextuel en deux et éventuellement perdre la référence à un sous arbre involontairement.

L'avantage de cette méthode réside dans le fait que seules les références mémoires sont manipulées et donc il est possible d'ajouter des références à un modèle mémoire IFC partiel. En effet, la base de données peut contenir plusieurs centaines de Mo de données et donc seule une partie peut être chargée en mémoire. Si cette partie est mise à jour alors les nouveaux éléments sont mis dans la base de données et de nouvelles références sont créées dans les éléments contenus dans la base de données. Cela

signifie que les processus manipule les ressources fichiers et ou la base de données de la même manière (cf. fig. 5).

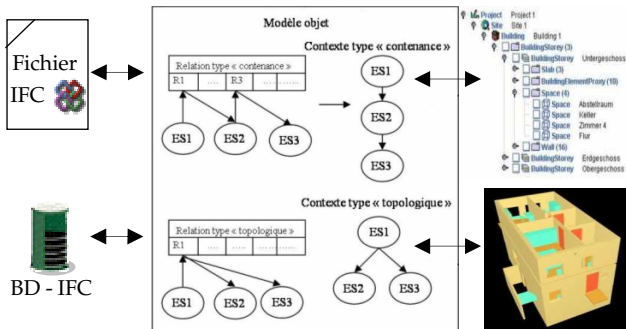


fig. 5 : Modèle mémoire IFC et arbres contextuels

6 IMPLEMENTATION

Actuellement le serveur ACTIVE3D BUILD fonctionne sur une machine biprocesseur Dell (Pentium 4, 2.4 GHz, 512 Mo de DDR-RAM, un espace disque de 180 Go, Linux Red Hat 7.2). La couche de données est gérée par la base de données relationnelle Oracle 9i. Nous avons développé la couche réseau ainsi que la couche gestion à l'aide de Java (JDK 1.4.1 de Sun Microsystems) pour faciliter le développement et pour porter le serveur ACTIVE3D BUILD sur d'autres systèmes d'exploitation. Nous employons Sax de Microstar's et le parseur Dom Sherry's pour tous les flux XML. Nous développons actuellement le module de visualisation appelé "IFC Viewer" en employant les bibliothèques GL4JAVA pour la partie 3D dans l'environnement de développement NetBeans 3.4.1. L'IFC Viewer est composé de trois parties. L'arbre en haut à gauche représente l'arbre contextuel de contenance, le navigateur 3D permet de visualiser les informations de l'arbre contextuel géométrique et le composant graphique en bas à gauche permet de visualiser les informations sémantiques associé à l'élément sémantique courant, qui a été sélectionné soit dans l'arbre soit dans la scène 3D. Tous les objets composant le bâtiment sont contrôlés par l'arbre ou en employant dynamiquement la scène

L'IFC Viewer est certifié ISO/PAS certifié 16739 depuis mai 2003 par l'IAI [12]. Cette 3D est "sémantiquement" dynamique car la structure de la scène est calculée en temps réel depuis la base de données IFC selon les dispositifs géométriques (distance du point de vue, objets cachés, ...) et les dispositifs sémantiques (droites de l'utilisateur, type d'utilisateur, étape de la visualisation du projet, ...).

La figure 6 présente une capture d'écran de l'IFC Viewer. La plateforme entière est développée en utilisant les Web Services qui sont associés dynamiquement dans une application de collaboration raccordant plusieurs utilisateurs selon leurs droits [13]. Ce fichier a une taille 7Mo et fait partie des tests de performance.

Nous avons évalué le chargement de différents fichiers IFC. Les performances sont basées d'une part sur le chargement, la conversion et l'analyse du fichier et d'autre part sur le calcul des géométries lors de la visualisation. Le tableau 1 montre les résultats des essais :

Taille	4.4Mb	7Mb	15Mb	22Mb
Chargement	12''	18''	52''	1'9''
Calcul	11''	1'23''	3'11''	7'02''

tableau 1. Performances de chargement et de calculs des géométries

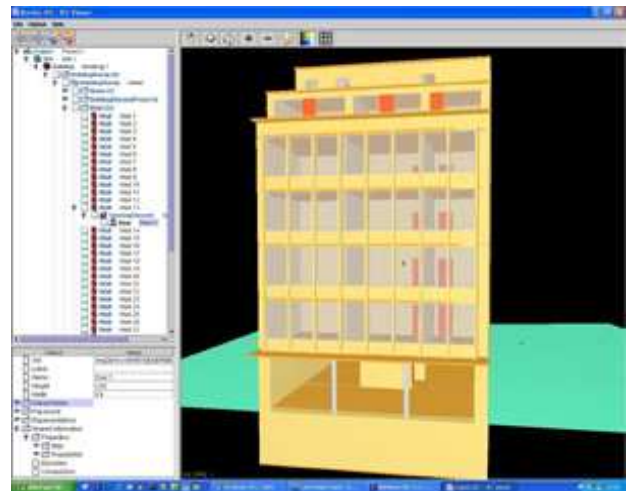


fig. 6. Capture d'écran de l'IFC Viewer

REMERCIEMENTS

Nous remercions l'ANVAR et l'ANRT pour leur soutien. Nous remercions aussi le président d'ACTIVE3D Olivier Gaudard, ainsi que tous les membres de l'équipe de développement, Florent Barth, Sylvain Bonandrini, Samuel Desgouilles, Sébastien Delune et Estelle Jacquemin pour leur temps et leur coopération.

REFERENCES

- [1] William V Baxter III, Avneesh Sud, Naga K Govindaraju and Dinesh Manocha, *GigaWalk : Interactive Walkthrough of Complex Environments*, Eurographics Workshop on Rendering, 2002.
- [2] Funkhouser T., Teller S., Sequin C. and Khorramabadi D., *The UC Berkeley System for Interactive Visualization of Large Architectural Models*, Presence, the Journal of Virtual Reality and Teleoperators, vol5, nb1, MIT Press(1996), pp.13-44
- [3] Wonka P., Wimmer M. and Sillion F., *Instant Visibility*, EG 2001 Proceedings, vol20(3), Blackwell Publishing, pp. 411-421
- [4] Balaguer, J.-F., and De Gennaro S. *VENUS: A virtual reality project at CERN*. Computer Graphics 30, 4 (Nov. 1996), 40–48.
- [5] Airey, J. M., Rohlf, J. H., and Brooks, JR., F. P. *Towards image realism with interactive update rates in complex virtual building environments*. Computer Graphics (1990 Symposium on Interactive 3D Graphics) 24, 2 (Mar. 1990), 41–50.
- [6] Patrice Torguet, Olivier Balet, Enrico Gobbetti, Jean-Pierre Jessel, Jérôme Duchon, and Eric Bouvier. *Cavalcade: A system for collaborative prototyping*. In Gérard Subsol, editor, Proc. International Scientific Workshop on Virtual Reality and Prototyping, pages 161-170, June 1999. Conference held in Laval, France
- [7] Varol Akman, Mehmet Surav, *Steps Toward Formalizing Context*, AI Magazine 17(3): 55-72, 1996
- [8] F. Recanati. *Direct Reference: From Language to Thought*. Blackwell Publishers, Oxford, UK, 1993
- [9] Extensible Markup Language (XML) : <http://www.w3.org/XML/>
- [10] Extensible Stylesheet Language (XSL) : <http://www.w3.org/Style/XSL/>
- [11] Active3D: <http://www.active3d.net>
- [12] International Alliance for Interoperability IAI: http://www.iai-international.org/iai_international/
- [13] Christophe Nicolle, Kokou Yétongnon, XML Based Toolkits for the Interoperability of Web Information Systems, Chapter book, Web-Enabled Systems Integration: Challenges X Practice, 2002