



HAL
open science

SIGA3D : modélisation, échange et visualisation d'objets 3D du bâtiment et d'objets urbains géoréférencés ; application aux IFC pour la gestion technique de patrimoine immobilier et urbain

Clément Mignard

► To cite this version:

Clément Mignard. SIGA3D : modélisation, échange et visualisation d'objets 3D du bâtiment et d'objets urbains géoréférencés ; application aux IFC pour la gestion technique de patrimoine immobilier et urbain. Autre [cs.OH]. Université de Bourgogne, 2012. Français. NNT : 2012DIJOS050 . tel-00842227

HAL Id: tel-00842227

<https://theses.hal.science/tel-00842227v1>

Submitted on 8 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mémoire de Thèse

Présenté pour obtenir
Le grade de Docteur ès Sciences
Mention Informatique
Par

Clément Mignard

SIGA3D : Modélisation, Échange et Visualisation d'Objets 3D du Bâtiment et d'Objets Urbains Géoréférencés; Application aux IFC pour la Gestion Technique de Patrimoine Immobilier et Urbain

Le 13 novembre 2012

Composition du jury :

Pr. Frank Boochs, Professeur des Universités, I3Mainz, Mayence, Allemagne, **Rapporteur**

Pr. Tahar Kechadi, Professeur à l'University Collège of Dublin, Irlande, **Rapporteur**

Pr. Christophe Nicolle, Professeur des Universités, LE2I, Dijon, France, **Directeur de thèse**

Pr. Gilles Gesquière, Professeur des Universités, LIRIS, Lyon, France, **Co-Encadrant**

Doctorat préparé au sein de l'équipe Systèmes d'Information et Systèmes d'Images, dans l'équipe projet CHECKSEM du laboratoire Le2i de l'Université de Bourgogne, et financé par la société Active3D, filiale du Groupe Archimen.



Remerciements

Les remerciements... généralement, lorsque l'on arrive à ce stade d'une thèse c'est plutôt bon signe. C'est donc avec un grand plaisir que je me prête à cet exercice. D'aucuns prétendent que, pour un doctorant, sa thèse est son « bébé », une obsession qui réveille la nuit et travaille la journée. Pour l'avoir maintenant vécu pendant plus de trois ans, je puis vous le confirmer et vous annonce la venue de mon mémoire de thèse que vous êtes en train de découvrir. Mais, comme tout un chacun le sait, on ne fait pas un bébé tout seul. *Si je dois résumer ma thèse aujourd'hui avec vous, je dirais que c'est d'abord des rencontres.* Chacune d'entre elles a su apporter sa contribution à la conception de cette thèse, à sa façon, sans forcément en être consciente, et je tiens ici à remercier toutes ces personnes.

En premier lieu, non seulement parce que c'est la coutume, mais aussi parce que c'est la personne qui m'a le plus apporté durant cette thèse, je tiens à remercier Christophe Nicolle, mon directeur de thèse. Il est à l'origine de notre collaboration au sein de son équipe de recherche et je le remercie pour son aide précieuse, sa disponibilité, sa pédagogie et son talent d'innovation dans le milieu de la recherche scientifique, son extrême patience, et son soutien indéfectible, même durant ces moments de doutes qui parsèment une thèse.

Ensuite, je tiens à remercier très chaleureusement Gilles Gesquière qui a co-encadré cette thèse. Sa disponibilité et ses conseils toujours très avisés ont été d'une très grande aide dans l'élaboration de cette thèse. Il a notamment apporté de l'énergie, de la passion et un regard neuf sur mon projet de recherche.

Je tiens également à remercier l'entreprise GROUPE ARCHIMEN (incluant Active3D) qui m'a accueilli et son président, M. Olivier Gaudard, pour avoir financé mes recherches au sein de la société. Il m'a donné les moyens de réaliser mes recherches et de pouvoir les appliquer au développement de l'application informatique Active3D.

J'adresse également mes remerciements sincères aux membres de mon jury de thèse : Pr. Tahar Kechadi (Professeur à l'University Collège of Dublin, Rapporteur) et Pr. Franck Boochs (Professeur des Universités, I3Mainz, Rapporteur), pour leur rapport et avoir accepté de me faire soutenir, ainsi que les Pr. Abder Koukam (Professeur des Universités, UTBM, Belfort, Examineur) et Pr. Marc Neveu (Professeur des Universités, LE2I, Examineur) pour avoir accepté d'être examinateurs.

Bien sûr, atteindre ces objectifs n'aurait pas été possible sans l'aide des membres d'Active3D qui ont participé au projet. Je souhaite notamment remercier Julien Reverdiau, chef de projet informatique pour son soutien, ses conseils et sa compréhension quant aux difficultés à avoir un chercheur dans une équipe de développeurs. Je remercie également Olivier Clercq et Marc Rastoix, ingénieurs en développement, pour leur coopération, l'aide et l'écoute qu'ils ont portées à mon projet, ainsi que pour leur disponibilité et leur implication sur le projet européen SIGA3D.

Je désire en outre remercier tous les membres de Groupe Archimen et Active3D pour leur sympathie, leur amitié, les nombreux pots autour de la fameuse banque du deuxième et ces soirées à faire du sport, ce qui contribue fortement au succès de cette thèse. J'ai eu beaucoup de plaisir à travailler avec eux. Le cadre de travail était idéal.

Je remercie également tous les membres du projet CHECKSEM, pour leur aide et leur bonne humeur. Nous avons partagé de bons moments, tant studieux que ludiques.

J'adresse un grand merci à la société lat/ion en Allemagne pour notre collaboration au sein du projet européen SIGA3D, et en particulier à Jens Fitzke, Andreas Poth et Andreas Schmitz avec qui nous avons beaucoup échangé. Je salue leur motivation et leur aide précieuse sur certains aspects techniques du projet, ainsi que leur accueil toujours chaleureux à plusieurs reprises en Allemagne.

Pour finir, je tiens à remercier mes proches. Tout d'abord Éloïse pour son soutien et sa patience durant ces trois dernières années. Ce n'est pas forcément évident de supporter tous les jours un doctorant à la maison. Enfin, je ne remercierai jamais assez mes parents pour leur soutien de tous les jours. J'espère qu'ils sont fiers de moi et de ce travail. Merci beaucoup également à mes deux petites sœurs, Lucie et Élise, pour leurs encouragements et leur joie de vivre.

Résumé

Cette thèse définit une nouvelle approche de gestion technique de biens immobiliers et urbains. Pour cela, un processus de production et de gestion de l'information du bâtiment, de son environnement proche et des objets urbains est défini. Il permet de gérer ces objets tout au long de leur cycle de vie, au sein d'un concept appelé Urban Facility Management, ou gestion technique de patrimoine urbain. Cette technologie s'appuie sur un Modèle d'Information Urbain qui permet de modéliser dans une ontologie dynamique et évolutive toute l'information de la ville et du bâtiment. Un mécanisme de niveaux de détail géométrico-contextuels a par ailleurs été défini afin de supporter la montée en charge d'un tel système. En effet, le nombre d'objets à gérer dans une scène par rapport à un système de modélisation du bâtiment est bien supérieur, tout comme la superficie de l'information à traiter. Aussi, les niveaux de détail contextuels permettent d'optimiser la scène en fonction de critères sémantiques.

La proposition que nous avons faite se base sur une architecture dérivée des travaux en systèmes hypermédia adaptatifs. Elle est composée de six couches permettant de résoudre les problématiques de manière systémique : une couche de connexion aux sources de données, une couche d'import de ces données dans le système, une couche sémantique, une couche de contextualisation, une couche de connexion et une couche d'interface avec l'utilisateur. Cette architecture autorise la définition d'un workflow que nous avons décomposé au sein d'une architecture de processus. Celui-ci décrit la manière d'acquérir des données à la fois de sources SIG mais aussi IFC ou plus généralement CAO. Ces données sont importées dans le modèle d'information de la ville au sein d'une ontologie définie à l'aide d'outils présents dans la couche sémantique. Ces outils consistent en un ensemble d'opérateurs permettant de définir des concepts, relations et énoncés logiques. Ils sont couplés à un mécanisme de contextes locaux qui permet de définir des niveaux de détail contextuels au sein de l'ontologie. Associé à des représentations graphiques provenant des sources données, il est ainsi possible d'associer plusieurs représentations géométriques, 2D ou 3D, à un objet de notre base de connaissance, et de choisir le modèle à appliquer en fonction de critères sémantiques. La couche de gestion du contexte apporte des informations contextuelles au niveau des modèles de données grâce à un mécanisme basé sur les named graphs. Une fois l'ontologie mise en place, les données peuvent être exploitées par l'intermédiaire d'un moteur graphique que nous avons développé et qui gère l'information contextuelle et les niveaux de détail.

Cette architecture de processus a été implémentée et adaptée sur la plateforme Active3D. Une partie des travaux de recherche a consisté à adapter l'architecture formelle du modèle d'information urbain à l'architecture existante afin de répondre aux contraintes industrielles du projet et aux critères de performances requis. La dernière partie de cette thèse présente les développements techniques nécessaires à la réalisation de ces objectifs.

Abstract

This thesis defines a new approach in urban and building facility management. A production and information management process of the building, its immediate environment and urban objects has been defined. It allows to manage these objects throughout their life cycle, in a concept called Urban Facility Management. This technology relies on an Urban Information Model to model all the information of cities and buildings in a dynamic and evolutive ontology. A mechanism for geometric and semantic levels of detail has also been defined to support the scalability of such a system. Indeed, the number of objects to be managed in a scene, compared to a building modeling system, is much higher, as the area of information processing. Also, contextual levels of detail optimize the scene according to semantic criteria.

The proposal we made is based on an architecture derived from work in adaptive hypermedia systems. It consists of six layers that aim at solving problems in a systemic form: a connecting layer to data sources, a layer to import these data into the system, a semantic layer, a level of contextualization, a connection layer and an interface layer with the user. This architecture allows the definition of a workflow that we decomposed in an architecture of processes. This one describes how to acquire data from both GIS sources and IFC, but also from CAD in general. Data are then imported into the city information model into an ontology defined using tools from the semantic layer. These tools are composed of a set of operators that aim at defining concepts, relations and logical statements. They are coupled to a mechanism that allows the definition of local contexts used to express contextual levels of details within the ontology. Associated with graphical representations issued from diverse data sources, it become possible to combine several geometric representations, 2D or 3D, to an object of the knowledge base, and choose the representation to be displayed according to semantic criteria. The layer for context management provides contextual information on global data models through a mechanism based on named graphs. Once the ontology is developed, data can be exploited via a graphical engine that has been specially developed and that manages contextual information and levels of details.

This architecture of processes has been implemented and adapted to the platform Active3D. Part of the research has been to adapt the formal architecture of the urban information model to the existing architecture. The goal was to provide a solution to the industrial constraints of the project and required performance criteria. The last part of this thesis therefore presents the technical developments necessary to achieve these goals.

Liste des abréviations, sigles et acronymes

ADE Application Domain Extensions	KML Keyhole Markup Language
AEC Architecture, Engineering & Construction	LD Logique de Description
BIM Building Information Model	LoD Level of Detail
CAO Conception assistée par ordinateur	MADS Metadata Authority Description Schema
C-DMF Contextual Data Model Framework	MNT Modèle Numérique de Terrain
CMF Context Model Framework,	NTF Nouvelle Triangulation de la France
COLLADA COLLABorative Design Activity	NWIP ISO/NP New Work Item Proposa <i>Voir</i> ISO/NP
CRS Coordinates Reference System	OGC Open Geospatial Consortium
CS-DMF Contextualisation and Synchronisation of Data Model Framework <i>Voir</i> C-DMF	OS OwnerShip
CSTB Centre Scientifique et Technique du Bâtiment	OSI Open Systems Interconnection
DAO Dessin Assisté par Ordinateur	OWL Web Ontology Language
DGN DesiGN	Pset Property Sets
DILAS Digital LANDscape Server	RCC Region Connection Calculus
DMF Data Model Framework <i>Voir</i> CDMF	RDF Resource Description Framework
DWG DraWinG	RGF93 Réseau Géodésique Français 1993
DXF Drawing eXchange Format	RuleML Rule Markup Language
EPSG European Petroleum Survey Group	SCU Système de Coordonnées Utilisateur <i>Voir</i> UCS
FM Facility Management	SIG Systèmes d'Information Géographique
FU Functional Unit	SPARQL Protocol and RDF Query Language
GIS <i>Voir</i> SIG	STEP STandard for the Exchange of Product model data
GML Geography Markup Language	SWRL Semantic Web Rule Language
GMLJP2 GML in JPEG 2000 for Geographic Imagery	TCP-IP Transmission Control Protocol/Internet Protocol
GUID Global Unique Identifier	TIN Triangular Irregular Network
H-LoD Hierarchical Level of Detail	TPP Tangential Proper Part of
IAI International Alliance for Interoperability	UBM Unified Building Model
IFC Industry Foundation Classes	UCS User Coordinates System
ifcXML <i>Voir</i> IFC, XML	UDM Urban Data Model
IFG IFC for GIS	UIM Urban Information Model
IGN Institut national de l'Information Géographique et forestière	UML Unified Modeling Language
ISO International Organization for Standardization	URI Uniform Resource Identifier
ISO/DTS Draft Technical Specification	VRML Virtual Reality Markup Language
ISO/IEC International Electrotechnical Commission <i>Voir</i> ISO	W3C World Wide Web Consortium
ISO/IS International Standard <i>Voir</i> ISO	WCS World Coordinates System
ISO/NP New work item Proposal <i>Voir</i> ISO	WFS Web Feature Service
ISO/PAS Publicly Available Specification <i>Voir</i> ISO	WMS Web Map Service
ISO/TC Technical committees <i>Voir</i> ISO	X3D Extensible 3D
ISO/TS Technical Specification	XML eXtensible Markup Language
	XSL eXtensible Stylesheet Language

Table des matières

Liste des abréviations, sigles et acronymes.....	9
Chapitre 1. Introduction	
1 Interopérabilité et Génie civil.....	23
2 Active3D et le « Facility Management ».....	24
3 Le projet SIGA3D	25
4 Organisation du document.....	28
Chapitre 2. Contexte	
1 Le modèle des IFC.....	33
1.1 Pourquoi les IFC.....	34
1.2 Présentation de la norme IFC	35
1.3 Exemple	38
1.4 Les IFC en pratique	40
2 Les ontologies.....	41
3 Active3D 2001-2004	42
4 Active3D 2004-2007	43
4.1 Modélisation de la connaissance.....	44
4.2 C-DMF	46
4.2.1 Définition de la brique « Data Model Framework ».....	48
4.2.2 Définition de la brique « Context »	48
4.2.3 Les opérateurs de graphe.....	49
4.2.4 Élément de structure des contextes	50
4.2.5 Exemple appliqué à la gestion de patrimoine immobilier.....	51
5 Active3D 2011-2012 : le projet SIGA3D.....	56
Chapitre 3. Etat de l'art	
1 Introduction aux Systèmes d'Information Géographique.....	63
1.1 Organisation des données	64
1.2 Gestion des coordonnées	66
1.3 Historique et origine des diversités	67
1.3.1 Les systèmes de référence géodésiques	67
1.3.2 Les systèmes de coordonnées.....	69
1.3.3 Le repère	70
1.3.4 Définition du datum	71
2 Modélisation Urbaine.....	72
2.1 IFG.....	74
2.2 L'intégration des IFC dans CityGML.....	75
2.3 Un framework de conversion 3D conçu par l'université technique de Berlin	75
2.4 Framework pour la conversion automatique de BIM (IFC) en CityGML.....	77
2.5 Une ADE CityGML	78
2.6 The Unified Building Model	79
2.7 Limites et conclusion	80
3 La 3D dans les SIG.....	81
3.1 Premiers pas vers la troisième dimension	81
3.2 La topologie en 3D.....	83
3.2.1 La topologie dans les IFC et CityGML	83
3.2.2 Gestion d'un ensemble de données important	84

3.2.3	Niveaux de détails (Levels of Details).....	85
3.2.4	Le découpage (culling).....	87
4	Normes.....	87
4.1	Présentation des normes géographiques.....	89
4.1.1	Les normes sur les ontologies.....	90
4.1.2	ISO 19110 - Méthodologie de catalogage des entités.....	91
4.1.3	ISO 19150-1 - Ontologie – Partie 1 : Structure.....	92
4.1.4	ISO19107 - Schéma spatial.....	93
4.1.5	ISO 19108 - Schéma temporel.....	94
4.1.6	ISO19136 - GML.....	94
4.2	Conclusion.....	97
5	La multireprésentation dans les SIG.....	98
5.1	La multireprésentation en cartographie.....	98
5.2	La multireprésentation dans les ontologies.....	100
5.2.1	Ontologie géographique / évolutive.....	100
5.2.2	MADS et ses applications aux ontologies géographiques.....	101
5.2.3	La multireprésentation en 3D.....	103
6	Conclusion.....	104

Chapitre 4. Extension d'Active3D à la modélisation urbaine

1	Modélisation des données et Géométrie.....	112
2	Modélisation des données et multireprésentation sémantique.....	114
2.1	Pre-requis.....	115
2.2	Les classes et propriétés.....	116
2.2.1	Déclaration contextuelle de classes.....	116
2.2.2	Déclaration contextuelle de propriétés.....	117
2.2.3	Les prédicats.....	118
2.2.4	Les opérateurs.....	120
3	La couche de gestion du contexte.....	129
3.1.1	Le systemGraph.....	129
3.1.2	La validité du et dans le modèle.....	132
4	Conclusion.....	135

Chapitre 5. Architecture SIGA3D

1	Niveau 1 – Sources de données.....	139
2	Niveau 2 – Import / Export SIGA3D.....	140
3	Niveau 3 – Data Model Framework de SIGA3D.....	142
4	La gestion spatiale des objets urbains.....	143
5	Niveau 4 – La couche de gestion du contexte.....	146
5.1	Le contexte.....	146
5.2	La gestion des référentiels et la synchronisation spatio-temporelle.....	148
6	Niveau 5 – La couche de connexion.....	149
7	Conclusion.....	149

Chapitre 6. Développement

1	Les processus autour de l'Urban Facility Management.....	153
1.1	Processus de modélisation.....	155
1.2	Processus d'acquisition des données.....	155
1.3	Processus d'import.....	156

1.4	Processus de géoréférencement	158
1.5	Stockage des données	159
1.6	Les processus de streaming et de visualisation	160
2	Le projet européen SIGA3D	162
2.1	Préparation technique d'Active3D	163
2.2	Gestion des coordonnées	164
2.3	Modélisation du terrain et gestion des objets urbains.....	165
3	Implémentation SIGA3D.....	168
3.1	Le développement d'un nouveau moteur graphique.....	169
3.2	Stockage du graphe de scène	173
3.3	Définition de l'ontologie	179
4	Conclusion	180
Chapitre 7. Conclusion et Travaux Futurs		183
Bibliographie		189
Rapports Techniques.....		195
Publications		197

Liste des figures

Figure 1. Classification des objectifs du projet SIGA3D.....	28
Figure 2. Schéma de l'architecture IFC2x4 RC2 (Source: IFC2x4 RC2 Final Online Documentation)	36
Figure 3. Structure des classes IFC (Source : (Khemlani, 2010))	37
Figure 4. Un étage et un mur sont reliés par un élément de relation IfcRelContains.....	40
Figure 5. Exemple de liens directs entre les éléments sémantiques	40
Figure 6. Famille de formalisme par domaine d'étude	44
Figure 7. Pile des langages du web	45
Figure 8. Placement de C-DMF dans la pile du Web sémantique	47
Figure 9. Architecture de C-DMF.....	47
Figure 10. Hiérarchie des graphes C-DMF.....	49
Figure 11. Vue graphique de l'union de graphes	49
Figure 12. Exemple de SystemGraph	51
Figure 13. Exemple de SpaceSystem pour Active3D	52
Figure 14. Exemple de bâtiment	53
Figure 15. Application de C-DMF à un système de suivi des modifications d'un bâtiment	54
Figure 16. Définition de deux modèles de données avec C-DMF	55
Figure 17. Une scène 3D correspondant au contexte de plomberie	56
Figure 18. Les types de représentations graphiques des données géographiques	64
Figure 19. Carte d'un quartier de ville en représentation vectorielle.....	65
Figure 20. Description des objets géographiques au niveau sémantique et géographique.....	65
Figure 21. La topologie pour une meilleure cohérence de données	66
Figure 22. Approximation du géoïde par deux ellipsoïdes (global et local)	68
Figure 23. Système de coordonnées sphérique	70
Figure 24. Écarts à la transformation standard sur les points en colocalisation NTF/RGF93 (IGN)	71
Figure 25. Approche de la modélisation urbaine par les SIG.	72
Figure 26. Processus de reconstruction en deux étapes (Nagel et al., 09)	76
Figure 27. Transformation de modèles IFC en CityGML (Kolbe et al., 2009)	78
Figure 28. Liste de classes IFC utiles dans un contexte géographique (extrait de ...)	79
Figure 29. Les étapes pour développer un UBM.....	80
Figure 30. Exemple de MNT habillé avec une orthophoto pour plus de réalisme (IGN).	82
Figure 31. La topologie dans les IFC	84
Figure 32. Exemple de niveaux de détail discrets sur une sphère	85
Figure 33. Fonctionnement des niveaux de détails hiérarchiques.....	86
Figure 34. Les LoD dans CityGML	87
Figure 35. Les normes et spécifications techniques de l'ISO/TC211 (CNIG, 2006)	90
Figure 36. Les schémas conceptuels de l'ISO 19107	95
Figure 37. Structure de l'élément geometricCollection en GML	96
Figure 38. GML et les normes	97
Figure 39. Exemple de généralisation cartographique (données issues du Geoportail).....	99
Figure 40. Relations topologiques.....	103
Figure 41. Couplage GIS et BIM avec SIGA3D	104
Figure 42. Convergence des méthodes BIM et SIG pour l'ouverture de la modélisation sémantique	109
Figure 43. Positionnement des niveaux de détail contextuels de SIGA3D	110
Figure 44. Exemple de relation spatiale (inclusion tangente) entre deux modèles de données	111

Figure 45. Détail des couches de données et de contexte de SIGA3D	112
Figure 46. Exemple de graphe acyclique orienté	113
Figure 47. Exemple d'un graphe de scène avec des groupes de nœuds et niveaux de détail	113
Figure 48. SIGA3D : détail de la couche de modélisation des données	114
Figure 49. SIGA3D : détail de la couche de modélisation du contexte	130
Figure 50. Exemple de définition d'un modèle de données (city district 1)	134
Figure 51. Architecture de processus SIGA3D.....	139
Figure 52. Exemple de pièces à importer dans l'ontologie SIGA3D	141
Figure 53. Illustration d'octree sur deux niveaux.....	144
Figure 54. Exemple de correspondance entre la vue figée et le graphe de scène.....	145
Figure 55. Exemple de visualisation de patrimoine urbain dans SIGA3D	150
Figure 56. Urban Facility Management, processus métier et gestion	153
Figure 57. Architecture des processus de l'application Active3D avant le projet SIGA3D.....	154
Figure 58. Architecture de processus.....	154
Figure 59. Outil de géoréférencement.....	156
Figure 60. Processus de géoréférencement.....	157
Figure 61. Processus de géoréférencement : positionnement des points de référence	157
Figure 62. Snapshot of the SIGA3D platform	162
Figure 63. Rendu dans la même scène de deux sites dont les systèmes de coordonnées sont locaux.....	165
Figure 64. Illustration du problème de positionnement vertical des objets par rapport au terrain.....	166
Figure 65. Exemple de repositionnement vertical d'un bâtiment	166
Figure 66. Illustration d'un problème dû au manque de topologie	167
Figure 67. Architecture de collaboration SIGA3D	168
Figure 68. Exemple de bâtiment simple : affichage 3D et graphe de scène.	169
Figure 69. Fonctionnement du lien entre le moteur graphique et la visualisation des informations	170
Figure 70. Comparatif des moteurs de rendu JME et Ogre4J.	172
Figure 71. Architecture logicielle 3-tier Active3D	173
Figure 72. Extrait du modèle conceptuel de la base de données relationnelle stockant le graphe de scène	175
Figure 73. Définition du graphe de scène	175
Figure 74. Représentations	175
Figure 75. Comparatif d'une représentation explicite par Trimesh et d'une représentation par primitive	176
Figure 76. Exemple de fichier IFC et son graphe de scène.....	177
Figure 77. Lien entre le modèle graphique et le modèle sémantique	178
Figure 78. Le géoréférencement.....	178
Figure 79. Mise en interface MemoryModel/EngineCore	179
Figure 80. Extrait du schéma conceptuel de base de données de la GTP.....	180
Figure 81. Correspondance entre l'architecture SIGA3D et les éléments de l'interface utilisateur	181

Liste des Tableaux

Tableau 1. Synthèse des approches par catégories	105
Tableau 2. Liste des opérateurs contextuels de logique de description.....	116
Tableau 3. Performance du calcul d’empreinte 2D en fonction de la résolution de l’image.....	159
Tableau 4. Performances du moteur par rapport au poids des fichiers et taille des géométries.....	176

Liste des Scripts

Script 1 Exemple de fichier IFC.....	38
Script 2. Exemple de syntaxe RDF/XML	45
Script 3. Exemple de Syntaxe OWL	46
Script 4. Exemple de Syntaxe SWRL	46
Script 5. Exemple de Définition de Hiérarchie OWL.....	46
Script 6. Exemple de création de modèles avec DMF	48
Script 7. Exemple d'une mise à jour de modèle de données.	55
Script 8. Exemple de définition de géométries avec GML.	97
Script 9. Exemple de définition d'un concept estampillé, d'après (Benslimane et al., 2003)	102
Script 10. Extension de la grammaire des Logiques de Description pour la multireprésentation sémantique ..	102
Script 11. Exemple de relations topologiques.....	103
Script 12. Définition des espaces de noms.....	115
Script 13. Définition de dmf:Class contextuel	116
Script 14. Définition de dmf:localContext	117
Script 15. Exemple de description contextuelle d'une vue d'un Bâtiment	117
Script 16. Exemple de description contextuelle d'une vue d'un Bâtiment en logique de description	117
Script 17. Définition de dmf:Property	117
Script 18. Exemple de définition d'une propriété contextuelle "contenir"	118
Script 19. Définition de prédicats dmf	118
Script 20. Définition des propriétés des prédicats	118
Script 21. Exemple de description d'un étage avec des prédicats contextuels	119
Script 22. Correspondance des prédicats unaires contextuels avec la logique de description	119
Script 23. Exemple de définition de la multi représentation des cardinalités avec les prédicats	119
Script 24. Équivalence des prédicats binaires contextuels en logique de description.....	120
Script 25. Définition de dmf:Imp contextuel.....	120
Script 26. Exemple de l'utilisation de $\text{dmf} : [c_1, \dots, c_n] \text{Imp}$	121
Script 27. Définition du contexte local appliqué à l'opérateur $\text{dmf} : [c_1, \dots, c_n] \text{Imp}$	123
Script 28. Exemple d'utilisation contextuelle de l'opérateur dmf:And	124
Script 29. L'opérateur contextuel $\text{dmf}[c_1, \dots, c_n] \text{And}$	124
Script 30. Exemple d'utilisation de $\text{dmf} : [c_1, \dots, c_n] \text{Or}$	125
Script 31. L'opérateur contextuel $\text{dmf} : [c_1, \dots, c_n] \text{Or}$	125
Script 32. Exemple d'utilisation de dmf:OrX	126
Script 33. L'opérateur contextuel $\text{dmf}:[s_1, \dots, s_m] \text{OrX}$	126
Script 34. Exemple d'utilisation de dmf:Not	126
Script 35. L'opérateur contextuel $\text{dmf}:[s_1, \dots, s_m] \text{Not}$	127
Script 36. Exemple d'utilisation de dmf:Diff	127
Script 37. L'opérateur contextuel $\text{dmf}:[s_1, \dots, s_m] \text{Diff}$	127
Script 38. L'opérateur contextuel $\text{dmf}:[s_1, \dots, s_m] \text{spatialEntity}$	128
Script 39. Définition du SystemGraph de la couche de gestion des contextes de la brique CS-DMF	130
Script 40. Définition du graphe spécial SyncModelGraph utilisé dans CS-DMF.....	132
Script 41. Exemple de fichier de mappage IFC.....	141
Script 42.Exemple d'attributs utilisés pour le mappage de l'information	141
Script 43. Exemple de Named Graph	147

Chapitre 1

Introduction

Résumé

Mes travaux de recherche s'inscrivent dans la continuité du projet Active3D initié en 2001. L'objectif est de construire une ontologie pour la gestion technique de patrimoine urbain. Pour cela, il est nécessaire de définir ce qu'est un patrimoine urbain et de comprendre qu'hors du bâtiment, la modélisation 3D et la modélisation sémantique nécessitent la prise en compte du géoréférencement des objets urbains ainsi que la gestion des niveaux de détails au sein d'un moteur 3D. À ces contraintes s'ajoutent celles de la philosophie du projet Active3D. Ce chapitre introduit SIGA3D et les objectifs de recherche associés.

Plan

1	Interopérabilité et Génie civil	23
2	Active3D et le « Facility Management »	24
3	Le projet SIGA3D	25
4	Organisation du document	28

A l'origine, l'informatique avait pour ambition de décrypter des messages pour percer des secrets industriels et militaires de pays voisins. Depuis cette époque, dans les années 1930 à la fin du XXe siècle, l'informatique s'est développée exponentiellement, toujours dans le but de décrypter des messages, mais aussi pour partager la connaissance. D'une certaine manière, l'informatique a pour objectif fondamental de faire comprendre à l'homme son environnement. C'est dans cette optique que de nombreux travaux permettant l'interopérabilité de systèmes d'information hétérogènes ont été développés. Cette interopérabilité peut prendre plusieurs aspects, mais elle consiste à associer des ressources (humaines ou matérielles) à la réalisation d'un objectif commun. Le développement des réseaux informatiques a largement contribué à la réussite de l'interopérabilité. Néanmoins, de nombreux verrous ont dû être levés pour résoudre les problèmes d'hétérogénéités. Avec un peu de recul, on peut identifier un fonctionnement général dans le processus de résolution des différents types d'hétérogénéité. Tout d'abord il y a eu le développement de systèmes indépendants et utilisant un format propriétaire. Puis sont arrivés les outils et méthodes pour interfacier ces systèmes. Par la suite, une phase d'évolution de ces outils et méthodes vers des concepts de plus en plus abstraits (du traducteur de schéma au méta modèle) a eu lieu. Enfin, a suivi la sélection d'une norme commune et ouverte. Ce fonctionnement a été mis en place pour l'adoption in fine de standards pour l'échange d'information sur des réseaux (norme OSI, protocole TCP-IP), sur le stockage des données (formalisation relationnelle), sur la représentation semi-structurée de l'information (les langages basés sur la norme XML) ...

Bien que ce fonctionnement soit un succès pour résoudre les problèmes d'hétérogénéité de protocole, de structure, de schématique et de stockage, il ne permet pas de résoudre le problème de l'hétérogénéité sémantique. Cette forme d'hétérogénéité intervient lorsque deux éléments qui sont identiques dans leur structure ont des sens différents dans un contexte donné (les homonymies en sont un exemple plein de sens, ou le mot « sens » ici, dans ce contexte, a une sémantique différente que dans un contexte cartographique (la ville de Sens, ou encore le sens d'une route)). De nombreux travaux dans ce domaine ont été publiés. Des normes telles qu'OWL¹ et RDF² proposent de représenter la sémantique des systèmes. Néanmoins, la construction d'une modélisation sémantique entre des systèmes hétérogènes reste un travail de recherche long et difficile.

1 Interopérabilité et Génie civil

Dans le domaine du génie civil, et plus particulièrement tout au long du cycle de vie d'un patrimoine immobilier, de nombreux acteurs interviennent. Au cours de ce projet, ces acteurs mettent en place des processus opérationnels qui leur sont propres, sans consultation et interaction avec les autres équipes. De sa conception à sa construction, un bâtiment nécessite la coordination, la compréhension et l'enchaînement de nombreux systèmes hétérogènes. Jusqu'à la fin du XXe siècle, le fonctionnement général adopté pour réduire ces problèmes d'hétérogénéité a été l'usage du plan sur papier, généralement dessiné à partir d'un logiciel de DAO (Dessin assisté par ordinateur) en vectoriel. L'équipement informatique des acteurs était très faible, et sur les chantiers, les lignes téléphoniques quasi inexistantes, ce qui rendait difficile les échanges entre les différents acteurs. Depuis vingt ans, les processus et les outils d'aide au secteur de la construction sont en pleine évolution. Les causes de celle-ci sont multiples, mais trouvent principalement leur source dans l'évolution des normes et des technologies de l'information pour répondre aux pressions liées aux contrôles des coûts, des retards et du respect des normes (écoconstruction, hygiène et sécurité...). Dans la même logique que pour l'évolution de l'informatique, le monde du bâtiment a dû s'ouvrir et proposer une norme ouverte pour modéliser le bâtiment de manière uniforme. Pour atteindre cet objectif, une association a été créée il y a une quinzaine d'années, initialement appelée International Alliance for Interoperability (IAI), puis aujourd'hui

¹ Ontology Web Language, <http://www.w3.org/TR/owl2-overview/>

² <http://www.w3.org/RDF/>

BuildingSmart³. Après plusieurs années de concertation et d'évolution, cet organisme a développé une nouvelle norme de représentation des connaissances du bâtiment : les IFC (Industry Foundation Classes) (IFC, 2005). L'objectif des IFC est de fournir une vue structurée et partagée des objets qui composent le bâtiment. Cette norme pousse les éditeurs de logiciels de dessin industriel à intégrer, en plus d'une représentation vectorielle d'un bâtiment, une représentation objet associant une description géométrique et une description sémantique des objets du bâtiment. La norme IFC actuelle est numérotée IFC 2X3, ce que l'on peut interpréter en simplifiant par deuxième version, 3e évolution. Malheureusement, la structure qui combine les différents objets du bâtiment ne permet pas d'identifier, dans le cycle de vie du bâtiment, ceux qui peuvent être créés, modifiés ou supprimés par tel ou tel acteur. Parfois, un même objet possède une définition qui est partagée entre différents corps de métiers (par exemple le radiateur par le thermicien et le plombier). Il est donc nécessaire d'ajouter à cette norme des vues adaptées à chaque corps de métiers, en organisant un processus de construction qui respecte la norme et le travail de chacun. Une autre limitation des IFC est que pour l'instant, la norme se limite à décrire un bâtiment et non l'intégralité d'un patrimoine immobilier (les espaces verts, les réseaux hors de ceux contenus par le bâtiment...).

2 Active3D et le « Facility Management »

Le « Facility Management », ou gestion technique de patrimoine immobilier, est une approche intégrée pour faire fonctionner, mettre à jour, améliorer et adapter les bâtiments et l'infrastructure physique d'un organisme. Dans la pratique, le Facility Management peut couvrir un éventail de services comprenant la gestion immobilière, la gestion financière du bâtiment, la gestion des ressources humaines liées au bâtiment, la gestion de l'état de santé du bâtiment et des risques associés, la gestion de l'entretien du bâtiment (services domestiques tels que nettoyage, garanti...) et la gestion des consommables (Barrett et al., 2003).

Aujourd'hui, et notamment grâce à la norme IFC, le BIM (Building Information Model) est devenu de plus en plus présent dans les diverses applications du bâtiment (Schevers et al., 2007) (Shen et al., 2010). Il reste néanmoins des efforts à faire pour les industriels au niveau de l'implémentation de la norme afin d'obtenir une interopérabilité parfaite entre les différents outils existants (Jeong et al., 2009).

Depuis 10 ans, l'équipe de recherche CHECKSEM dirigée par Christophe Nicolle travaille sur le projet Active3D⁴. Porté initialement par le Groupe Archimen (Bureau d'étude Bourguignon), puis par la société Active3D-Lab (éditeur de logiciels), ce projet a pour objectif de développer une plateforme de gestion technique de patrimoine. La plateforme Active3D propose une architecture où chaque objet du bâtiment est identifié, associé à une modélisation 3D et manipulé selon des vues contextuelles tout au long de son cycle de vie (Vanlande, 2007). Néanmoins cette modélisation objet s'appuie uniquement sur la norme IFC et s'arrête au périmètre du bâtiment, sans prendre en compte son environnement ni même des ensembles structurés de patrimoines bâtis. Les travaux de recherche que nous menons visent à combler ce manque en ouvrant l'architecture de modélisation sémantique du bâtiment à une nouvelle dimension. Celle-ci permet la représentation et la visualisation de la connaissance acquise sur les objets urbains et extra-urbains. Les objets urbains sont des éléments extérieurs au bâtiment que l'on trouve dans les villes tels que les réseaux physiques (route, réseau gaz, téléphone...) et leurs composants (valve, tuyaux, coude...). Les éléments extra-urbains étant définis de la même façon, mais en dehors des villes, nous parlerons dans la suite, et par abus de langage, uniquement de la représentation d'objets urbains. Ainsi, notre objectif est de construire un modèle d'information urbain pour permettre la gestion technique de patrimoine de ces objets.

La première phase de travail concernant le BIM a été menée au sein d'Active3D de 2001 à 2004 dans le cadre des travaux de thèse de Christophe Cruz (Cruz, 2004a). La proposition réalisée à l'issue de ces travaux offrait

³ <http://buildingsmart.com>

⁴ <http://www.active3d.net>

une gestion coordonnée du cycle de vie du bâtiment associée à une modélisation 3D des bâtiments basée sur la norme IFC. Une seconde phase de travail menée entre 2004 et 2007 par Renaud Vanlande (Vanlande 2007) a permis la définition d'une architecture appelée C-DMF (Contextual Data Model Framework). Cette formalisation permet la modélisation des bâtiments dans le respect de la norme IFC et le modèle de coopération Active3D, d'intégrer les données hétérogènes dans une ontologie générique et de modéliser des vues contextuelles (qui peuvent présenter l'information différemment selon l'utilisateur). Ces deux phases de travail ont permis de développer un outil performant de gestion du patrimoine qui est utilisé par des milliers d'utilisateurs⁵.

Outre de nombreuses fonctionnalités offertes à tous les intervenants, professionnels ou usagers d'un bâtiment, cette plateforme permet d'interconnecter la grande majorité des logiciels développés dans le domaine du bâtiment. De plus, elle permet de visualiser et de « manipuler » le bâtiment selon le niveau de droit de l'utilisateur au travers d'une interface associant modélisation 3D et description sémantique. Dans cet environnement, une modification réalisée sur un des objets du bâtiment est automatiquement répercutée et rendue visible dans les autres vues.

3 Le projet SIGA3D

Le projet Active3D est considéré comme une réussite, à la fois du point de vue de l'innovation, mais aussi d'un point de vue industriel. Néanmoins, il reste limité au périmètre du bâtiment, sans prendre en compte le patrimoine immobilier dans son ensemble. Pour répondre à ce problème, la société Active3D a souhaité entamer une troisième collaboration de recherche avec le laboratoire Le2i pour étendre la description sémantique des IFC à des objets qui se situent hors du bâtiment. Ce nouveau projet constitue le projet SIGA3D, porté à la fois par la société Active3D mais également par un projet européen Eureka.

Cette extension doit se faire dans le respect des normes que nous utilisons telles les IFC pour le bâtiment. Pour la modélisation extérieure du bâtiment, d'autres standards issus de l'OGC (Open Geospatial Consortium⁶) ou normes de l'ISO TC 211 sont à prendre en compte. En effet, les traitements effectués dans les Systèmes d'Information Géographique (SIG) impliquent aussi l'utilisation d'objets géométriques dans l'espace (objets 3D). Nous pouvons citer par exemple la norme internationale ISO 19107 (Geographic information -- Spatial schema) qui décrit les objets complexes à prendre en compte et les traitements qu'il est possible de leur appliquer. La naissance récente du standard CityGML⁷ est aussi à prendre en compte afin de faire évoluer le produit dans le sens d'un rapprochement du monde des SIG et du bâtiment.

L'objectif de cette thèse est de venir compléter et adapter le modèle qui existe dans la plateforme Active3D pour traiter des éléments externes aux bâtiments. Le but est de pouvoir centraliser les outils de gestion de patrimoine immobilier et urbain au sein d'une même structure sémantiquement cohérente et interopérable par le respect des normes existantes (comme IFC pour le bâtiment et CityGML pour les objets géographiques) : nous voulons, à terme, pouvoir représenter l'information d'un patrimoine urbain complet, avec une connaissance précise sur chacun des objets le composant, mais également sur l'ensemble du patrimoine stocké (liens entre les objets, droits d'accès...). L'affichage doit être simple et compréhensible par des utilisateurs non experts de la construction du bâtiment (gestionnaire de biens immobiliers par exemple), mais également complet et fournir les informations intéressantes à des experts. Cela passe notamment par la représentation en trois dimensions des informations avec un affichage propres aux besoins des utilisateurs et des objets représentés. De plus, ce système doit pouvoir être alimenté aussi bien avec des données issues de la gestion de patrimoine immobilier que des Systèmes d'Information Géographique et des modèles urbains.

⁵ www.active3d.net/fr/ils-nous-font-confiance

⁶ www.opengeospatial.org

⁷ <http://www.citygml.org/>

La maturité acquise, à la fois sur les aspects recherches et sur les aspects industriels, nous a permis de réaliser une critique constructive et empirique des solutions que nous avons déjà développées. De plus, avec le développement des technologies dans le domaine de la modélisation (connaissances, 3D...), il est concevable aujourd'hui, et dans une certaine continuité, de passer de l'échelle du bâtiment à celle du territoire et réciproquement.

Nous avons donc défini plusieurs objectifs afin de répondre à la problématique qui nous est posée. Nous voulons en effet représenter graphiquement en trois dimensions un ensemble d'objets du bâtiment et de la ville (*objectif : optimiser le rendu des scènes pour afficher un grand nombre d'objets sur de vastes étendues, positionner les objets sur des cartes et traiter les nombreuses sources de données possibles de façon homogène*), avec une gestion de la connaissance de ces objets (*objectif : stocker et manipuler la connaissance sur chacun des objets*) en proposant un rendu dynamique (*objectif : définir une architecture permettant le rendu dynamique de scènes depuis une base de données distante*) et personnalisé (*objectif : définir la notion de contexte et proposer un mécanisme de représentations personnalisées des objets*).

Le but est de faire évoluer la plateforme Active3D existante, résultant des travaux de recherche achevés fin 2007. Comme les outils de développement pour la gestion de modèles sémantiques et 3D se sont développés depuis et que les versions actuellement à disposition sont plus puissantes en terme de modélisation et plus rapides en terme d'exécution, nous avons choisi de réécrire une partie de l'application précédemment développée en la basant sur un nouveau modèle de représentation des connaissances et une nouvelle architecture. Nous nous sommes donc attachés à faire ressortir, d'après notre expérience et la littérature trouvée sur le sujet, les propriétés qui caractériseraient la notion de modélisation d'informations pour la gestion d'objets urbains et extra-urbains. Nous en avons dénombré treize :

- 1. un modèle pour la définition de modèles de données (aspects structurel et sémantique).** Ce « métamodèle » permettra la représentation dans un format unique des différents modèles hétérogènes intervenant dans la gestion technique de patrimoine. Cette propriété garantit la propriété d'extensibilité de notre solution (Nicolle, 2001). De plus, ce modèle devra permettre la définition de contraintes comme des contraintes de cardinalités ou de restrictions de domaines de valeurs.
- 2. un modèle permettant des inférences.** Outre des contraintes, le système devra permettre de définir des règles logiques (mécanismes de subsomption ou hiérarchies d'héritage). Ces règles seront utilisées pour classer des concepts entre eux et, pour chaque application, d'en déduire des propriétés algorithmiques liées à ce classement. Les logiques de descriptions apporteront des solutions pour ces problématiques.
- 3. un modèle disposant d'outils pour la manipulation des informations.** Les outils devront permettre de définir des processus pour parcourir le réseau d'informations et pour en sélectionner une sous-partie en fonction d'une requête.
- 4. un modèle garantissant l'évolutivité d'un modèle de données.** Le formalisme de modélisation de données choisi doit permettre de gérer l'évolution des modèles. Par exemple, le modèle IFC a évolué sur ses six dernières années par ajout de nouveaux concepts ou réécriture d'anciens concepts. Il est nécessaire de disposer d'une solution qui assure la compatibilité ascendante des données.
- 5. un modèle capable de gérer des données qui évoluent dans l'espace et le temps (géolocalisation, multireprésentation).** La dynamique demandée pour notre système de données implique que celui-ci doit pouvoir représenter les données en fonction de leur position dans l'espace et dans le temps. L'idée est de pouvoir mettre en place, au niveau sémantique, un mécanisme de multireprésentation des informations dépendant de ces deux dimensions.
- 6. un modèle garantissant la montée en charge.** Le système ne se limitant plus au bâtiment, il faut prévoir la montée en charge sur deux axes : le nombre d'éléments à traiter dans notre système d'information, et

l'augmentation de la taille des zones géographiques de répartition de ces éléments. Cette propriété doit notamment être vérifiée dans l'environnement de représentation 3D de la plateforme SIGA3D.

7. **un modèle permettant l'interopérabilité (respect des standards).** L'hétérogénéité est un des problèmes à résoudre. Une des solutions est de définir un formalisme commun et des règles de transformation pour permettre l'échange et la fusion de données issues de modèles hétérogènes (Nicolle, 2003).
8. **un modèle permettant la traçabilité.** Un autre besoin est le suivi dans le temps des informations. Une variable de temps est utilisée par le système pour garder un historique des informations et permettre la gestion des versions de celles-ci. L'évolution d'un patrimoine urbain pourra être suivie au cours du temps en interrogeant cette variable temps.
9. **un modèle permettant la gestion multilingue.** Les différents acteurs intervenant de manière collaborative dans le patrimoine immobilier peuvent être de différentes nationalités. Il est nécessaire que chacun se comprenne et que chacun puisse avoir accès aux données dans sa langue d'origine. Cet aspect est d'autant plus important que le projet SIGA3D possède une dimension européenne.
10. **un modèle permettant la gestion de divers types de données.** Notre objectif est de modéliser des informations de tous types : texte, images, 2D, 3D, documents... suivant les besoins de l'application métier qui est utilisée par l'acteur du patrimoine.
11. **un modèle permettant une gestion et une présentation des données selon un contexte d'utilisation** L'introduction de la notion de contexte permet d'extraire, de structurer et d'afficher les informations selon une vue utilisateur défini par un contexte d'utilisation. Il rend notamment possible la sélection d'une représentation d'un élément de notre modèle lorsque plusieurs ont été définies (point 5).
12. **un modèle permettant une utilisation dans des environnements mobiles.** Les environnements mobiles sont, à la base de notre projet, les environnements où les professionnels du bâtiment et de la gestion technique de patrimoine n'ont pas systématiquement accès à l'ensemble des informations qui lui sont accessibles (limite du réseau), ainsi qu'à la puissance des machines qui sont utilisées dans les conditions habituelles de travail. Ce phénomène est d'autant plus accentué aujourd'hui de par la prolifération des technologies mobiles ces 4 dernières années (Smartphones et ardoises tactiles)(White, 2011).
13. **Un modèle permettant d'établir des relations entre modèles.** Les objectifs précédents conduisent à la multiplication des modèles liés aux environnements urbains, ainsi qu'à la diversification de leur type (modèles de bâtiments, mais aussi modèles de réseaux ou modèles géographiques...). Il devient nécessaire de pouvoir créer des relations spatiales et temporelles entre ces différents modèles pour limiter les incohérences entre ceux-ci. C'est ce que nous appelons par la suite la synchronisation de modèles.

Cette liste de propriétés reprend et étend celle que Renaud Vanlande avait définie dans sa thèse. Certaines d'entre elles englobent un domaine plus large, par exemple la propriété 7 sur l'interopérabilité. Dans les travaux de Renaud Vanlande, cette interopérabilité visait à améliorer l'échange des informations entre les différents acteurs du bâtiment et leur logiciel propriétaire. Dans notre proposition, nous étendons cette notion pour que l'interopérabilité serve aussi à unifier les domaines du BIM et des Systèmes d'Information Géographique, en restant respectueux des aspects normatifs des deux domaines.

La Figure 1 présente une répartition de ces propriétés selon cinq grandes catégories. Elles orientent l'état de l'art sur les domaines de recherche qui ont déjà identifié ou traité des problèmes similaires. Nous voyons ainsi que nos travaux se positionnent de manière transversale sur plusieurs domaines.

- la *géométrie*, utilisée pour modéliser en 2D et en 3D l'information du patrimoine urbain, en gérant la montée en charge due au nombre et à l'espace des informations du référentiel,
- la *sémantique*, base de la modélisation et de la contextualisation des données, dans le cadre du projet SIGA3D à l'échelle du bâtiment, mais aussi de la ville et au-delà,
- la *spatialité* et la *temporalité*, c'est-à-dire la prise en compte et le traitement de l'information géographique et des éléments de temps,
- les *connexions*, regroupant les mécanismes de gestion des données intégrées ou liées (référéncées)

- la topologie, à la fois spatiale et temporelle, utilisée pour synchroniser des modèles.

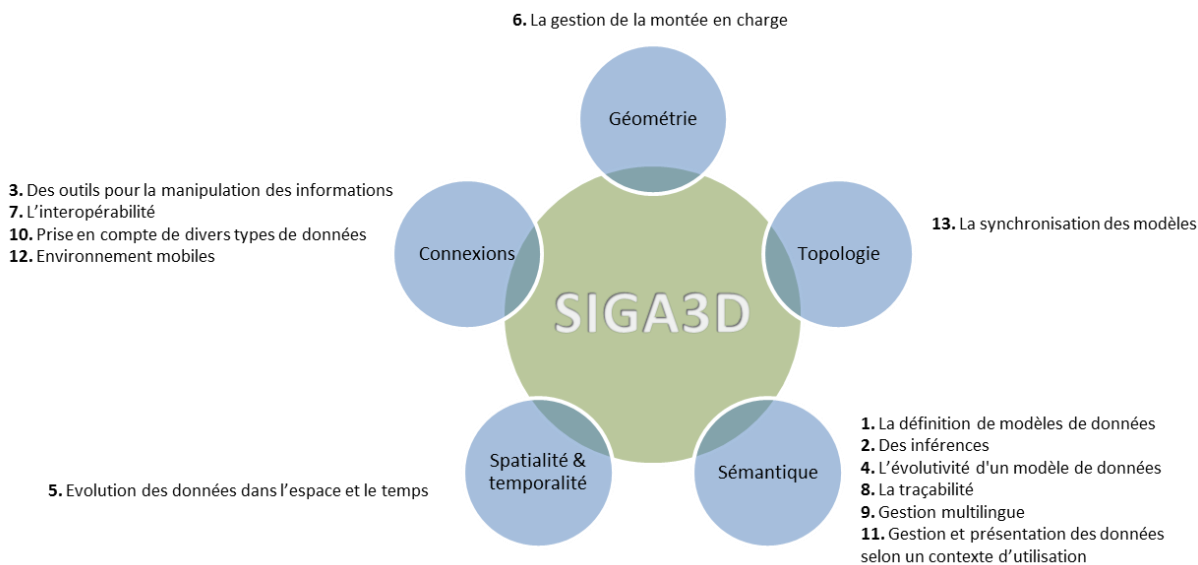


Figure 1. Classification des objectifs du projet SIGA3D

4 Organisation du document

Pour concevoir une solution qui réponde à ces différents objectifs, nous avons étendu la plateforme Active3D définie par Renaud Vanlande pour intégrer les contraintes des Systèmes d'Information Géographique et plus spécifiquement les notions de spatialité et de temporalité. Comme le nombre de données à représenter dans la scène 3D devient très volumineux (un environnement modélisé en 3D avec des objets urbains et des bâtiments IFC représentant plusieurs millions d'objets 3D), nous avons dû développer une nouvelle solution de gestion des niveaux de détail (LoD – Level of Detail). Cette solution innovante est basée sur une contextualisation sémantique des objets 3D à représenter. Ainsi, les contraintes géométriques ne sont pas les seules contraintes qui servent aux algorithmes à la gestion du niveau de détail. Nous avons introduit dans notre approche une gestion des niveaux de détails basée aussi sur la sémantique des objets et leur relation de pertinence avec le contexte de l'utilisateur.

Pour comprendre exactement la construction de la proposition réalisée dans ce travail de recherche, nous avons souhaité structurer ce document en ajoutant un chapitre contexte à la suite de l'introduction. Ce travail de recherche fait suite à 7 années de recherche et deux thèses. Il nous a paru primordial de replacer notre approche dans ce contexte.

Le chapitre 2 « contexte » présente une synthèse des travaux précédents, c'est une sorte d'état de l'art très spécifique sur les travaux d'Active3D. Une première collaboration de recherche a donné lieu à la première proposition d'indexation sémantique des grammaires XML d'une manière générale et des IFC en particulier. Plus tard, une deuxième collaboration a permis de développer une architecture complète d'indexation sémantique des IFC avec une gestion contextuelle des données. Cette partie nous permet d'introduire précisément le périmètre de recherche des travaux menés dans le cadre de cette thèse.

Le chapitre 3 « État de l'art » nous permettra de positionner nos objectifs de recherche par rapport aux travaux existants. La gestion de données urbaines nécessite tout d'abord une bonne connaissance des SIG. Ensuite, nous devons, dans ce domaine spécifique, étudier les normes liées à nos travaux. Active3D a en effet pour ambition de baser son développement sur l'usage de normes, comme elle l'avait fait pour les normes du

bâtiment avec les IFC. Une fois que ces deux éléments auront été étudiés, nous devons focaliser notre étude sur les propositions de modélisation sémantique des informations spatio-temporelles. Cette étude nous permettra d'identifier les verrous et les voies d'extension de l'architecture sémantique existante pour prendre en compte la gestion de ces données. Ensuite, nous devons étudier les solutions qui nous permettront de réduire le nombre d'objets urbains qui seront affichés dans le moteur 3D pour améliorer la rapidité et la pertinence du système. Enfin, nous devons étudier les différentes tentatives d'association des contraintes de la modélisation des bâtiments aux contraintes de modélisation de l'information géographique et vice-versa.

L'objectif du chapitre 4 est d'apporter une proposition formalisée au problème de la modélisation sémantique, graphique et géographique d'informations du bâtiment et d'objets urbains dans un contexte de gestion technique. Le but est d'élargir la portée des outils et techniques de modélisation connus dans le monde du BIM et dans les SIG. L'idée sous-jacente est de pouvoir traiter de manière uniformisée les capacités de modélisation, géométriques et sémantiques, de l'univers du bâtiment et les notions typiquement issues des SIG (géolocalisation, environnement urbain et terrains).

Par extension des travaux de Renaud Vanlande, l'approche de SIGA3D consistera à construire, d'un côté, un graphe de scène contenant les informations relatives aux géométries des objets, avec notamment la possibilité d'associer plusieurs représentations géométriques à ces objets, puis de l'autre côté de construire un graphe sémantique à l'aide de l'architecture SIGA3D. Ce graphe permet d'associer dans une ontologie plusieurs représentations sémantiques à un seul concept. En couplant ces deux approches, nous pouvons créer des niveaux de détail basé sur le contexte de visualisation des objets. L'objectif des niveaux de détail contextuels est donc d'optimiser le rendu géométrique des scènes en créant des niveaux de détail propres à chaque utilisation qui est faite des données. De sorte, la simplification des scènes est personnalisée et ne dépend plus des seuls critères géométriques, mais également de critères sémantiques.

Le chapitre 5, « Architecture SIGA3D », replace l'architecture dans son contexte industriel. Ce chapitre est une transition entre la phase de recherche portée par la collaboration de recherche et le travail de thèse et la phase de développement, portée notamment par un projet européen Euréka. L'objectif du projet Euréka est d'obtenir, au bout de 3 ans, un ensemble de fonctionnalités qui permettront de mettre en place l'extension d'Active3D pour la gestion technique de patrimoine urbain. Pour atteindre cet objectif, une architecture de processus plus complète a été développée autour du noyau d'architecture sémantique réalisé dans le cadre de ce travail de thèse. Cette architecture de processus est articulée en 7 Niveaux. Ces processus permettent la gestion des sources de données hétérogènes, la gestion des imports et exports des données, la gestion des connexions entre la base de données, les éléments web et le moteur 3D et enfin la gestion de l'affichage dans une interface Web. Les deux niveaux restants correspondant à la couche de modélisation sémantique des données de SIGA3D et à la couche de modélisation des contextes qui font l'objet de ce mémoire.

Le chapitre 6 présente les développements industriels réalisés dans le cadre du projet Euréka SIGA3D. Lors de ces développements, j'ai coordonné une équipe de développeurs présents au sein de la société Active3D, mais aussi Lat-Ion, partenaire allemand du projet. Outre ces aspects de coordinations, j'ai développé une grande partie des différents niveaux présentés dans l'architecture. L'ensemble des développements réalisés et présentés dans ce chapitre concerne l'implémentation de l'ontologie en base de données, le développement d'un nouveau moteur 3D et le développement d'un streaming pour la gestion des niveaux de détail sémantiques.

Chapitre 2

Contexte

Résumé

Le projet présenté dans ce document est la troisième collaboration entre le laboratoire et la société Active3D. L'objet de cette coopération concerne le développement d'une méthode d'indexation sémantique, puis la mise au point d'une architecture sémantique pour répondre aux contraintes du domaine de la gestion technique de patrimoine immobilier. Ce chapitre présente le contexte de collaboration autour de la norme IFC et des deux précédents projets réalisés.

Plan

1	Le modèle des IFC.....	33
1.1	Pourquoi les IFC.....	34
1.2	Présentation de la norme IFC.....	35
1.3	Exemple.....	38
1.4	Les IFC en pratique.....	40
2	Les ontologies.....	41
3	Active3D 2001-2004.....	42
4	Active3D 2004-2007.....	43
4.1	Modélisation de la connaissance.....	44
4.2	C-DMF.....	46
4.2.1	Définition de la brique « Data Model Framework ».....	48
4.2.2	Définition de la brique « Context ».....	48
4.2.3	Les opérateurs de graphe.....	49
4.2.4	Élément de structure des contextes.....	50
4.2.5	Exemple appliqué à la gestion de patrimoine immobilier.....	51
5	Active3D 2011-2012 : le projet SIGA3D.....	56

Les travaux de recherche s'inscrivent dans une continuité de collaboration avec la société Groupe Archimen depuis le début des années 2000, puis avec la société Active3D, créée en, 2005 pour porter le projet Active3D. De 2002 à 2004, une première collaboration de recherche a donné lieu à la première proposition d'indexation sémantique des grammaires XML d'une manière générale et des IFC en particulier. Cette indexation a permis la construction de la première plateforme Active3D avec une visualisation et une manipulation des objets du bâtiment dans un environnement 3D. De 2004 à 2007, une deuxième collaboration a permis de développer une architecture complète d'indexation sémantique des IFC avec une gestion contextuelle des données. Cette gestion a permis la construction de fonctionnalités d'adaptation dynamique des éléments manipulés en fonction du profil des internautes connectés. En 2008, la société a souhaité enrichir l'architecture développée pour étendre les fonctions de gestions techniques de patrimoine immobilier au domaine de la gestion technique de patrimoine urbain, tout en restant entièrement compatible avec la norme IFC. Depuis 2004, la plateforme Active3D est certifiée IFC⁸.

Ce chapitre présente ce contexte de collaboration. La première section présente la norme IFC, ses avantages et ses limites. La seconde section est une synthèse des travaux de Christophe Cruz, travaux réalisés dans le cadre de la première collaboration de recherche. La troisième section est une synthèse des travaux de Renaud Vanlande, réalisé dans le cadre de la deuxième collaboration de recherche.

1 Le modèle des IFC

Dans les projets de génie civil, les logiciels de CAO (Conception assistée par ordinateur) modélisent chaque élément du bâtiment par un ensemble de vecteurs. Dans ce formalisme, les informations sémantiques modélisées sur les objets du bâtiment sont très limitées (types de traits et quelques attributs). Pour résoudre cette problématique, l'IAI a proposé une norme appelée IFC qui décrit la représentation des objets rencontrés dans les projets de construction. Le format IFC est un modèle qui associe la sémantique métier à la géométrie 2D/3D des éléments constituant le bâtiment. L'addition de la sémantique métier permet de limiter les redondances d'informations, car elle identifie instantanément chaque élément composant le bâtiment, pour qualifier plus rapidement celui-ci. Les classes de base des IFC incluent la description des objets et fournissent une structure permettant l'interopérabilité des données entre les applications métiers. Par exemple, une porte IFC n'est pas simplement une collection de lignes et de primitives géométriques identifiées comme porte, mais cette porte est reconnue comme porte en tant que telle par la machine et possède des attributs correspondants à sa nature. L'adoption de ce format par tous les leaders des logiciels CAO permet une meilleure interopérabilité dans les échanges d'informations entre les diverses professions du génie civil. Chaque profession intervient dans le projet dans un contexte particulier, enrichissant le plan avec son propre vocabulaire, ses propres concepts et ses propres objets métiers. À la fin du projet, le fichier IFC correspondant au bâtiment contient l'ensemble de ses éléments, grâce à une définition multicontexte. Les fichiers IFC sont des fichiers textuels dont la taille peut atteindre 100 méga-octets. Plusieurs fichiers IFC peuvent coexister sur le même projet de génie civil. Par conséquent, l'objectif industriel fixé a été d'appliquer notre méthode d'indexation sémantique aux IFC. Cela permet de les intégrer à notre système avec d'autres données provenant des différents corps de métiers participants aux projets d'ingénierie civile.

Cette partie du chapitre explique l'origine des IFC les motivations qui ont conduit à leur développement. Un exemple illustrera son fonctionnement puis seront présentés les utilisateurs typiques de la norme aujourd'hui, et la façon dont on peut exploiter les informations du modèle.

⁸ La certification IFC est un programme buildingSMART qui délivre un label après un processus de certification (<http://www.buildingsmart-tech.org/certification/ifc-certification-2.0/ifc2x3-cv-v2.0-certification>).

1.1 Pourquoi les IFC

Pendant une opération de construction, une grande quantité de documents est échangée entre les intervenants : plans, notes de calcul, spécifications, compte-rendu de chantier... La plupart de ces documents sont disponibles sous forme de fichiers et l'on a vu se développer de nombreux modes de transmission (CD, pièces jointes dans les mails...). Chaque acteur utilise son traitement de texte, son tableur, son outil de CAO, son fax, le courrier et le mail pour atteindre ces objectifs. Apportant un confort non négligeable, ces outils ne permettent malheureusement pas une gestion coordonnée du projet : chacun possède son propre format de fichiers pour les plans et autres documents et chacun passe souvent plus de temps à convertir, retrouver, parfois même recréer les informations du projet, qu'au projet lui-même (Cruz, 2004b)(Vanlande 2007). Ces problématiques sont d'autant plus d'actualité que l'on parle davantage de développement durable depuis ces cinq dernières années et que des contrôles sont nécessaires de plus en plus tôt dans le cycle de conception du bâtiment (Bazjanac et al., 2011). Par exemple, c'est le cas du thermicien qui doit intervenir au plus tôt et faire des simulations dès que possible sur le futur bâtiment. En conséquence, l'interopérabilité entre les outils de l'architecte et ceux du thermicien est essentielle. Pendant longtemps, le format DWG, *standard de fait*⁹ pour les plans numériques, a été le principal support de ces échanges, mais il reste fermé (car propriétaire) et une seule entreprise (Autodesk^{®10}) décide de son évolution.

L'IAI s'est fixée pour but d'améliorer l'interopérabilité des logiciels utilisés dans le secteur de la construction. Elle réunit aujourd'hui plus de 500 membres qui sont regroupés en chapitres en fonction de la langue ou de la proximité : Amérique du Nord, Australie, Corée du Sud, chapitre francophone, chapitre Germanophone, Benelux, Chine... L'IAI ne produit pas de logiciels, mais des spécifications destinées à faciliter l'échange et le partage d'informations entre logiciels. Le principal résultat des travaux de l'Alliance est un langage qui rassemble aujourd'hui plus de 700 classes d'objets et a pour nom IFC. Plusieurs versions des IFC ont été publiées, du fait de l'accroissement progressif du domaine couvert. Depuis octobre 2000 et à la demande des éditeurs de logiciels, le cœur du modèle a été stabilisé pour plusieurs années. Cette partie des IFC, appelée plateforme, a obtenu l'homologation de l'ISO sous la référence ISO/PAS 16739 :2005¹¹ (Liebich, 2010). La prochaine version des IFC, nommée IFC4, devrait voir sa version finale publiée en même temps que la norme internationale ISO/IS16739 (le nom ISO de la norme pour les IFC4). Les IFC définissent un nouveau standard international, pour échanger des informations techniques entre logiciels dédiés à la construction et au génie civil. Disponible en import/export avec la plupart des nouveaux outils de CAO pour architectes, ce format apparaît également dans d'autres outils du bureau d'étude (calcul de structure, analyses thermiques, économie du bâtiment...) et des applications de gestion du patrimoine. Avec les IFC, on n'échange plus uniquement des informations géométriques, comme avec le format propriétaire DWG (Autodesk), mais de vrais objets de construction (murs, poteau, poutre, portes, fenêtres, étage, escalier...), le but étant d'obtenir un langage qui contient des informations sur la forme, les attributs des composants ainsi que les liens entre ces composants. Les IFC basent leur logique sur l'utilisation du BIM (appelé aussi « maquette numérique »). Le BIM est un outil de structuration et d'échange de données techniques, reconnu par tous comme un atout indéniable en termes de performances économiques comme énergétiques. Le terme « maquette numérique » est couramment utilisé en France, mais reste ambigu sur le fait que sa portée dépasse le simple aspect de maquette physique ou maquette 3D. Un BIM à la norme IFC produit par l'architecte peut être exploité par de nombreux logiciels et constituer la base d'évaluations du projet : consommations, impacts, coûts...

⁹ Le standard de fait désigne la solution d'un acteur dominant dans un domaine, les autres acteurs faisant en sorte d'être compatibles avec lui. Il s'oppose au standard ouvert dont nous discuterons ultérieurement dans ce chapitre.

¹⁰ <http://www.autodesk.fr/>

¹¹ ISO/PAS, Spécification publiquement disponible (Publicly Available Specification). Ce type de normes correspond à des spécifications élaborées à l'extérieur de l'ISO et ayant fait l'objet d'un vote positif de la majorité des pays membres du comité.

Les applications informatiques utilisent aujourd'hui les IFC pour construire une base de données numérique orientée-objet. Celle-ci peut alors être partagée à tous les participants d'un projet et s'enrichir tant que le projet se poursuit, de la conception à l'exploitation en passant par la construction. Le conseil européen du génie civil estime que l'utilisation des IFC peut réduire les facteurs de risques pour les gestionnaires de patrimoine immobilier de 20% sur les nouvelles installations, et au-delà de 50% pour les anciennes structures (ECCE, 2002).

La première version des IFC a été publiée en 1998 (Release 1.5). Elle a été très rapidement remplacée par la version 2.0. Aujourd'hui c'est la version 2x3 qui est couramment utilisée dans le milieu de la construction et de la gestion des bâtiments. Ces évolutions ont permis à cette dernière version d'être acceptée en tant que norme. Depuis 2007, un gros travail de sensibilisation et de promotion sur l'usage des IFC est fait. En France, par exemple, cette initiative a donné lieu au projet national eXpert¹² qui vise à mener plusieurs projets de construction en se basant sur la norme IFC pour tous les échanges effectués, depuis l'architecte jusqu'au gestionnaire du bâtiment en passant par les différents bureaux d'étude et de contrôle (calcul thermique, structure, économie...).

Un autre aspect des IFC est le travail collaboratif. Si les IFC répondent parfaitement à leur rôle de format d'échange de données, ils apportent également des éléments de réponses sur le travail collaboratif, et notamment sur la modification collaborative des maquettes numériques. En effet, les fichiers IFC sont partagés entre plusieurs personnes (soit de manière centralisée, soit par échange de fichiers). Afin de résoudre certains problèmes d'accès et de compétences sur les modifications des plans, déjà facilités par le partage d'une même information au format IFC, buildingSMART (sur une proposition de Tekla¹³ et Solibri¹⁴) se penche sur l'élaboration d'une nouvelle norme spécialisée dans le domaine du BIM, openBIM Collaboration Format (BCF) qui crée un workflow facilitant la communication entre les différents outils de création du BIM (Stangeland, 2011).

1.2 Présentation de la norme IFC

Actuellement, la version stable utilisée est issue des spécifications IFC2x3 TC1 et comporte 653 classes ainsi que 327 Property Sets¹⁵ (Pset). La prochaine version, actuellement en release candidate 2, contient 759 classes et 393 Pset. La Figure 2 présente l'architecture générale de la spécification IFC 2x4 RC2. La base de cette architecture est une simplification de celle des IFC en version 2x3 (plus facile à lire). Cette architecture est destinée aux développeurs de logiciels qui veulent travailler avec les IFC (CAO par exemple) et est transparente pour l'utilisateur final. On peut distinguer quatre couches dans celle-ci : les cercles supérieurs regroupent les différents domaines, la ligne des rectangles juste en dessous forme la couche d'interopérabilité, les deux lignes en dessous (carrés + triangles) forment le cœur de l'architecture (avec le noyau et les extensions du cœur du schéma), et enfin les octogones forment la couche des ressources. Ces ressources sont organisées sous forme de graphes hiérarchiques comme l'illustre la Figure 3. Chaque couche de la plateforme ainsi définie comprend plusieurs catégories dans lesquelles sont définies les différentes classes. Les classes IFC sont entièrement décrites sur le site officiel de l'IAI¹⁶.

¹² <http://www.mediaconstruct.fr/index.php?q=node/190>

¹³ <http://www.tekla.com/>

¹⁴ <http://www.solibri.com/>

¹⁵ Les property sets (ensemble de propriétés) définissent toutes les propriétés extensibles de manière dynamique.

¹⁶ <http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm>

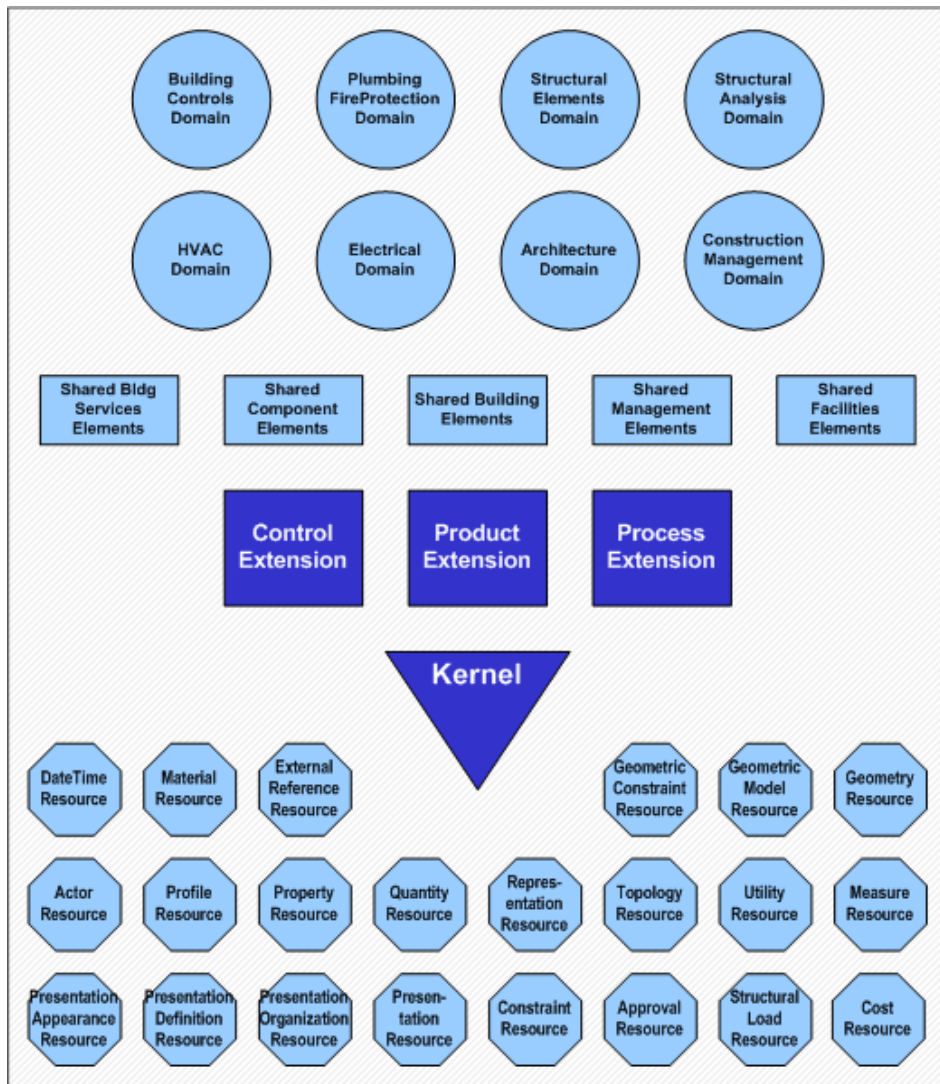


Figure 2. Schéma de l'architecture IFC2x4 RC2 (Source: IFC2x4 RC2 Final Online Documentation)

Le modèle de données IFC ne représente pas seulement les éléments tangibles d'un immeuble comme les murs, les portes, les plafonds, le mobilier... mais aussi des concepts plus abstraits comme des plannings, des activités, des espaces, organisations, coût de construction... sous forme d'entités (classes). Chacune de ces classes possède un certain nombre de propriétés comme le nom, la géométrie, les matériaux, les finitions, les relations... (cf. Figure 3). Le modèle IFC définit également, en plus des classes et des Pset, des types de données définies (comme `IfcAreaMeasure` qui définit l'aire mesurée d'une surface), des types d'énumérations (comme `IfcRoleEnum` pour définir les rôles qui peuvent être joués par un acteur), des types de sélection (comme `IfcColour` qui définit l'apparence basique des éléments à afficher), des fonctions (`IfcVectorSum`), des règles (`IfcSingleProjectInstance` qui vérifient qu'une seule instance de `IfcProject` existe).

Pour le format de fichier, l'IAI a retenu pour les IFC la norme internationale ISO 10303-21 (format STEP). Chaque ligne du fichier est composée de valeurs d'attributs dédiés ou hérités et commence par un identifiant de ligne et le nom de la classe IFC décrite. Le Script 1 est un extrait de fichier IFC, avec l'entête du fichier et les premières informations sur le propriétaire et l'application qui a servi à générer le fichier.

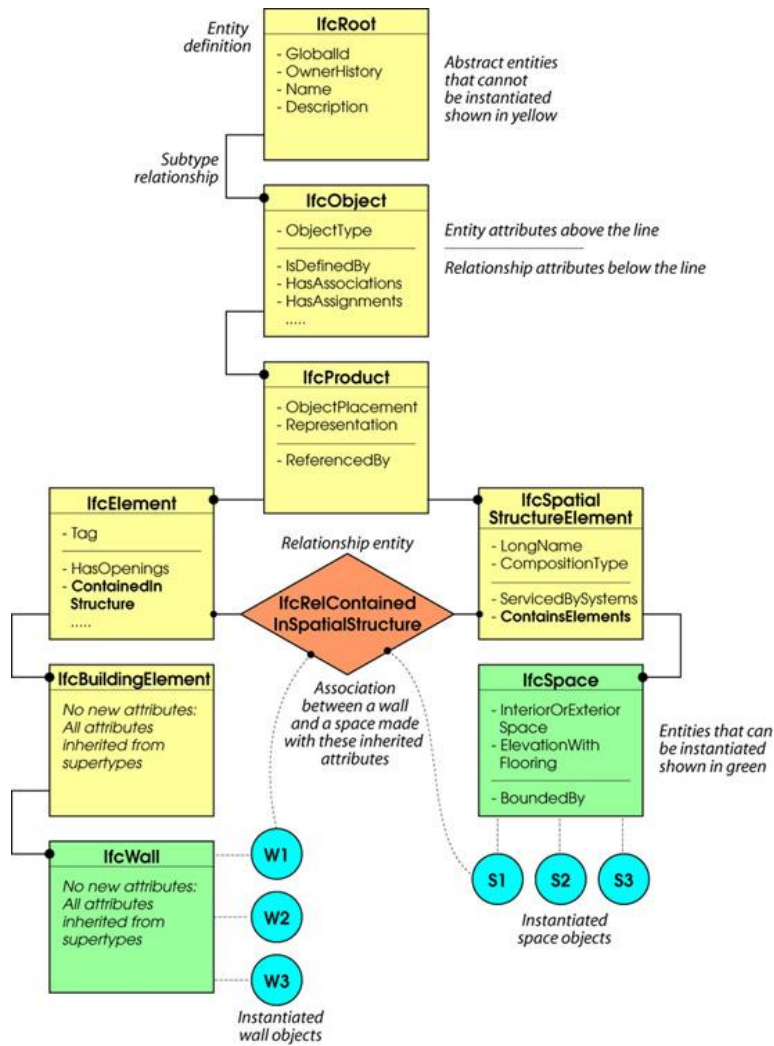


Figure 3. Structure des classes IFC (Source : (Khemlani, 2010))

L'IAI propose également une solution basée sur XML, ifcXML, qui contient le modèle des IFC sous la forme d'un schéma XML et constitue une alternative au format STEP. Cette solution permet d'utiliser XML, un format soutenu par le W3C¹⁷, supporté et reconnu internationalement, comme principale technologie pour la publication et l'échange des informations. Cependant, la taille occupée par un fichier ifcXML est souvent beaucoup plus importante que le fichier qui décrit l'IFC en STEP (de 4 à 8 fois). De fait, cette solution n'est que très peu utilisée.

Un autre format très peu utilisé est l'IFCZip¹⁸. Celui-ci consiste en fait uniquement en la compression des fichiers .ifc ou .ifcxml avec un algorithme de compression PKzip 2.04g. Le gain en taille sur les fichiers peut être très important (de 60% à 80% pour les IFC décrits avec la norme STEP, et de 90% à 95% pour les IFC décrits en XML). Cependant ce format nécessite une décompression préalable du fichier avant un import dans les logiciels compatibles.

Un logiciel comme Active3D qui fait de la gestion de patrimoine immobilier n'a pas besoin d'utiliser toutes les classes IFC disponibles pour atteindre ses objectifs. En implémentant un peu plus de 200 classes dont les plus couramment utilisées : IfcSpace, IfcBuilding, IfcBuildingStorey..., il est possible de charger

¹⁷ W3C est un organisme de normalisation chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, SPARQL, CSS, PNG, SVG et SOAP. <http://www.w3.org>

¹⁸ <http://www.buildingsmart-tech.org/implementation/ifc-implementation/ifc-impl-agreements/cv-2x3-154>

dans l'application toutes les informations dont les gestionnaires de patrimoine immobilier peuvent avoir besoin. Si l'application rencontre une classe qui n'est pas connue, elle est ignorée (un avertissement apparaît dans le journal qui stocke les actions). Ce modèle est bien entendu facilement extensible et les informations sont organisées de manière personnalisée pour chaque utilisateur de la plateforme.

1.3 Exemple

L'exemple de structure de fichier IFC donné dans Script 1 décrit un bâtiment avec plus de 111000 objets métier (une ligne par objet). Pour comprendre la complexité des IFC, cette section présente le modèle des IFC et l'étude des instances IFC.

```
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 Exemple de fichier IFC

Les fichiers IFC sont faits d'objets et de liens entre ces objets. Les attributs des objets décrivent la « sémantique métier » des objets. Les liens entre les objets sont représentés par des « éléments de relation ». Le modèle des IFC est un modèle objet réalisé à l'aide du langage EXPRESS (norme internationale ISO10303-11¹⁹). Pour information, le schéma XSD²⁰ sur lequel est basé le format ifcXML est généré depuis la représentation EXPRESS principale par un langage de liaison décrit dans ISO10303-28ed2²¹. Ce modèle décrit approximativement 600 classes. Il y a trois types de classes d'IFC : les classes « objet », les classes « relation » et les classes « ressource ».

¹⁹ http://www.iso.org/iso/fr/catalogue_detail?csnumber=18348

²⁰ XML Schema, <http://www.w3.org/XML/Schema>

²¹ http://www.iso.org/iso/fr/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40646

1. Les classes « objet » sont composées de triplets (GUID, OS, FU) qui correspondent respectivement à un identifiant unique global (Global Unique Identifier), aux informations sur le propriétaire (Ownership) et aux unités fonctionnelles. Les unités fonctionnelles décrivent le contexte d'utilisation des classes (par exemple la représentation géométrique, sa localisation, sa composition...). Un exemple d'unité fonctionnelle est une « fenêtre double vitrage avec un châssis en PVC », et l'unité à considérer est « un mètre carré d'éléments d'ouverture remplis par ce produit ». Pour la description des unités fonctionnelles, on utilise les classes « ressource » (Vanlande et al., 2008). Le GUID nous assure d'identifier à coup sûr l'objet dans le plan, même si le plan qui le contient a été modifié. En effet, les logiciels qui éditent et exportent des IFC ne génèrent qu'une seule fois l'identifiant unique à la création de l'objet et celui-ci ne change donc pas si l'objet n'est pas modifié. Cela nous permet, dans les logiciels de gestion de patrimoine immobilier, de stocker les informations sur un objet avec une référence à son identifiant unique. On peut ainsi en cas de mise à jour de la maquette IFC, retrouver systématiquement les informations attachées à cet objet. Dans le Script 1, l'élément #5 de type `IfcPersonAndOrganisation` référence les éléments #3 et #4.
2. Les classes « ressource » constituent l'ensemble des attributs utilisés dans la description des unités fonctionnelles. Ces ressources sont organisées dans un graphe hiérarchique.
3. Les classes « relation » représentent les diverses relations entre les classes « objet » et possèdent aussi des unités fonctionnelles. On y retrouve par exemple des relations de capacité, de contenance, d'agrégat... Les éléments de relation correspondants sont préfixés par `IfcRel`. L'élément `IfcRelAggregates` du Script 1 ayant l'identifiant #111030 constitue une relation d'agrégat entre l'élément #30 et la liste d'éléments suivants (# 34, # 16236, # 29699, # 56800, # 62077, # 67336, # 72633, # 91702, # 110989). L'élément #110989 est également référencé par l'élément #111029 qui est un lien appelé `IfcRelContainedInSpatialStructure`. Ceci signifie que si un élément peut être référencé par plusieurs éléments alors deux éléments peuvent mutuellement se référencer par l'intermédiaire d'une ou plusieurs relations. Cette référence mutuelle forme un graphe cyclique.

L'étude des fichiers IFC montre la complexité du chevauchement entre les instances de classes « relation », et les instances de classes « objet ». À ce niveau, il existe deux types de lien entre les objets. Nous les appelons les liens directs et indirects. Les liens indirects sont définis par les instances des classes de relations. Les liens directs sont définis par les instances des classes « ressource ». Les liens indirects sont caractérisés dans les Figures 4 et 5 par un triangle. Les instances de classes « objet » du champ architectural deviennent les éléments sémantiques. Dans ces figures, ces éléments sont graphiquement représentés par un rond. Les instances de classes « ressource » sont représentées par un losange. La Figure 4 montre les liens indirects entre les éléments sémantiques, en utilisant un élément de relation. La Figure 5 montre les liens directs entre les éléments sémantiques, ils sont notés en pointillés. Il y a deux types de liens directs. Le premier type définit les ressources de l'élément. Ces ressources sont structurées en utilisant une arborescence. Le deuxième type définit un lien direct entre deux éléments sémantiques. Le modèle IFC définit seulement un type de lien entre deux éléments sémantiques, c'est le lien de placement entre les éléments sémantiques, pour la conception d'un bâtiment dans une scène 2D/3D. Cette relation est effectuée par l'attribut `IfcLocalPlacement` de l'élément sémantique. Elle définit la marque de référence de l'objet courant comparée à la marque de référence de l'objet de père de la relation directe. L'ensemble des liens directs formé par les attributs `IfcLocalPlacement` forme une structure arborescente de la scène 2D/3D. La difficulté principale consiste à manipuler en même temps les éléments sémantiques cycliques et la structure hiérarchique des éléments 3D.

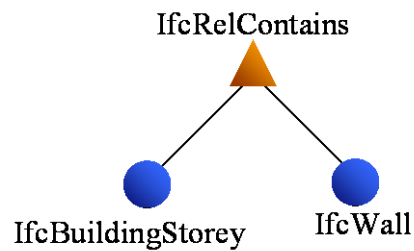


Figure 4. Un étage et un mur sont reliés par un élément de relation IfcRelContains

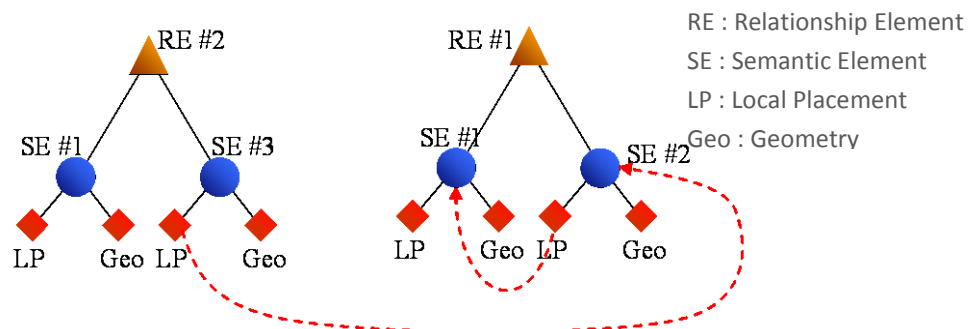


Figure 5. Exemple de liens directs entre les éléments sémantiques

1.4 Les IFC en pratique

L'utilisation des IFC commence en premier lieu par la possibilité de produire des maquettes numériques dans ce format, que ce soit lors de phase de conception pour un bâtiment ou du relevé de l'existant. Aujourd'hui la majorité des logiciels de CAO savent exporter de l'IFC même si tous ne sont pas certifiés IFC. Les trois éditeurs principaux utilisés en France que sont Autodesk (États-Unis), Nemetschek²² (Allemagne) et Bentley²³ (États-Unis) sont tous trois certifiés IFC. Ils participent activement au groupe de travail sur l'élaboration de la norme IFC, au même titre que Active3D et d'autres sociétés. En effet, la réutilisation des maquettes numériques IFC produite par les logiciels de conception architecturale ne peut se faire sans les outils adaptés pour les bureaux d'études ou gestionnaires de patrimoine. Ainsi, de nombreuses entreprises en informatique se sont positionnées avec succès sur le créneau des logiciels techniques spécialisés, et se sont investies dans le développement d'outils informatiques respectant la norme IFC, notamment dans le cadre de nouvelles réglementations techniques. Parmi les solutions les plus innovantes et remarquées en France, et avec lesquelles nous avons eu des contacts dans le cadre de projets innovants (projet eXpert en particulier), nous pouvons citer Advance Béton de Graitec²⁴ pour les calculs de structure, ClimaWin de BBS-Slama²⁵ pour les calculs thermiques et Winquant de Attic+²⁶ pour l'économie du bâtiment.

Les IFC constituent la base des travaux de recherche menés par Active3D depuis sa création. La prochaine partie de ce chapitre introduit la notion d'ontologie afin de se familiariser avec le vocabulaire inhérent à ce

²² <http://www.nemetschek.fr/>

²³ <http://www.bentley.com/fr-FR/>

²⁴ <http://www.graitec.com/fr/index.asp>

²⁵ <http://www.bbs-slama.com/>

²⁶ <http://www.attic-plus.fr/>

domaine, omniprésent dans les différents travaux présentés dans ce document. Ensuite, les travaux effectués pour exploiter les IFC seront présentés dans les deux parties suivantes.

2 Les ontologies

Avant de rentrer dans le vif du sujet, faisons une rapide présentation de la notion d'ontologie. Pour définir une ontologie, il est nécessaire de partir du concept de réseau sémantique. L'idée de réseau sémantique est attribuée à Quillian (Quillian, 1968) pour l'avoir utilisé dans la modélisation du fonctionnement de la mémoire. L'idée centrale des réseaux sémantiques est de décrire la réalité sous forme de graphes (réseaux) composés de nœuds représentant les concepts, reliés par des arcs, exprimant les relations entre les concepts.

Les nœuds et les arcs sont en général étiquetés. Aux premiers sont associés les objets (concept, évènement, situation...), aux seconds les relations entre les objets, d'où la désignation de « structure objet-relation ». Les liens les plus spécifiques de ce type de réseau sont les liens « sorte-de ». D'autres liens sont utilisés comme « à pour partie », « instrument pour »... Le lien « sorte-de » permet de définir l'une des notions les plus importantes des réseaux sémantiques et de tout autre formalisme intégrant un minimum de logique, qui est la déduction par héritage des propriétés. L'héritage est essentiellement basé sur la transitivité de la relation « sorte-de ». En effet, attacher une caractéristique commune à un ensemble de concepts à un niveau élevé d'une hiérarchie permet d'éviter de stocker en mémoire l'ensemble des propriétés pour chaque concept.

L'usage de la notion de réseau sémantique a été étendu à la représentation des connaissances, en particulier pour modéliser les ontologies. On appelle ontologie, un réseau sémantique qui regroupe un ensemble de concepts décrivant complètement un domaine ou une partie d'un domaine. Les ontologies sont utilisées à des fins de définition, de représentation de fait ou de raisonnement. Elles ont pris un essor important avec le développement du Web sémantique. Une ontologie fournit une référence pour la communication entre les machines, mais aussi entre humains et machines en définissant le sens des objets. Outre ses apports évidents par rapport à des approches classiques d'annotation par mots clefs, les ontologies sont utilisées pour le partage et la réutilisation de données, ainsi que pour l'amélioration de la communication. Une ontologie possède des caractéristiques qui favorisent la réutilisation et le partage de données. Cependant, la conception d'ontologies communes occasionne des difficultés. Dans (Gruber, 1993), l'auteur définit trois niveaux de convention pour mettre en place des techniques d'interopérabilité basées sur les communications (le format de représentation du langage, le protocole de communication des agents et la spécification du contenu du vocabulaire partagé). La communication peut se catégoriser en trois types distincts : communication homme-homme, homme-système ou système-système. Ils possèdent chacun des caractéristiques particulières qui posent des problèmes solubles par les ontologies.

Pour synthétiser, les ontologies permettent de standardiser le vocabulaire, d'uniformiser le langage d'échange entre les différents acteurs, de comparer les différents systèmes, de structurer la connaissance pour simplifier l'analyse et la synthèse des connaissances d'un domaine, et de spécifier un contexte. En particulier, l'ontologie est devenue incontournable dans l'adaptation de l'information à l'utilisateur en permettant de modéliser à la fois les utilisateurs et le domaine de manière intelligible pour tout type d'utilisations et d'acteurs. Le contenu d'une ontologie varie aussi avec le type d'ontologie considéré : une ontologie de domaine contiendra des connaissances propres à un domaine de connaissances (par exemple, la construction), une ontologie de tâche contiendra des connaissances propres à une activité (par exemple, la gestion technique), une ontologie de haut niveau contiendra des connaissances abstraites très générales, destinées à rassembler d'autres ontologies (par exemple, des notions d'entité, d'évènement, de rôle, etc.). Le contenu dépendra aussi du degré de formalisation (langage naturelle, langage restreint, formalisme simple, logiques complexes...).

3 Active3D 2001-2004

L'idée initiale du projet Active3D était d'indexer sémantiquement des objets dans une scène 3D. Cette indexation permettait de manipuler les objets 3D indépendamment de la structure de la scène. La scène était décomposée en un ensemble d'objets 3D sémantiquement identifiés. Cette décomposition permettait de stocker la scène, par le contenu, dans une base de données relationnelle. Les éléments pouvaient ensuite être combinés entre eux et mis en relation avec d'autres données non graphiques pour construire une interface 3D de manipulation de données. Ce premier travail a été réalisé en collaboration par Christophe Cruz et Christophe Nicolle en 2001. Dans la méthode développée à l'époque, le langage X3D²⁷ (successeur XML du langage VRML²⁸) était utilisé.

Si les travaux menés par Christophe Cruz ont été appliqués au formalisme des IFC dans le projet Active3D, la méthode Active3D a consisté, tout d'abord, à formaliser les schémas et les données XML (Berstel et al., 2000). Le but est de construire une ontologie générique qui permet d'effectuer une correspondance sémantique entre les éléments structurels des documents XML (Nicolle, 2005). Une fois cette ontologie construite, il est possible de définir des ontologies de domaine et d'intégrer des données hétérogènes dans ces ontologies.

Pour construire l'ontologie générique, un ensemble de règles a été défini par Christophe Cruz. Celles-ci permettent de formaliser la construction des grammaires XML et de définir les concepts, relations et attributs de l'ontologie générique. Ces règles fournissent un vocabulaire formant la taxinomie du système. Ce vocabulaire est composé des mots *concept*, *relation*, *facteur conceptuel*, *facteur relationnel*, *attribut*, *facteur attribut*, *élément sémantique* et *élément relationnel*. Chacun de ces mots représente un concept de l'ontologie et correspond à une classe. Celles-ci peuvent être considérées comme des métaclasse dont les instances sont des classes qui définissent les ontologies de domaine. La métaclasse concept permet de définir des classes comme « mur » spécifiant les attributs du concept « mur » dans une ontologie du domaine du bâtiment par exemple. Ces métaclasse conservent le lien entre les classes et les instances. Ceci permet une recherche rapide dans le système d'information soit à l'aide des métadonnées, soit directement sur les données. L'un des points fondamentaux des travaux de Christophe Cruz concerne la modélisation des contextes.

Un contexte est une vue utilisateur du système en fonction d'un domaine d'application spécifique. Cette vue est un arbre XML que nous appellerons arbre contextuel. Les arbres contextuels sont le résultat de l'extraction des données du document intégré dans un document XML indépendamment de leur structure originelle. Dans ce système, un objet qui possède plusieurs descriptions dans des contextes d'utilisation différents intégrera l'ensemble de ces descriptions dans le document intégré. Ainsi, l'utilisateur de cet objet dans un contexte spécifique pourra obtenir une définition de l'objet plus riche que celle initialement décrite dans ce contexte.

Soit un objet « mûr » et un objet « fenêtre » décrit par deux documents XML. Le premier document est fourni par l'architecte. Il donne une description géométrique des deux objets. Le second document est fourni par un thermicien. Il définit les propriétés thermiques des deux objets. L'intégration de ces documents permettra de construire une définition de l'objet mûr et de l'objet fenêtre, intégrant les données géométrique et thermique. Si le thermicien souhaite modifier les données thermiques de ces objets, il pourra obtenir une visualisation graphique de ces objets.

À l'inverse, l'utilisation des arbres contextuels permettra de réduire la définition d'un objet aux seules données pertinentes pour les domaines d'application correspondant au contexte. Dans notre exemple, l'architecte peut vouloir extraire du système intégré seulement la description géométrique de la fenêtre. L'avantage de cette approche est de limiter la taille du flux d'information en augmentant la pertinence des données transmises.

²⁷ <http://www.web3d.org/realtime-3d/x3d/what-x3d/>

²⁸ Virtual Reality Modeling Language, <http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/>

Dans ces travaux, Christophe Cruz a étudié et développé des algorithmes de création et manipulation d'arbre pour l'appliquer aux traitements d'un modèle objet complexe comprenant une quantité importante d'information. Suite aux problèmes rencontrés dans la pratique, pour le traitement de ce système d'information spécifique au monde du bâtiment, il a dû trouver des optimisations sur la manipulation des structures arborescentes et des graphes. Le but était de formaliser et généraliser les algorithmes de traitements et d'optimisations appliqués au modèle IFC.

La définition d'un contexte permet de sélectionner un ensemble d'éléments sémantiques par rapport à : un ensemble de types de relation, un ensemble de types d'éléments attributs et un ensemble d'attributs simples. La sélection par rapport aux éléments de relations ne concerne que les éléments sémantiques référencés comme père ou fils par un élément de relation défini par le contexte. Le résultat de cette requête est un arbre contextuel formant ainsi une vue métier sur les données. Cette requête est un filtre sémantique sur les informations, donnant ainsi une vue personnalisée des données. Grâce à un ensemble de contextes, l'intervenant d'un projet a accès à un ensemble de vues métiers sémantiquement différentes et dépendantes d'un contexte.

Les travaux d'Active3D ont montré que la valeur d'un élément sémantique dépend de son contexte d'utilisation. En définissant un contexte d'utilisation des éléments sémantiques, il a été montré qu'il est possible de réduire le flux des informations en augmentant sa pertinence. La notion de contexte permet de définir un arbre contextuel qui sera extrait du système. Cette approche permet, d'une part, de cibler l'information pertinente pour l'intervenant d'un projet. Cette propriété permet de limiter la taille des données à déplacer sur le réseau. D'autre part, elle permet de coupler des informations qui ne l'étaient pas au départ. Cette propriété permet de fusionner des informations hétérogènes. Par exemple, le système autorise la fusion des informations thermiques et la représentation graphique des données dans un seul document. Néanmoins, ce système possède des inconvénients. Tous les arbres contextuels extraits du système d'information ne sont pas valides par rapport aux schémas XML intégrés précédents. Par conséquent, il n'est pas possible de valider les documents XML créés. De plus, si l'on désire créer une vue X3D d'un arbre contextuel, il est absolument nécessaire de créer une feuille de style XSL pour traduire l'arbre sémantique en arbre X3D ou de développer une interface graphique qui interprétera les arbres sémantiques de la définition d'un arbre contextuel.

Pour répondre à ces interrogations et permettre l'évolution de la plateforme vers des fonctions de gestion technique de patrimoine permettant une gestion collaborative du bâtiment, une nouvelle collaboration de recherche a été entamée entre 2004 et 2007. L'objectif de cette collaboration portée par la thèse de Renaud est la définition d'un modèle de données répondant à un cahier des charges précis pour permettre d'atteindre ces objectifs. Durant cette période, la formalisation d'une nouvelle architecture a été réalisée : C-DMF (Context-Data Model Framework). La finalité de cette architecture est la création d'ontologies dynamiques et évolutives. L'architecture propose un ensemble de constructeurs qui permettent la création de modèles de données contextuels appliqués à la gestion de patrimoine. Dans la suite de ce document, nous présenterons les objectifs visés dans ce travail et nous décrivons les concepts de cette architecture.

4 Active3D 2004-2007

La thèse de Renaud Vanlande est née avant tout du besoin d'avoir un modèle de données dynamique pour faire de la gestion de patrimoine et suivre par exemple les évolutions de l'état d'un bâtiment. Les objectifs que devait remplir le modèle de données du patrimoine sont les suivants :

1. un modèle structurel et sémantique permettant une définition de modèles de données,
2. un modèle logique permettant des inférences,
3. un modèle disposant d'outils permettant la sélection des informations,
4. un modèle permettant de gérer l'évolution d'un modèle de données,
5. un modèle qui permet de gérer la fusion de données s'appuyant sur des modèles de données différents,

6. un modèle gérant le cycle de vie des informations, c'est-à-dire d'assurer la pérennité et de gérer les informations dans le temps,
7. un modèle permettant la gestion de plusieurs langues,
8. un modèle permettant la manipulation de types d'informations divers (texte, images, 3D, ...),
9. un modèle permettant une gestion et une présentation collaborative et adaptative des données.

La difficulté pour la réalisation d'un tel modèle réside dans la complexité que forme l'ensemble de ces contraintes hétérogènes. Pour résoudre cette problématique, il a été décidé de s'orienter vers des modèles de représentation des connaissances intégrant des mécanismes logiques pour réaliser différents calculs sur les données. Pour bien comprendre l'approche proposée par Renaud Vanlande, nous avons souhaité rappeler brièvement les différents modèles de représentation de connaissance.

4.1 Modélisation de la connaissance

Plusieurs formalismes ont été développés pour modéliser la connaissance, que l'on peut classer de plusieurs manières. La Figure 6 illustre une organisation des principales familles de formalismes selon les principaux domaines d'études de la représentation de connaissances : le génie logiciel, l'intelligence artificielle et le Web sémantique.

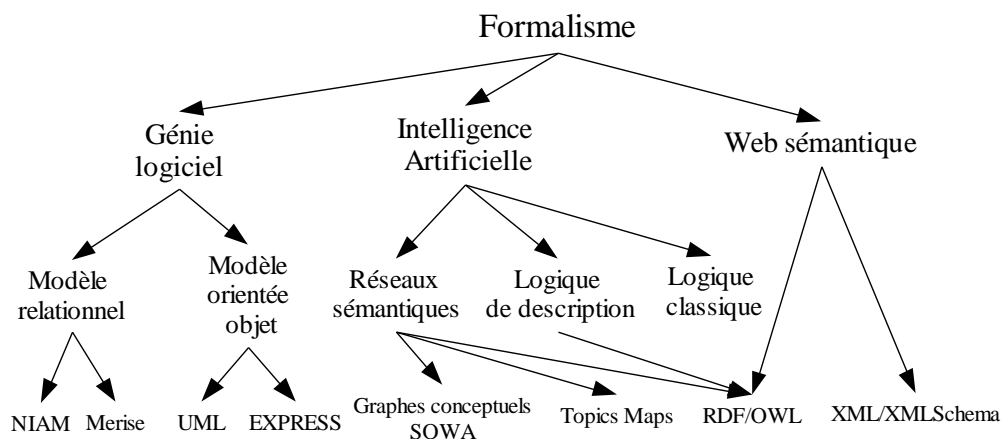


Figure 6. Famille de formalisme par domaine d'étude

Les logiques de description et les réseaux sémantiques (plus particulièrement les graphes conceptuels) sont les principaux formalismes utilisés pour décrire des bases de connaissances de nature ontologique. Dans sa thèse, Renaud Vanlande fait une comparaison des différents formalismes évoqués dans ce schéma et constate notamment qu'aucun d'entre eux ne présente une réponse globale à l'ensemble des besoins évoqués. Il construit alors son propre formalisme de représentation des données, C-DMF, en se basant sur certains du web sémantique (RDF, OWL et SWRL²⁹). Nous allons présenter rapidement ces trois formalismes.

La pyramide présentée à la Figure 7 est composée des langages du web sémantique et constitue une des versions proposées par le W3C. On y retrouve RDF et la partie ontologie qui englobe notamment OWL et SWRL. Une caractéristique de tous ces langages est d'être systématiquement exprimable et échangeable dans une syntaxe XML. Ceci permet de bénéficier de l'ensemble des technologies développées autour d'XML : XML

²⁹ Semantic Web Rule Language, Un langage de règles sémantiques combinant OWL et RuleML : <http://www.w3.org/Submission/SWRL/>

Schema, outils d'exploitation des ressources XML (bibliothèques JAVA...), bases de données gérant des fichiers XML.

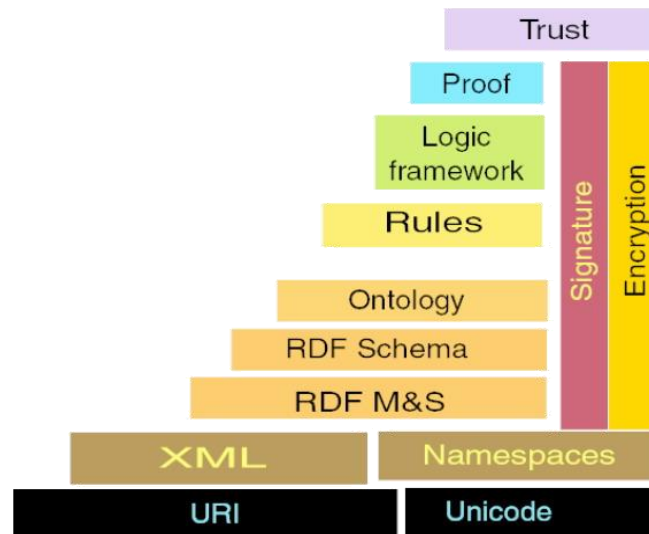


Figure 7. Pile des langages du web

Resource Description Framework (RDF) est un langage développé par le W3C et conçu pour apporter une couche sémantique au Web. Il permet de relier des ressources web entre elles par des arcs orientés et étiquetés. La structure d'un document RDF est un graphe complexe, orienté et étiqueté. Un document RDF est composé d'un ensemble de triplets {sujet, prédicat, objet}, le prédicat reliant le sujet à l'objet. Le prédicat est appelé également propriété. En termes de graphe, le sujet et l'objet sont des nœuds reliés par un arc allant du sujet vers l'objet. Les nœuds et les arcs sont de type ressource. Une ressource est identifiée par une URI (Uniform Resource Identifier). Le Script 2 ci-dessous nous montre un exemple de fichier RDF/XML (il existe plusieurs façons d'écrire le formalisme RDF).

```
<rdf:RDF
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mb = "http://www.a3d.net/modelBatiment#"
  xmlns    = "http://www.a3d.net/Batiment#"
>

  <mb:Batiment rdf:ID="b1">
    <mb:contenir rdf:ressource="#e1"/>
  </mb:Batiment>
  <mb:Etage rdf:ID="e1"/>
</rdf:RDF>
```

Script 2. Exemple de syntaxe RDF/XML

Ontology Web Language (OWL) décrit la structure d'un domaine en termes de classes et de propriétés comme les approches orientées objet. Il est inspiré des logiques de description et fournit un très grand nombre de constructeurs permettant de définir de manière très fine les propriétés des classes. Néanmoins certaines caractéristiques font qu'il n'existe aucun algorithme d'inférence décidable. Par exemple une classe peut être traitée simultanément comme une collection d'individus ou comme un individu. De ce fait, OWL est décliné en sous-langages (OWL-Lite, OWL-DL et OWL-Full) de plus en plus expressifs conçus pour faire un compromis entre son pouvoir expressif et son pouvoir de raisonnement. Le Script 3 est un exemple qui définit une classe et une relation (un bureau est une sous-classe de Local) dans la syntaxe OWL.

```
<owl:Class rdf:ID="Bureau">
<rdfs:subClassOf rdf:resource="Local"/>
</owl:Class>
```

Script 3. Exemple de Syntaxe OWL

Semantic Web Rule Language (SWRL) vise à unifier OWL et le langage de règles d'inférences RuleML³⁰. Il constitue un rapprochement entre les langages OWL et les règles logiques. Il enrichit la sémantique d'une ontologie définie en OWL. Contrairement à OWL, SWRL permet de manipuler des instances par des variables. Il ne permet pas de créer des concepts et des relations. SWRL répond aux besoins d'expressions de règles sur des instances. La construction d'une règle est basée sur le schéma suivant: *antécédent* → *conséquent*. La règle présentée dans le Script 4 présente la syntaxe de SWRL et indique que tout local contenant un bureau sur lequel travaille une personne contient alors une personne.

```
Local (?x) ^Bureau (?y) ^Personne (?z) ^contient (?x, ?y) ^travailleSur (?y, ?z) →
contient (?x, ?z)
```

Script 4. Exemple de Syntaxe SWRL

Une des problématiques rencontrées dans ce projet est de trouver un moyen pour faire comprendre à une machine la notion de contexte : suivant un contexte, une définition de concept est différente. Pour cela il est rappelé le fonctionnement des Named graphes. Les Named Graphs constituent une proposition pour l'implémentation de la couche « Trust » de la pile du Web Sémantique. Un Named Graph est un graphe RDF auquel on assigne un nom sous la forme d'une URI. Les noms des graphes peuvent se référer au contenu du graphe lui-même, à d'autres graphes ou à rien. Les Named Graphs apportent une nouvelle dimension aux graphes RDF : les informations sont décrites non plus avec des triplets, mais avec des quadruplets. Cette nouvelle dimension permet d'encapsuler chaque triplet afin, par exemple, de contextualiser les informations. La caractéristique clé des Named Graphs par rapport à d'autres propositions de contextualisation est le fait qu'ils proposent une solution minimaliste et incrémentale dans la construction du Web sémantique.

4.2 C-DMF

L'objectif du projet est d'utiliser un modèle pour décrire un patrimoine immobilier qui respecte toutes les contraintes énoncées en introduction de la section 3. L'idée de définir un nouveau formalisme vient du fait que OWL et SWRL, qui rappelons-le définissent des opérateurs de modélisation et des règles logiques au-dessus de modèles de données générique et extensible définis avec RDF, ne sont pas complètement adaptés aux besoins du projet et montrent une certaine ambiguïté dans l'utilisation de leurs opérateurs. Par exemple la règle SWRL suivante *Homme (?x) → Personne (?x)* possède la même sémantique que la définition OWL suivante (Script 5) :

```
<owl :Class rdf:id= "Homme">
  <owl:subClassOf rdf:resource= "#Personne"/>
</owl:Class>
```

Script 5. Exemple de Définition de Hiérarchie OWL

La Figure 8 présente une extension de la pile du web sémantique donnée par Tim Berners-Lee. Cette extension correspond à l'ajout de la brique C-DMF qui représente la proposition de la thèse de Renaud. Cette brique est

³⁰ Rule Markup Language, langage de modélisation de règles : <http://ruleml.org/>

basée sur un noyau RDF et unifie, dans un ensemble restreint, des opérateurs d'OWL, SWRL, RuleML avec une gestion des contextes dérivés des Named Graph.

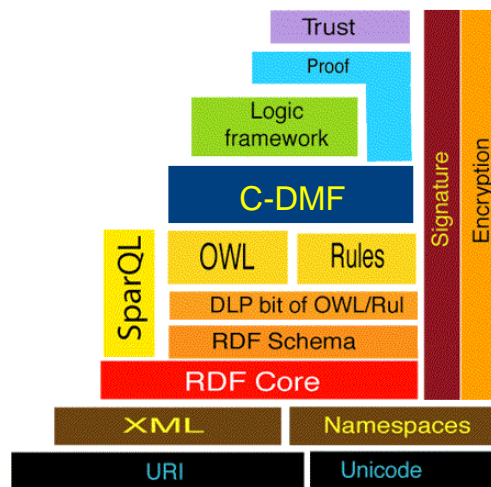


Figure 8. Placement de C-DMF dans la pile du Web sémantique

La Figure 9 présente l'architecture globale de C-DMF. Elle présente les deux briques C et DMF qui composent ce langage, pour pouvoir traiter à la fois le modèle de données (DMF) et la gestion des contextes (C). Nous allons, dans la suite de ce chapitre, présenter chacune des deux briques.

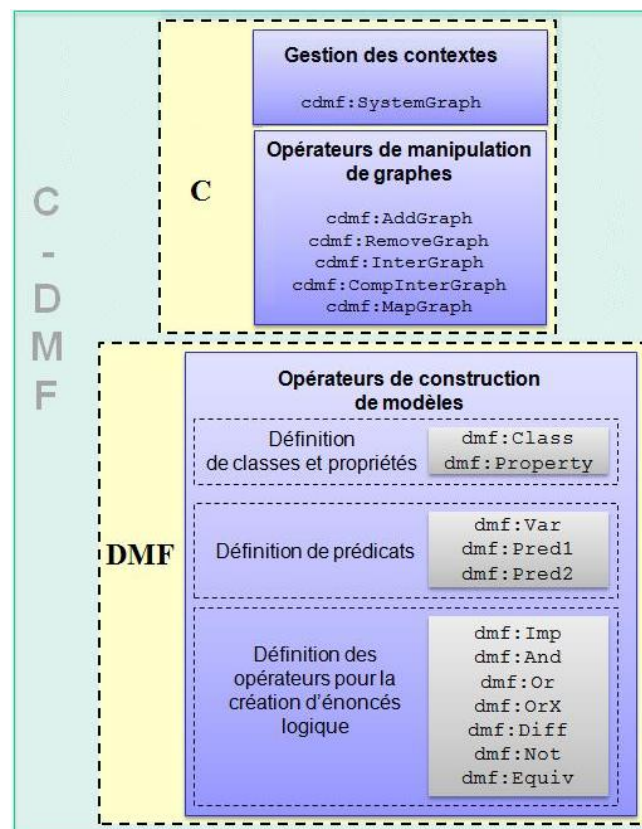


Figure 9. Architecture de C-DMF

4.2.1 Définition de la brique « Data Model Framework »

Cette brique répond aux besoins de description ontologique par la définition de règles. Comme nous pouvons le voir dans la Figure 9, DMF (Data Model Framework) définit les opérateurs de construction de modèles et est composée de trois blocs : définition des classes et des propriétés, définition des prédicats (groupe d'individus identifiés par des variables) et définition des opérateurs (d'implication, d'intersection, d'union...) pour la création d'énoncés logiques (à l'image de ce que fait le couple OWL et SWRL). Le concept important de cette brique est l'utilisation d'un opérateur d'implication (composé d'une condition et d'une conséquence) qui permet de constituer des règles pour exprimer des contraintes sur des prédicats. Cet opérateur permet de retranscrire la plupart des opérateurs d'OWL et de SWRL (comme les sous-propriétés, les propriétés transitives ou symétriques, les inverses...) avec un nombre restreint d'éléments et dans un seul formalisme. Par exemple, l'énoncé suivant : $\text{Bureau}(?x) \rightarrow \text{Zone}(?x)$ signifie en logique du premier ordre que tout individu de type Bureau est un individu de type Zone. Cette expression se traduit en DMF par le bloc XML suivant (Script 6) :

```
<dmf:Class rdf:ID='Zone' /> //définition de classe
<dmf:Class rdf:ID='Bureau' />
<dmf:Var rdf:ID='_x' /> //déclaration de variables
<dmf:Pred1 rdf:ID='pred1_0'> //définition de prédicat
  <dmf:classPred rdf:resource='#Zone' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:Pred1 rdf:ID='pred1_1'>
  <dmf:classPred rdf:resource='#Bureau' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:Imp rdf:ID='imp_0'> //définition de règle
  <dmf:argCond rdf:resource='#pred1_1' />
  <dmf:argCons rdf:resource='#pred1_0' />
</dmf:Imp>
```

Script 6. Exemple de création de modèles avec DMF

4.2.2 Définition de la brique « Context »

Après avoir rapidement expliqué comment est défini le langage de modélisation des données, DMF, nous allons nous intéresser plus en détail à la contextualisation des données. Comme le montre la Figure 9, la brique C de cette architecture est composée de deux blocs. La Figure 10 nous montre la hiérarchie des graphes C-DMF. On y voit apparaître un graphe générique, *CdmfGraph* dont tous les autres graphes héritent. L'élément Graph désigne un graphe simple composé d'un ensemble de triplets RDF. Le premier bloc de la brique C définit les opérateurs de combinaisons de graphes *AddGraph*, *RemoveGraph*, *InterGraph*, *CompInterGraph*, *MapGraph*. Le second bloc est constitué de l'élément *SystemGraph* et ses propriétés. Le fonctionnement de cette brique est le suivant : Un contexte est défini à l'aide d'un *SystemGraph*, et un graphe de données DMF lui est associé. Ce graphe peut éventuellement être une opération de graphes afin de simplifier l'écriture des graphes en général.

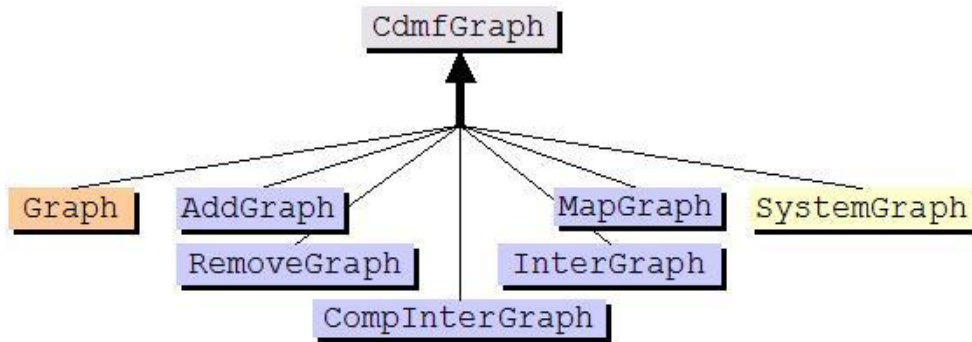


Figure 10. Hiérarchie des graphes C-DMF

4.2.3 Les opérateurs de graphe

L'objectif de la définition d'opérateurs portant sur des graphes est d'améliorer les traitements lors de la manipulation de graphes RDF, de réduire le volume de données et de simplifier la maintenance des données en rendant les sources d'informations uniques. Les opérateurs définits dans C-DMF sont les suivants : `AddGraph` réalise l'union de plusieurs graphes, `RemoveGraph` réalise la suppression d'un graphe dans un autre, `InterGraph` fait une intersection entre deux graphes, `CompInterGraph` est semblable au précédent, mais permet une intersection sur le sujet ou l'objet d'un triplet et enfin `MapGraph` définit des règles à appliquer sur un graphe, engendrant ainsi un nouveau graphe. La Figure 11 est un exemple d'utilisation de l'opérateur `AddGraph` sur deux graphes. `G1` définit deux objets `b1` et `e1` de type `Batiment` et `Etage`. `G2` définit une relation `contenir` entre `b1` et `e1`. Les ressources `b1` et `e1` dans `G1` et `G2` sont les mêmes ressources, car leurs URI sont identiques. Le graphe `Ga1` est le résultat de l'union.

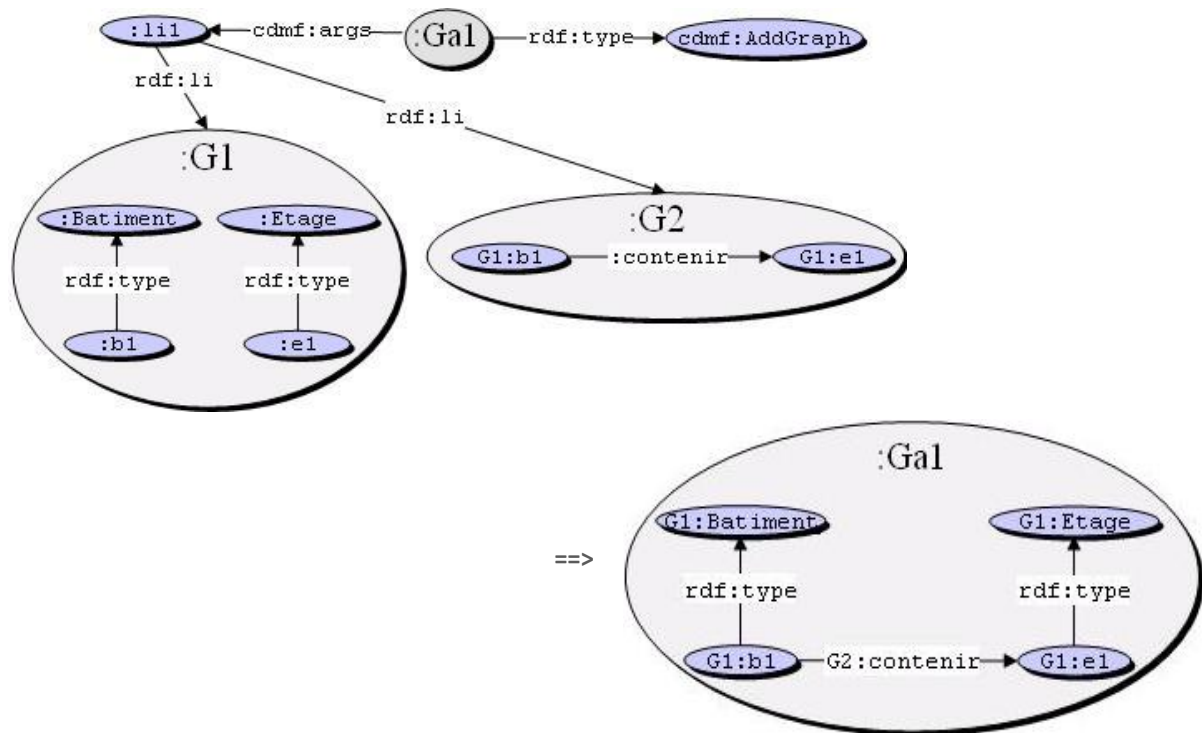


Figure 11. Vue graphique de l'union de graphes

4.2.4 Élément de structure des contextes

Le graphe particulier `SystemGraph` est certainement le cœur de l'innovation de l'architecture C-DMF. Celui-ci est associé à un graphe de données DMF (de type `CdmfGraph`) et définit pour celui-ci la notion de contexte en reprenant notamment le concept des `Named Graph`. Il contient des propriétés définissant des informations telles que le type de graphe, ce qu'il modélise, son contexte d'accès ou encore les opérations applicables sur le graphe (par exemple il va indiquer le fait qu'un graphe représente le patrimoine de telle personne à telle date). Toutes les propriétés définies dans un `SystemGraph` s'appliquent aux données du graphe associé. Nous allons décrire le fonctionnement du `SystemGraph` en basant nos explications sur l'exemple présenté dans la Figure 12. Le `SystemGraph` que nous allons décrire est celui nommé `:S0`. Il est associé au graphe `/Graph/G0` comme le définit la propriété `cdmf:graph`. On voit tout d'abord qu'il contient une propriété `cdmf:model` : celle-ci définit le modèle sur lequel s'appuie ce `SystemGraph` et indique les objets et les propriétés qui vont pouvoir être générés dans le graphe associé. Le fait d'associer un `SystemGraph` à un modèle permet également d'effectuer un contrôle de cohérence de données. Le modèle d'un `SystemGraph` est également un `SystemGraph`. Comme nous pouvons le voir dans l'exemple, C-DMF définit un `SystemGraph` particulier de type `SystemModelGraph` (ici appelé `Sm0` et incorrectement typé sur le dessin) dont le modèle fait systématiquement référence au modèle DMF. Ce dernier contient un modèle, c'est à dire des définitions de classes, de propriétés et de règles définies dans la syntaxe DMF (`/Graph/G1`).

La deuxième propriété que l'on trouve dans les `SystemGraph`, `cdmf:of` est celle qui définit le contexte en associant une ressource ou liste de ressources RDF (requête) au graphe. Dans l'exemple, le contexte nous indique (par une liste de ressources RDF comme l'indique la propriété `rdf:bag`) que le système est la représentation du Lycée Carnot dans sa Version 2 et est accessible par la ressource `userdata:User1` qui représente l'utilisateur Vanlande Renaud.

La troisième propriété importante qui apparaît est `cdmf:action` qui définit les actions d'ajout et de suppression réalisables sur le graphe associé. Il est possible de définir quelles actions sont possibles, sur quelle partie des données et à partir de quel modèle. Si aucune action n'est paramétrée, alors seule la visualisation est autorisée. À noter que l'action de suppression peut se faire de deux manières : soit l'on supprime directement la ressource dans le graphe associé, soit, si ce dernier est un `cdmf:RemoveGraph`, il est possible de lui ajouter la ressource. Dans l'exemple, toutes les opérations d'écriture et de suppression sont autorisées.

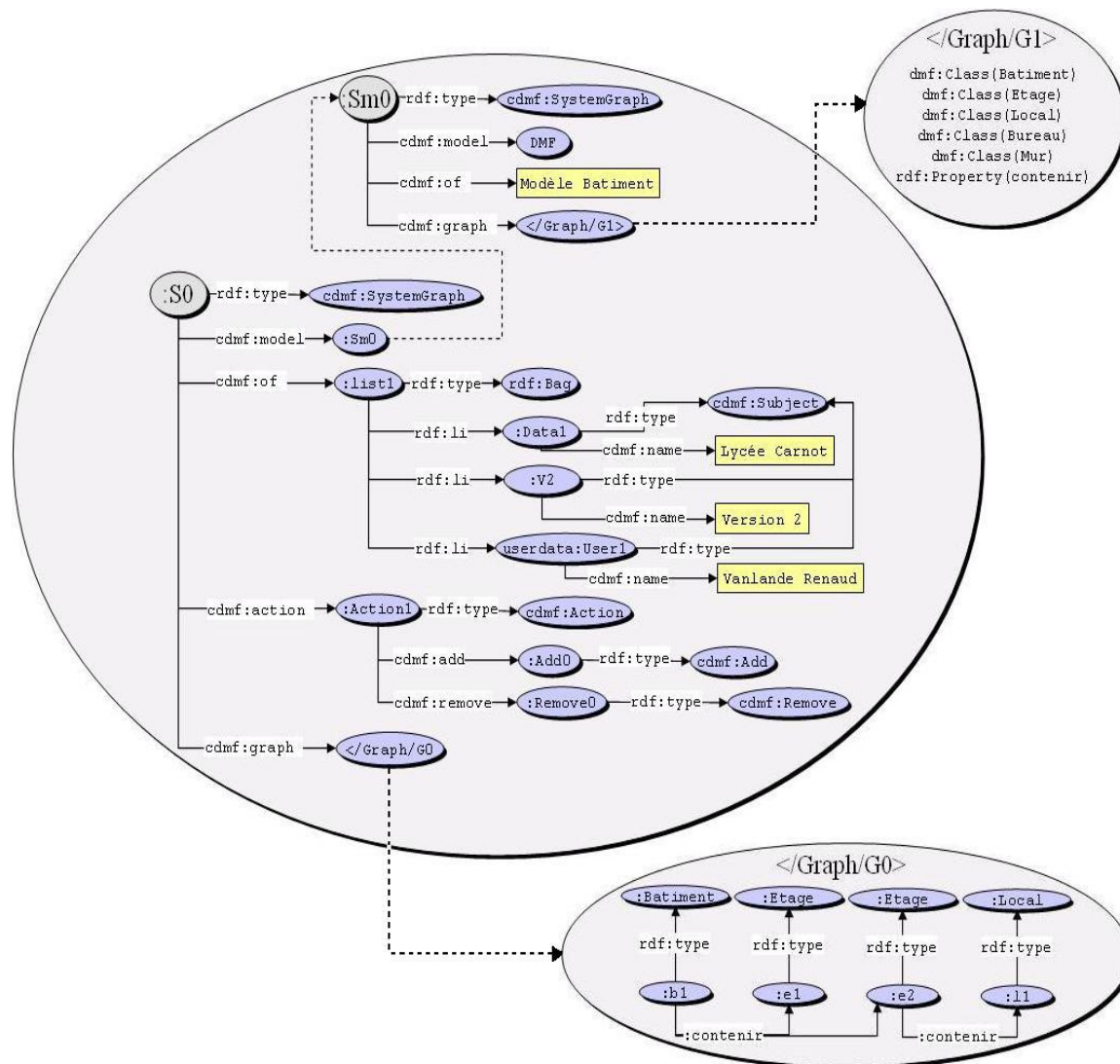


Figure 12. Exemple de SystemGraph

Maintenant que nous avons présenté l'idée générale de la proposition de Renaud Vanlande, nous pouvons expliquer comment cette architecture peut être utilisée pour résoudre la problématique posée pour la gestion de patrimoine.

4.2.5 Exemple appliqué à la gestion de patrimoine immobilier

Le but de cette partie est de montrer par l'exemple comment C-DMF permet de répondre aux exigences fixées pour le projet de gestion de patrimoine immobilier. Certaines d'entre elles sont inhérentes au fait que C-DMF est basé sur RDF, par exemple la possibilité de définir ses propres modèles de données, le calcul d'inférences et les outils de sélection des informations. Pour cela nous pouvons utiliser les mêmes outils que nous utilisons pour RDF. L'auteur a fait le choix d'utiliser JENA³¹ pour la manipulation des données RDF et le moteur d'inférence. Le requêtage des données, quant à lui, se fait à l'aide de SPARQL³². Les autres propriétés sont réalisées par la construction de modèles particuliers à l'aide de C-DMF. Chacune des fonctionnalités demandées peut être résolue de différentes façons, aussi nous ne présentons ici qu'un exemple simple de

³¹ Framework de développement d'applications sémantiques : <http://jena.apache.org/>

³² Langage de requête pour RDF : <http://www.w3.org/TR/rdf-sparql-query/>

réalisation des fonctionnalités. L'exemple se décompose comme suit : initialisation du projet et notion de `SpaceSystem`, création du système représentant un modèle de bâtiment puis un bâtiment, définition d'un système de versionning, représentation multilingue et mise à jour du modèle de document avec possibilité de transformer les bâtiments existants d'un modèle à un autre.

- Initialisation du projet et création d'un bâtiment.** L'espace de configuration de l'environnement applicatif C-DMF s'appelle le `SpaceSystem`. Il est lui-même défini comme un `SystemGraph` dont le modèle de données associé est la description de la brique C. Il est composé d'un document RDF principal qui contient l'ensemble des `SystemGraph` contenant eux même les données d'initialisation du projet. Dans notre exemple, le `SpaceSystem` nous permet de créer des `SystemGraph`, basés sur le modèle DMF. Nous devons créer un `SystemModelGraph` pour pouvoir ensuite créer un `SystemGraph` associé à un bâtiment. Pour Active3D, ce `SystemModelGraph` est basé sur le modèle de bâtiment IFC comportant plus de 600 classes d'objets. Dans notre exemple, nous avons simplifié le modèle au maximum (Figure 13).

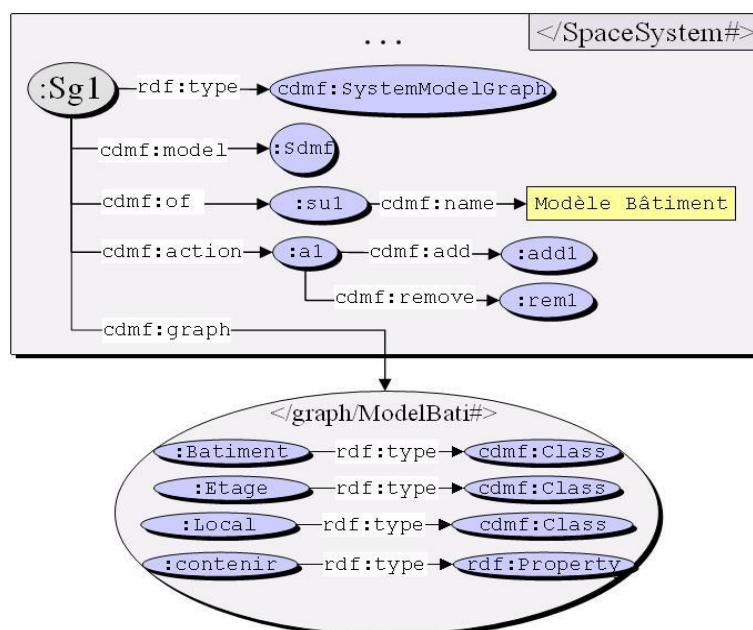


Figure 13. Exemple de SpaceSystem pour Active3D

Une fois le modèle du bâtiment défini, nous pouvons créer un bâtiment A. Pour cela, nous créons un nouveau `SystemGraph` représentant ce bâtiment A (Figure 14) dont le modèle est `Modele Batiment` et le contexte est `Bâtiment A`. Nous pouvons créer et supprimer des données et celles-ci se situent dans le graphe `</graph/BatA#>`. Le système créé, nous générons un bâtiment `b1`, un étage `e1` et une relation de contenance entre le bâtiment et l'étage.

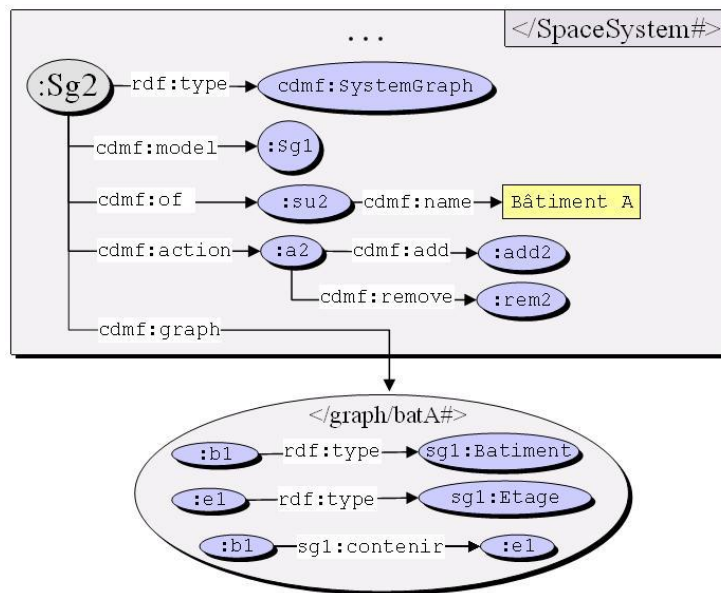


Figure 14. Exemple de bâtiment

- **Création d'un système de versioning multilingue.** Après avoir créé un bâtiment, nous allons adapter ce modèle pour effectuer les quelques opérations demandées pour ce projet. Nous allons commencer par étudier un système de versioning. Pour cela nous allons utiliser les possibilités offertes par C-DMF comme la contextualisation et les opérations sur les graphes pour le faire.

La Figure 15 nous montre les *SystemGraph* utilisés pour réaliser le système de versioning des bâtiments. On remarque tout de suite que l'étiquetage des graphes permet de stocker plusieurs versions d'un même bâtiment, ici la date nous aidant à les différencier, et les opérations sur les graphes nous permettent de stocker seulement les différences qu'il y a entre deux versions du bâtiment. Ainsi sur l'exemple ci-dessous, le *SystemGraph* *Sg4* qui représente le bâtiment A à la date du 12/05/2003 est associé à un graphe de type *addGraph* qui ajoute au graphe du même bâtiment à une date antérieure un autre graphe contenant les modifications (ajout d'un étage *e2* et de deux locaux *l1* et *l2*).

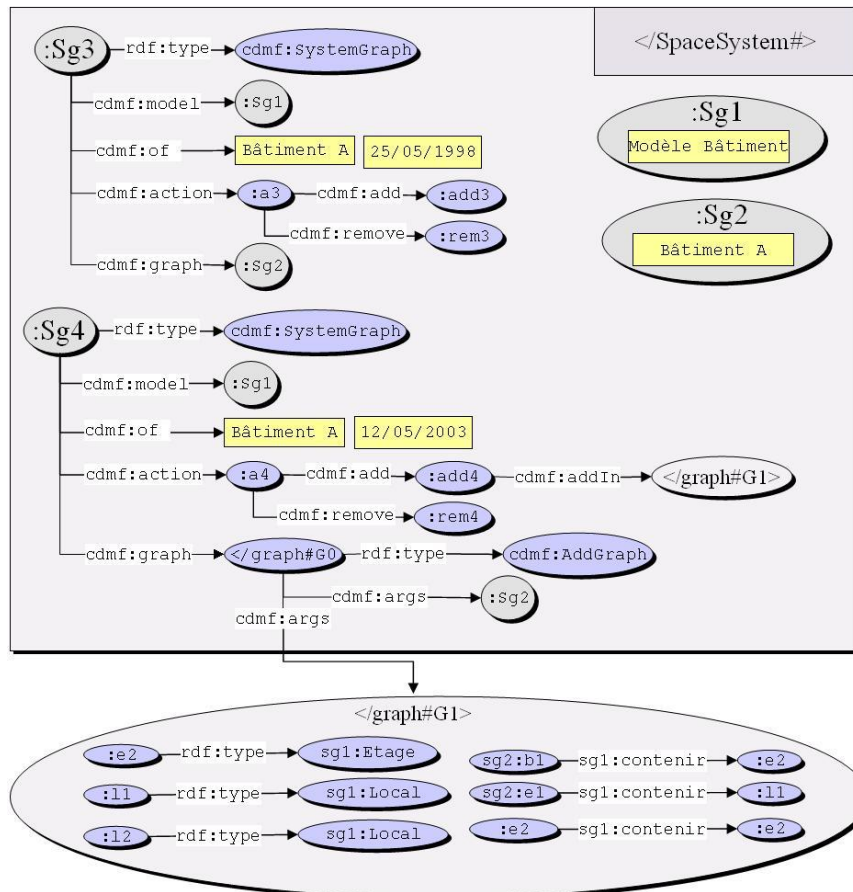


Figure 15. Application de C-DMF à un système de suivi des modifications d'un bâtiment

- La gestion des mises à jour des modèles de données avec C-DMF.** L'exemple proposé pour simuler la mise à jour d'un modèle de données (par exemple mise à jour du modèle IFC) est illustré avec la Figure 16 et le Script 7. On commence par définir complètement notre nouveau modèle de données pour le bâtiment (dans un graphe simple), contrairement à l'exemple précédent où l'on ne stockait que les différences entre deux versions. Dans ce nouveau modèle la notion d'étage disparaît pour le bâtiment, la classe Local et la propriété contenir sont respectivement devenus Espace et estComposeDe. Le but de la mise à jour du modèle de données est de pouvoir faire évoluer tous les bâtiments existants dans l'ancien modèle vers le nouveau. Pour cela, nous utilisons un graphe de type correspondance (MapGraph) qui va nous permettre de définir des règles de transformations. : on crée un SystemGraph Sg8 dont le graph associé G4 correspond au graphe Sg2 (bâtiment A) transformé à l'aide des règles définies dans le graphe G5. Ce dernier nous indique que dans le modèle Sg1, le bâtiment correspond au bâtiment du modèle Sg7, que la classe Local correspond à la classe Espace et que toutes les propriétés contenir attachées à l'étage deviennent des propriétés estComposeDe attachées au bâtiment lié avec la propriété contenir à l'étage qui disparaît.

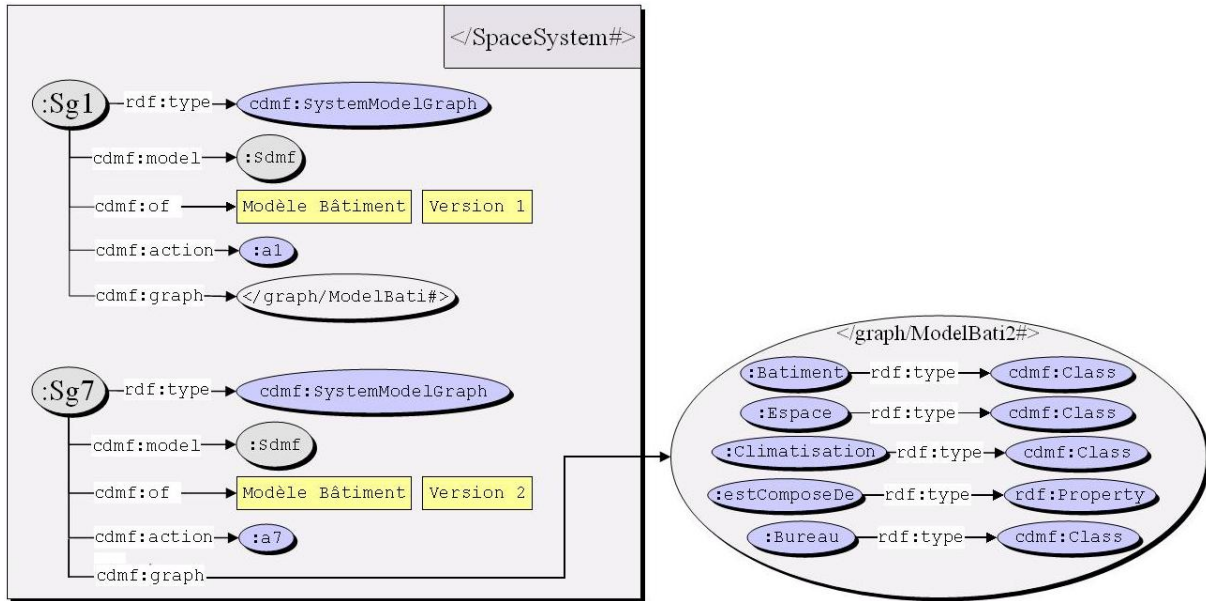


Figure 16. Définition de deux modèles de données avec C-DMF

```
// URI du graphe: </SpaceSystem#>
...
:Sg8      rdf:type      cdmf:SystemGraph
:Sg8      cdmf:model    :Sg7
:Sg8      cdmf:of       :su2
:Sg8      cdmf:graph    </graph#G4>

-----
// URI du graphe: </graph#>
:G4      rdf:type      cdmf:MapGraph
:G4      cdmf:src      </spaceSystem#Sg2>
:G4      cdmf:map      :G5
:G5      rdf:type      cdmf:Graph

-----
// URI du graphe: </graph#G5>
@prefix sg1: </ModelBati#>
@prefix sg7: </ModelBati2#>

sg1:Batiment(?x) -> sg7:Batiment(?x)
sg1:Local(?x) -> sg7:Espace(?x)
sg1:Batiment(?x) ^sg1:Etage(?y) ^sg1:contenir(?x,?y) ^sg1:contenir(?y,?z) ->
  sg7:estComposeDe(?x,?z)
```

Script 7. Exemple d'une mise à jour de modèle de données.

Cette architecture, connectée à un moteur de représentation 3D permet d'afficher une vue 3D (au sens base de données relationnelle) adaptée au contexte d'utilisation et au corps de métier de l'utilisateur (Figure 17).

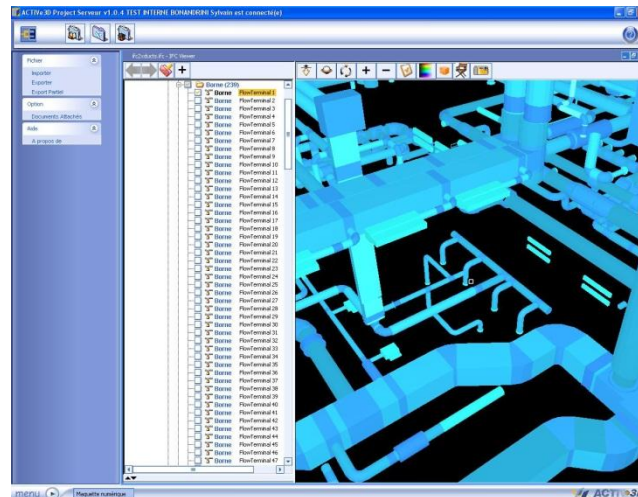


Figure 17. Une scène 3D correspondant au contexte de plomberie

Cette section est une synthèse des travaux de Renaud Vanlande. Elle permet de mettre en avant le fonctionnement du modèle défini et présente, avec quelques exemples, les possibilités offertes. La présentation qui est faite ici du framework est axée sur la gestion de patrimoine immobilier et limitée aux IFC. Il est cependant imaginable d'adapter les exemples pour créer un modèle central de gestion d'objets (graphe de données) qui serait peuplé par d'autres graphes de données eux-mêmes issus de modèles existants comme CityGML par exemple pour l'extension vers le monde extérieur. Ainsi ce modèle créé autour des IFC ne serait qu'une branche parmi d'autres dont le but serait la création d'un métamodèle générique capable de stocker n'importe quel objet de n'importe quelle source, et inversement de transformer n'importe quel objet du modèle générique en un objet correspondant de n'importe quelle branche. De plus, la notion de contexte présente pour chacun des graphes de données permet de séparer très facilement l'information utile aux différents métiers que nous sommes amenés à rencontrer.

5 Active3D 2011-2012 : le projet SIGA3D

Bien que les différentes propositions de Christophe Cruz, puis de Renaud Vanlande aient permis d'atteindre les objectifs fixés en matière de gestion technique de patrimoine immobilier, la gestion des données reste limitée à l'intérieur de l'enveloppe du bâtiment.

Ce projet a été introduit à une période où les problématiques environnementales sont de plus en plus présentes à la fois dans notre quotidien mais surtout dans le milieu de la construction. Celles-ci se font de plus en plus pressantes. De fait, les moyens de contrôler ses coûts dans la gestion et le développement du territoire sont de plus en plus sollicités. Cela peut passer de la gestion d'un patrimoine existant, comme un ensemble de bâtiments par exemple, que l'on souhaite répertorier et observer pour limiter les coûts de maintenance ou pour la création de nouvelles entités afin d'en anticiper les impacts écologiques et économiques, et ce à des échelles différentes. Ce besoin nécessite d'avoir de nombreuses informations sur les biens à gérer, et ce à plusieurs instants de son cycle de vie et à plusieurs échelles : le calcul de l'impact énergétique d'un bâtiment va demander d'avoir des informations précises sur l'intérieur du bâtiment comme le type de cloison, à l'échelle des plans d'architecte, alors que la gestion d'un canal qui s'étend sur plusieurs centaines de kilomètres va demander une échelle de travail beaucoup plus grande, comme le font les Systèmes d'Information Géographiques.

Dans le but d'unifier et de centraliser la gestion des biens immobiliers, urbains et extra-urbains, le projet SIGA3D vise à étendre les travaux développés depuis 10 ans par Active3D pour développer une architecture dynamique, qui dans un même environnement personnalisable et évolutif est capable de gérer tous ces objets (description sémantique) et de les afficher en trois dimensions de manière dynamique et contextuelle depuis une base de données distante. Notre approche pour définir cette architecture se base sur les domaines de la CAO, du BIM (qui touchent de très près la gestion de patrimoine immobilier, en particulier avec les IFC) et des SIG.

Le monde du SIG a longtemps été sujet à une guerre entre les sociétés spécialisées qui ont investi pour développer et imposer un format de données. En effet, l'historique des formats s'accompagne d'avancées sur les plans modélisations et organisationnels, chaque format tentant de répondre aux lacunes des autres formats sans jamais réussir à s'imposer, tout en suivant un cycle connu d'homogénéisation. Il existe de fait de très nombreux formats de fichiers pour représenter l'information géographique, mais un consensus semble être trouvé avec le développement de nouveaux standards, ouverts et indépendants comme GML (Geography Markup Language) qui permet de décrire de manière générique des éléments géographiques et qui sont utilisés pour l'échange de données géographiques sur internet.

Pour répondre au problème de la gestion technique de patrimoine urbain, trois approches sont généralement abordées pour résoudre cette problématique : les éditeurs de CAO vont privilégier l'extension de leur modèle pour intégrer des fonctionnalités SIG. Ainsi, si l'on regarde les IFC par exemple, il est prévu d'étendre le modèle avec de nouvelles classes pour décrire les objets en dehors des bâtiments. Cet aspect est très intéressant, mais n'apporte pas pour autant une description précise de l'environnement comme le font les SIG (le fait de n'avoir que quelques classes pour décrire les objets externes au bâtiment implique que la description sémantique de ceux-ci est plus faible que pour la description du bâtiment en lui-même) et de plus, l'interopérabilité avec les SIG est limitée. La seconde approche commune que l'on trouve est celle qui consiste à étendre les SIG pour permettre un niveau de connaissance plus précis sur le bâtiment. La technique la plus souvent utilisée consiste à transformer une maquette numérique dans un format compatible avec les SIG pour pouvoir l'importer. Le problème est que la transformation est souvent lente, source de perte de données du fait du peu d'éléments communs entre les deux formats utilisés et elle conduit à une duplication des données ce qui pose problème tant au niveau de la taille des données que de la facilité de la maintenance et de la cohérence de ces données. Enfin, la dernière approche consiste à proposer au sein d'une application une lecture directe de différents formats via des API embarquées (Towne, 2009).

Dans le projet SIGA3D, nous avons adopté une approche différente. Dans la même idée que le projet Active3D, nous proposons de centraliser tous les éléments de chaque format utilisé autour d'un formalisme commun afin de produire un format dématérialisé, évolutif et interopérable, ce qui inclut notamment la possibilité d'import et d'export depuis vers n'importe quel format lorsque cela est possible. Ainsi toutes les propriétés de chacun des formats sont conservées. Des liens entre les objets qui représentent une même entité d'après les différents formats peuvent être construits et les données peuvent être contextualisées. Cette dernière partie permet d'associer des informations aux objets comme un nom de propriétaire, une fonction, une date de modification... Cela permet par exemple de filtrer les objets que l'on affiche en 3D en fonction de l'utilisateur et même de charger une représentation de l'objet qui diffère d'un contexte d'utilisation à un autre (on introduit la notion de niveau de détail contextuel, qui est une représentation de l'objet qui ne dépend plus seulement de critères géométriques comme la distance, mais de critères sémantiques). Ce modèle permet également de définir son propre format d'objet géographique dans le but de construire une bibliothèque d'objets personnalisable. Nous proposons pour cela de nous baser sur une ontologie dynamique construite à partir d'un formalisme défini par Renaud Vanlande, C-DMF, que nous allons adapter et qui permet notamment de contextualiser les données d'une ontologie, d'effectuer des opérations sur les graphes sémantiques dans le but

d'alléger au maximum la taille des données à représenter, de simplifier les opérations de maintenance et d'effectuer simplement un suivi des modifications.

En complément du contrat de collaboration signé entre la société Active3D et le laboratoire LE2I pour atteindre ces objectifs, la société a initié un projet EUREKA appelé SIGA3D, en collaboration avec la société Allemande Lat-Lon, éditeur de logiciel dans le domaine des SIG.

Lors de ces trois années de thèses, j'ai eu la responsabilité supplémentaire de gérer le projet européen et de piloter les différents acteurs du projet.

Chapitre 3

État de l'art

Résumé

L'objectif de ce chapitre est de positionner nos travaux de recherche par rapport à l'existant. La gestion de données urbaines nécessite tout d'abord une bonne connaissance des travaux liés au domaine des SIG. Ensuite, nous devons étudier les normes liées à ce domaine spécifique. Active3D a pour ambition de baser son développement sur l'usage de normes, comme elle l'avait fait pour les normes du bâtiment avec les IFC. Une fois que ces deux éléments auront été étudiés, nous devons focaliser notre étude sur les propositions de modélisation sémantique des informations spatio-temporelles. Cette étude nous permettra d'identifier les verrous et les voies d'extension de l'architecture sémantique existante pour prendre en compte la gestion de ces données. Ensuite, nous devons étudier les solutions qui nous permettront de réduire le nombre d'objets urbains qui seront affichés dans le moteur 3D pour améliorer la rapidité et la pertinence du système. Enfin, nous devons étudier les différentes tentatives d'association des contraintes de la modélisation des bâtiments aux contraintes de modélisation d'information géographique et vice-versa.

Cet ensemble d'études nous permettra de disposer d'une bonne vision de l'état de la recherche dans ce domaine et de définir notre approche pour la résolution des problèmes posés dans le projet SIGA3D

Plan

1	Introduction aux Systèmes d'Information Géographique.....	63
1.1	Organisation des données	64
1.2	Gestion des coordonnées	66
1.3	Historique et origine des diversités	67
2	Modélisation Urbaine.....	72
2.1	IFG.....	74
2.2	L'intégration des IFC dans CityGML	75
2.3	Un framework de conversion 3D conçu par l'université technique de Berlin	75
2.4	Framework pour la conversion automatique de BIM (IFC) en CityGML.....	77
2.5	Une ADE CityGML	78
2.6	The Unified Building Model	79
2.7	Limites et conclusion	80
3	La 3D dans les SIG.....	81
3.1	Premiers pas vers la troisième dimension	81
3.2	La topologie en 3D.....	83
4	Normes.....	87
4.1	Présentation des normes géographiques.....	89
4.2	Conclusion	97
5	La multireprésentation dans les SIG.....	98
5.1	La multireprésentation en cartographie.....	98
5.2	La multireprésentation dans les ontologies	100
6	Conclusion	104

De nos jours, on constate un recouvrement important entre le monde du bâtiment (Architecture, ingénierie, construction et gestion de patrimoine, que l’on peut abrégé par AEC - Architecture, Engineering & Construction - et FM – Facility Management -), et celui des Systèmes d’Information Géographique. Les bâtiments sont des installations bâties qui sont positionnées dans un contexte géographique et qui demandent de la connaissance sur leurs relations par rapport au monde (terrain, parcelles, routes...). Les deux domaines partagent le concept du cycle de vie des informations et ont des approches similaires sur la gestion des risques d’un projet, processus de conception, gestion des coûts, gestion d’actifs, maintenance et autres facteurs. La norme IFC présente l’avantage d’homogénéiser les échanges du modèle d’information du bâtiment entre les différentes plateformes de gestion technique de patrimoine. Les IFC vont se focaliser sur le bâtiment et son environnement. Ils permettent de caractériser les objets qui composent un bâtiment, leurs relations, leurs propriétés, et ce pour toutes les phases du cycle de vie du bâtiment et selon différents points de vue (Liebich, 2001). L’environnement du bâtiment est quant à lui limité au site dans la version actuelle des IFC (2x3).

Un site IFC est un espace défini de terre sur lequel le projet est bâti. Il est donc tout à fait possible de dessiner un modèle numérique de terrain en IFC (IfcSite) avec des attributs : périmètre, surface, points du terrain (coordonnées x, y, z), désignation, adresse... et des propriétés associées : description, surface constructible... Cependant, cela reste facultatif et limité au relief. Pour l’instant, ce modèle est peu adapté à la définition d’objets en dehors du bâtiment. Ces limitations sont d’ailleurs bien connues de la communauté IFC comme en témoignent les travaux sur les IFG (IFC for GIS - IFC pour les SIG) qui visent à étendre la norme sur la modélisation d’objet géo-référencés à l’extérieur du bâtiment. Ces travaux concernent dans un premier temps l’intégration des BIM au sein des SIG. Dans la prochaine version 4 des IFC, des éléments géographiques pourront être définis (`IfcGeographicElement` et `IfcGeographicElementType`) dans le but d’augmenter l’interopérabilité avec les SIG. Cependant, même si l’initiative est intéressante, le but n’est pas d’inclure un système d’information géographique dans les IFC. En effet, il n’est, par exemple et pour l’instant, pas prévu d’outils d’analyse spatiale. Le but est d’utiliser les IFC comme moyen d’échange entre les SIG et les systèmes d’AEC/FM pour une information limitée, mais significative. De plus, les évolutions restent concentrées sur l’environnement immédiat du bâtiment et n’implémentent pas de classes spécifiques pour les objets urbains comme on peut en trouver pour le bâtiment, ce qui présente une nouvelle limite. On peut en conclure que les IFC n’ont pas vocation à décrire des environnements urbains complexes (c’est-à-dire un ensemble de bâtiments, d’objets urbain ou de réseau), les logiciels spécialisés dans le SIG le faisant très bien. Son rôle est réellement centré sur la normalisation et l’échange des informations, à la fois dans son domaine d’origine, la construction, mais aussi dans des domaines étendus comme celui des SIG.

Pour illustrer les faiblesses des IFC en termes de gestion extra-immobilière, nous allons prendre l’exemple du Conseil Régional de Bourgogne. Aujourd’hui, la région gère une centaine de lycées au moyen des IFC. Demain ils devront gérer l’ensemble des canaux présents en Bourgogne (plusieurs centaines de kilomètres de canaux). Pour une gestion efficace de ceux-ci, il semble nécessaire de pouvoir représenter (géométriquement, mais également sémantiquement) l’ensemble des biefs, des écluses, des maisons, des arbres, des bancs et autres éléments qui les composent et de pouvoir effectuer des requêtes spatiales sur ces éléments (pour récupérer par exemple toutes les écluses positionnées entre deux maisons données afin d’assigner les ressources suffisantes pour leur gestion). Les IFC trouvent très vite leur limite sur ce genre de projets (précision dans la localisation des objets sur site, mise en relation des différents objets, informations sur le paysage extérieur, requêtes spatiales...). Il faut alors coupler cette approche IFC avec une approche SIG. Le sujet est d’autant plus d’actualité que de nombreux appels d’offres regroupent des lots concernant la gestion de patrimoine immobilier avec ceux concernant la gestion cartographique afin d’avoir toutes les informations accessibles de manière homogène (exemple des aéroports de Paris en 2010^{33,34}). Aujourd’hui, dans ce type de projet, certaines

³³ <http://pro.01net.com/editorial/518527/comment-aeroports-de-paris-gere-son-patrimoine-industriel/>

administrations prennent le parti d'utiliser des logiciels spécialisés pour des opérations bien précises (Ferries et Légglise, 2006), ce qui évite de surcharger inutilement les applications non concernées avec des données métiers, ou bien d'avoir des applications généralistes, mais vides de connaissances métiers spécifiques

De nombreux travaux existent depuis les années 90 pour créer des modèles urbains liés aux SIG. D'après les définitions tirées du livre de Michael Batty (Batty et Smith, 2009), on peut qualifier les modèles urbains comme des représentations graphiques de fonctions et processus permettant de générer des structures spatiales urbaines en termes d'utilisation du terrain, de population, d'emploi et de transports. C'est ce que l'on trouve habituellement dans les programmes informatiques qui permettent de tester des théories de positionnement par rapport aux données et de générer des prédictions de modèles d'implantation à venir. La modélisation urbaine, quant à elle, est un processus de sélection de théories et de leur transformation en modèles mathématiques ou formels, de développements d'applications dédiées et enfin de confrontations du modèle avec des données afin de le calibrer, valider et vérifier avant de l'utiliser pour faire des prédictions.

Auparavant, les techniques de modélisation urbaines et les SIG étaient développées en parallèle avec peu d'interactions entre les deux domaines. À partir de la fin des années 80, afin d'améliorer les capacités d'analyses des SIG, des efforts ont été faits pour intégrer la modélisation urbaine. La multiplication des travaux dans les années 90 témoigne de la reconnaissance de tels bénéfices mutuels.

Les définitions des Modèles d'Information Urbain sont variées et il n'y a pas de définition bien établie comme pour les BIM. En particulier (Coors, 2003) définit le Urban Data Model (UDM) comme un modèle de données urbain qui représente la géométrie d'une surface ou d'un volume par un ensemble de faces planes et convexes. On voit donc ici l'importance de la géométrie dans la modélisation urbaine, mais il n'est pas fait état de l'information sémantique liée au modèle.

On définit ainsi le Modèle d'Information Urbain (UIM pour Urban Information Model, par analogie au Building Information Model) comme une extension des BIM (Hamilton et al., 2005). Ces travaux soulignent les difficultés liées à la mise en place de tels systèmes, notamment sur les problèmes d'hétérogénéité ou de la gestion de l'espace et du temps. De nombreuses thèses de doctorat ont été faites ces dernières années dans ce domaine, en particulier à l'IGN³⁴ (Institut national de l'Information Géographique et forestière) (Koussa, 2011) (De La Losa, 2000) (Ramos, 2003). D'autres travaux de thèses dont l'objectif était similaire ont cherché à convertir des modèles de bâtiments provenant des systèmes de CAO tout en gardant des éléments de sémantique que l'on pouvait y trouver, par exemple dans (Zlatanova, 2000). Ces travaux ont débouché plus tard sur la recherche d'intégration de fichiers IFC au sein de CityGML.

Arrêtons-nous d'ailleurs un instant sur CityGML. Initialement développé en Europe, le format CityGML s'est peu à peu imposé comme la référence pour l'échange de modèles 3D de villes numériques. Depuis Août 2008, CityGML est un standard de l'OGC. CityGML permet de décrire la représentation, le stockage et l'échange de modèles 3D de villes numériques et de modèles de terrains. CityGML est implémenté comme un schéma d'application de GML 3.1.1. CityGML fournit donc un modèle normalisé et des mécanismes pour décrire les objets 3D avec leur géométrie, leur topologie, leur sémantique et leur apparence. Un des gros avantages de CityGML est qu'il définit cinq niveaux de détail différents, ce qui lui permet d'être hautement dimensionnable et de représenter ainsi à la fois des objets urbains, mais aussi des quartiers, villes, régions ou pays. Nous reviendrons sur des aspects spécifiques de CityGML tout au long de ce chapitre.

Pour finir la présentation de la modélisation urbaine, nous allons introduire la notion de « city modeling », ou modélisation des villes. Ce terme possède une connotation quelque peu différente de l'urban information

³⁴ <http://www.autodesk.fr/adsk/servlet/pc/item?siteID=458335&id=15946853>

³⁵ <http://www.ign.fr/>

modeling. D’après la littérature, on remarque que le city modeling est considérée comme une partie de la modélisation urbaine (Vosselman, 2003)(Herold et al., 2003). Il est souvent associé à une notion de modélisation 3D et objet des villes (Sui, 1998). On peut d’ailleurs se demander si les BIM ne peuvent pas être considérés comme un sous ensemble du city Modeling, même si l’approche, au final, est différente. On peut alors, dans ce cas particulier, adapter et justifier les techniques de couplage du SIG et de l’urban modeling.

Pour mieux appréhender les différents domaines concernés par ce projet de recherche, nous allons tout d’abord étudier les approches qui convergent vers une modélisation urbaine. Cette convergence se réalise soit par extension des IFC vers des modélisations spatio-temporelles, soit des systèmes d’information géographique vers des modélisations plus sémantiques et précises du bâtiment. Dans la suite de ce document, nous commencerons par présenter le fonctionnement général des SIG et des normes associées. Nous porterons notre analyse à la fois sur le niveau de la modélisation sémantique et sur le niveau de la modélisation géométrique des caractéristiques d’un SIG. Puis nous ferons un bref état de l’art des propositions et les limites des approches de modélisation urbaine. Ceci nous amènera naturellement à aborder la notion de multireprésentation des données dans ces systèmes. L’objectif final est d’identifier les éléments qui devront être pris en compte dans notre approche pour atteindre nos objectifs.

1 Introduction aux Systèmes d’Information Géographique

Il existe de nombreuses définitions d’un Système d’Information Géographique et chacun peut se faire sa propre idée, spécialiste du domaine ou non. Jusqu’à aujourd’hui la 3D au sein des SIG avait essentiellement deux finalités : la communication (visite virtuelle d’une ville depuis le web, visualisation d’implantation de nouveaux bâtiments... C’est typiquement l’utilisation faite de la maquette numérique de la ville de Cannes³⁶ par exemple) et la simulation (propagation d’ondes pour l’acoustique ou l’électromagnétisme, voire les travaux du CSTB dans ce domaine³⁷. Avec l’avènement des solutions mobiles, une nouvelle utilisation est apparue sous la forme de réalité virtuelle ou réalité augmentée (Kim et al., 2009). Un SIG peut être présenté de différentes façons selon ce que l’on veut mettre en avant : l’aspect données (cartes par exemple) ou l’aspect système et traitement. D’après (Clarke, 1997), on peut dire qu’un SIG est un système permettant, depuis diverses sources, de saisir, éditer, stocker, analyser, partager, gérer et présenter des données numériques qui font référence ou qui sont liées à un emplacement (données géographiques) : toute donnée référencée à une localisation terrestre peut être considérée comme information géographique.

Un SIG permet de garder des traces de ce qui existe, d’évènements ou d’activités, à l’endroit où ça existe ou se produit. Il peut répondre à des questions (où ?, quoi ?, comment ?, quand ?, et si ?) et résoudre des problèmes (aménagement, gestion) en analysant les données (détermination de lien entre les données, de schémas particuliers, dégagement de tendances...) puis présenter des résultats compréhensibles et partageables rapidement (cartes, rapports, graphes...). Qu’il s’agisse de traitements opérationnels, tactiques ou stratégiques, à plus ou moins long terme, la notion d’évolution des données au cours du temps est un élément essentiel de ces systèmes (Longley, 2005). La possibilité de mettre à jour les données à chaque fois que de nouvelles informations arrivent est un avantage indéniable sur les cartes qui restent figées une fois imprimées.

Les SIG ont des domaines d’application aussi variés que la recherche scientifique, la gestion de ressources ou d’immeubles, l’archéologie, l’évaluation des impacts environnementaux, la planification urbaine, la cartographie, la criminologie, la géographie historique (numérisation et géoréférencement de cartes historiques qui peuvent contenir des informations importantes sur le passé), le marketing, la logistique, la cartographie de prospection de minerais et bien d’autres encore (Zaninetti, 2005). Il en découle que les

³⁶ <http://3d.cannes.fr/>

³⁷ <http://www.cstb.fr/>

fonctionnalités des SIG sont toutes aussi variées : traitements d'images, photo-interprétation (télédétection, photo aérienne...), traitements statistiques (plus particulièrement géostatistique), utilisation de GPS et DGPS (GPS différentiel), utilisation de mini-SIG embarqués, mise à disposition de cartes sur internet.

On remarque que la notion de SIG est très vague et peut être utilisée dans de nombreuses circonstances. Il existe une multitude de SIG, mais le principe de fonctionnement de ces systèmes est pratiquement toujours le même : **acquisition** des données, **stockage** en bases de données de celles-ci, **traitement** des informations et **présentation** des données et des résultats selon l'utilisation. On associe souvent le SIG à la carte, mais même si celle-ci est un bon moyen de synthétiser et présenter des résultats, ce n'est qu'une des façons de travailler avec des données géographiques et seulement un type de produit généré par le SIG. On trouve par exemple des systèmes dont le rôle est de regrouper des données géographiques provenant de différentes sources à travers le web afin de créer une nouvelle application hybride (c'est le concept de mashup, comme par exemple sur le site de « the OpenStreetMap Cycle Map »³⁸ qui utilise les données libres de Open Street Map pour créer une nouvelle application).

Les informations manipulées par les SIG sont souvent hétérogènes et peuvent être présentées sous différentes formes. Nous allons étudier, dans une première partie l'aspect données du SIG : quelles sont-elles, comment les représente-t-on et que représentent-elles. Nous nous pencherons ensuite dans une seconde partie sur la problématique de la localisation des objets dans l'espace, et plus spécialement sur la Terre. En effet, celle-ci ayant une forme particulière (on parle de *géoïde*), il est relativement difficile de localiser un point sur une représentation plane de la Terre (carte). Et malgré le fait que le stockage numérique des données géographiques permette de s'affranchir de cette limitation de planéité, la gestion des cartes reste l'une des fonctionnalités principales d'un SIG. Aussi, nous étudierons les techniques qui existent afin de pallier à ce problème. Enfin, nous aborderons dans une troisième partie les possibilités offertes par la 3D au sein des SIG, en présentant dans un premier temps ce qui se fait beaucoup aujourd'hui avec des modèles de 3D simplifiés : la 2,5D (où l'on ajoute un paramètre d'altitude à un point).

Dans la suite de cette section, nous présenterons les différents composants d'un SIG et surtout les éléments de divergence avec le monde du BIM.

1.1 Organisation des données

Nous allons commencer par voir comment sont conçus les modèles de SIG 2D avant d'aborder la 3D. Les bases de données géographiques en 2D se classent en deux grandes catégories, les données de type raster, et les données vecteur (SCHOLL et al., 1996).

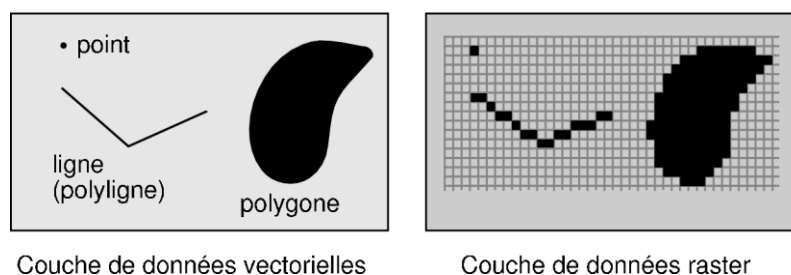


Figure 18. Les types de représentations graphiques des données géographiques

³⁸ <http://www.opencyclemap.org>

La Figure 18 présente les trois entités géométriques de base utilisées pour les représentations vectorielles (point, polygone et surface), sous forme vectorielle d’un côté, et son équivalent sous forme raster (ou maillée) de l’autre. Le type de données maillées est beaucoup mieux adapté aux applications liées à l’affichage sur écran d’ordinateur qu’à des applications d’analyse spatiale. Dans ce dernier cas, c’est le format vecteur qui sera préféré. Souvent, au sein d’un SIG, on utilise une entité pour un type d’objet particulier, mais cela n’empêche pas de composer des objets complexes à partir de plusieurs primitives. C’est ainsi que la Figure 19 montre un exemple d’utilisation de ces primitives : le point représente des lampadaires (points noirs), les polygones montrent les éléments bâtis (groupement d’immeubles et immeubles) alors que le réseau d’eau est composé à la fois de lignes (tuyaux) et de points (pour les points d’accès au réseau).

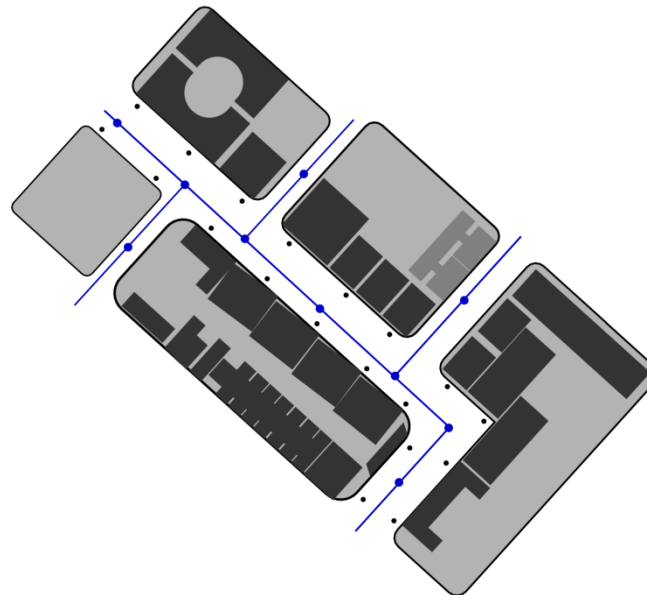


Figure 19. Carte d’un quartier de ville en représentation vectorielle

Il existe plusieurs modèles gérés par les SIG, du plus sommaire, le modèle dit spaghetti anarchique qui ne gère que des polygones, à des modèles plus complexes intégrant des dimensions supplémentaires de type topologique, directionnel, sémantique... (De La Losa, 2000). La plupart des SIG 3D réalisés durant la dernière décennie utilisent des modèles topologiques 3D (De La Losa, 2000) (Ramos, 2003) (Poupeau, 2008)(Gesbert, 2005), c’est-à-dire qu’en plus de décrire les géométries, ils décrivent les relations entre celles-ci. La Figure 20 illustre quelques une de ces relations.

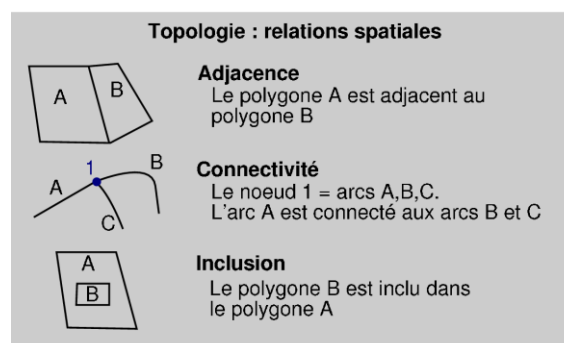


Figure 20. Description des objets géographiques au niveau sémantique et géographique.

La topologie revêt une certaine importance au sein des SIG, notamment afin de contrôler la cohérence des données. Nous allons faire un point rapide sur ce qu’est exactement la topologie. La topologie est la branche

des mathématiques qui s'intéresse aux propriétés des objets invariants par homéomorphisme (bijection continue dont la réciproque est continue), ou plus simplement par déformation, rotation, translation, étirement (sans déchirement). Lorsqu'en géomatique on parle de topologie, on se réfère davantage aux concepts véhiculés par la théorie des graphes. Un graphe peut être défini comme une relation binaire dans un ensemble. Un graphe est la donnée d'un couple (N, A) où N est l'ensemble des éléments (Nœuds) et A l'ensemble des couples de nœuds en relation (Arcs). Dans le contexte géomatique où les graphes portent sur des éléments géométriques, il faut être prudent à ne pas confondre le concept de graphe avec le dessin qui le symbolise (Langlois, 1994).

La modélisation topologique apporte de nombreux avantages dans les représentations géométriques des SIG, aussi bien en 2D qu'en 3D :

- économie de temps de calculs : certains calculs géométriques d'inclusion, d'adjacence, de frontière ou de parcours de réseau sont extrêmement gourmands en ressource système et l'ISO conseille de s'appuyer sur des structures combinatoires des complexes topologiques (ISO 19107),
- cohérence des données décrites : permet de connecter des primitives géométriques indépendantes pour éviter les incohérences en cas d'approximation, d'incertitude ou de modification des objets (comme illustré dans la **Figure 21**)
- et description de la sémantique spatiale : basée sur les travaux de (Zeitouni, 1995), la façon dont les objets sont connectés dans un espace donné constitue un modèle naturel, plus explicite qu'un modèle géométrique pour les utilisateurs de SIG ; on parle de topologie sémantique.

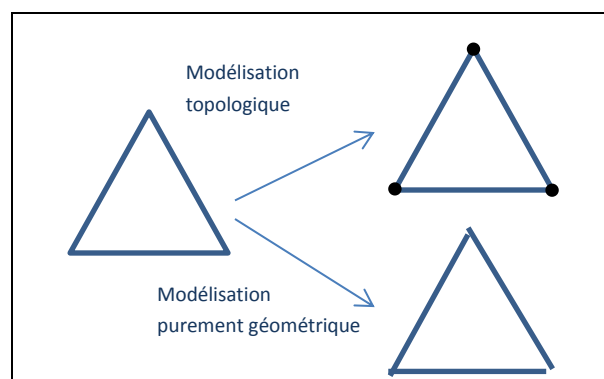


Figure 21. La topologie pour une meilleure cohérence de données

1.2 Gestion des coordonnées

L'un des aspects importants de notre étude est la gestion des coordonnées et des systèmes de référence géodésique. C'est en effet un point qui est très peu abordé dans la modélisation des bâtiments, même si des mécanismes pour les gérer existent. La géodésie est une science antique (IIe siècle av. J.-C.) qui vise à étudier la forme et les dimensions de la Terre et de son champ de pesanteur. Elle est omniprésente aujourd'hui et nécessaire au fonctionnement des SIG. Elle a pour but, entre autres, de déterminer les coordonnées de points de repère dans des systèmes de référence géodésiques et de représenter tout ou partie de la surface de la Terre dans différentes projections planes (IGN, 2008a). Nous allons donc étudier cet aspect des SIG en plusieurs points. Tout d'abord nous allons définir un peu plus précisément ce que sont les systèmes géodésiques, projections, systèmes de coordonnées et autres termes utilisés en géodésie moderne. Ensuite, nous nous pencherons sur l'étude de systèmes géodésiques particuliers (WGS84, RGF93) et des projections utilisés pour la France (Lambert I, II, II étendue, III, 93) ainsi que les différents liens qui les unissent. Enfin, nous ferons un petit comparatif des solutions qui existent pour travailler avec ces coordonnées et passer d'un système à un autre.

1.3 Historique et origine des diversités

La géolocalisation des données est ce qui va distinguer l’information géographique des autres types de données : elle est indispensable au sein des SIG. Ce n’est cependant pas un concept nouveau puisque depuis l’antiquité la représentation de la Terre et la localisation de points sur celle-ci est une problématique qui occupe les hommes. On sait depuis cette époque que la Terre est ronde et c’est en 240 av J.-C. qu’Eratosthène détermine le premier rayon terrestre en mesurant l’angle entre les verticales d’Assouan et d’Alexandrie et la distance qui les sépare (IGN, 08b). Au XVII^e siècle apparaît la géodésie moderne et les progrès dans ce domaine permettent de déterminer que la Terre est aplatie au pôle, théorie qui ne sera définitivement validée qu’au XVIII^e siècle. C’est d’ailleurs à la fin de ce siècle que le mètre est choisi comme unité de mesure de longueur par l’Académie des Sciences comme étant le quart de la dix-millionième partie du méridien terrestre.

Au début du XIX^e siècle il n’existe pas encore de vision vraiment globale de la planète. Aussi voit-on apparaître de nombreux systèmes de référencement dits « locaux », souvent à l’échelle d’un pays. La prise de conscience d’un manque de collaboration entre les pays mène à la création de l’Association Internationale de Géodésie en 1886. Puis ce n’est qu’au XX^e siècle, avec des mesures électromagnétiques et la puissance de calcul de l’informatique, que l’on va affiner notre connaissance de la forme de la Terre : le géoïde. Parallèlement, la géodésie spatiale a établi des réseaux qui ceignent la Terre et donnent un positionnement absolu et relatif en coordonnées géocentriques.

1.3.1 Les systèmes de référence géodésiques

Un système de référence géodésique (CRS pour Coordinates Reference System) est un ensemble de conventions qui permettent de définir de façon univoque le positionnement des points dans l’espace et le temps au voisinage de la Terre. De nombreux systèmes coexistent, en raison de dispositions légales, réglementaires ou historiques, de l’amélioration des techniques et des modèles et de l’élargissement de la zone d’application des techniques à la Terre entière. La définition d’un tel système passe par deux étapes : la définition d’un référentiel géodésique (système inaccessible et invariable) et la matérialisation de celui-ci par des repères terrestres (accessibles et perfectibles, le réseau géodésique) (Altamimi, 2006) (IGN, 2008a). La forme de la Terre est très particulière (la meilleure représentation est appelée le géoïde, visible dans la **Figure 22**) et les calculs sur une telle surface sont très complexes. La surface réelle est alors simplifiée à l’aide d’une surface abstraite et mathématiquement simple : l’ellipsoïde de révolution. Cependant, la définition d’un ellipsoïde de référence pose quelques problèmes qui pourraient conduire à créer un nombre infini d’ellipsoïdes : définir le centre O de la Terre (relativement difficile au vu de sa forme), et déterminer les dimensions de l’ellipsoïde (celui-ci étant caractérisé par la dimension de son demi-grand axe a et son coefficient d’aplatissement f). Il en découle qu’au fil des siècles (surtout à partir du XIX^e) de nombreux ellipsoïdes (des centaines) ont été proposés. On peut les classer en deux catégories : ceux qui donnent une approximation de la surface terrestre dans son ensemble (les ellipsoïdes globaux) et ceux qui proposent une représentation plus précise d’une portion de la surface (ellipsoïdes locaux). La **Figure 22** illustre ces deux cas de figure. Les «reliefs» du géoïde sont exagérés, ceux-ci étant très faibles par rapport aux dimensions de la Terre (-120 mètres à +80 mètres, c’est pour cela que depuis l’espace la Terre nous apparaît comme sphérique).

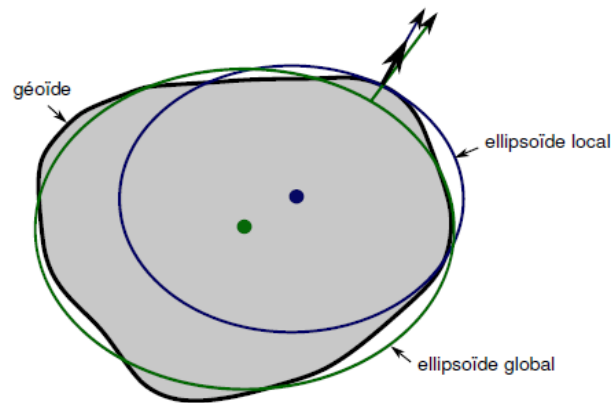


Figure 22. Approximation du géoïde par deux ellipsoïdes (global et local)

D'après le sujet 2 des spécifications abstraites de l'OGC³⁹, les sous-types principaux suivants de CRS sont distingués :

- **Géocentrique** : le système traite la courbure de la Terre en prenant un point de vue spatial en 3D. Il n'y a donc théoriquement pas d'approximation géométrique de la surface de la Terre à effectuer. L'origine d'un modèle géocentrique CRS est proche du centre de la masse de la Terre.
- **Géographique** : ce type de CRS est basé sur une approximation ellipsoïdale du géoïde, ce qui fournit une représentation exacte de la géométrie pour une grande partie de la surface de la Terre. Les CRS géographiques peuvent être 2D ou 3D.
- **Projeté** : type de système de coordonnées de référence qui est basée sur une approximation de la forme de la surface de la Terre par un plan. Il en découle inévitablement des distorsions, un ellipsoïde n'étant pas développable. Cependant, la distorsion inhérente à cette approximation est soigneusement contrôlée et connue, ce qui permet d'appliquer des corrections pour obtenir un résultat proche des valeurs réelles sur le terrain.
- **Ingénierie** : ce type de système est utilisé dans un contexte local. Ce sous-type permet de modéliser deux grandes catégories de systèmes de coordonnées de référence locaux :
 - les systèmes fixes, appliquées aux activités d'ingénierie sur ou près de la surface de la Terre ;
 - les coordonnées sur plates-formes mobiles telles que les voitures, navires, avions ou autres engins spatiaux.

Les systèmes d'ingénierie fixes sont généralement basés sur une représentation plane de la Terre, et les effets de la courbure de la Terre sur les entités géométriques sont ignorés ; on utilise dans ce cas l'arithmétique simple dans le plan, sans corriger les éventuelles déformations dues à la forme de la Terre. Ils sont utilisés sur des zones relativement petites (quelques centaines de mètres au plus) et le « contexte local » est dans ce cas équivalent à « l'espace local ».

- **Vertical** : ce type de CRS est utilisé pour le stockage des hauteurs et des profondeurs.
- **Les CRS verticaux** utilisent la direction de la gravité pour définir les concepts de hauteur et de profondeur. La hauteur de l'ellipsoïde n'étant pas constante, celle-ci ne peut pas exister de façon indépendante, mais seulement comme partie intégrante d'un tuple de coordonnées 3D défini dans un CRS géographique 3D.

³⁹ Les spécifications peuvent être trouvées à l'adresse <http://www.opengeospatial.org/standards/as>

Le « système de coordonnées monde » que l’on trouve dans les plans de bâtiment de CAO est un exemple d’un système de coordonnées de référence d’ingénierie fixe. En outre, il existe également des systèmes de coordonnées d’image (CRS d’ingénierie appliqué aux emplacements dans une image) et de temps.

1.3.2 Les systèmes de coordonnées

Pour se localiser sur la Terre, il est nécessaire d’utiliser un système géodésique duquel découlent les coordonnées géographiques figurant sur les cartes (IGN, 2008c). Chaque système de coordonnées de référence utilise un système de coordonnées. Les CRS composés peuvent utiliser différents systèmes de coordonnées. Les coordonnées des points sont stockées dans un système de coordonnées. Un système de coordonnées est l’ensemble des axes du système de coordonnées qui couvre l’espace des coordonnées. Un système de coordonnées peut avoir un ou plusieurs axes qui sont généralement :

- **1D** - Par exemple, un système de coordonnées linéaire ou vertical qui mesure la hauteur le long d’un axe unique.
- **2D** - Par exemple, le système de coordonnées d’une carte en 2D ou d’un plan de construction 2D
- **3D** - Par exemple, le système de coordonnées d’un modèle de terrain 3D ou d’un modèle de construction
- **4D** - Par exemple, ajouter l’axe temporel à un modèle 3D

Les types suivants (dans le Tableau 1) sont distingués dans le thème 2 des spécifications abstraites de l’openGIS :

Sous-type de système de coordonnées	Description	Utilisé avec le(s) type(s) de CRS
Cartésien	Système de coordonnées de dimension 1, 2 ou 3. Il donne la position des points par rapport à des axes orthogonaux suivant des droites, en 2D et 3D. Pour la 1D, il contient un unique axe de coordonnées suivant une droite. Dans le cas multidimensionnel, tous les axes ont la même unité de mesure.	Géocentrique Projeté Ingénierie Image
Cartésien oblique	Système de coordonnées 2D ou 3D d’axes rectilignes qui ne sont pas nécessairement orthogonaux.	Ingénierie Image
Ellipsoïdal	Système de coordonnées 2D ou 3D dans lequel la position est spécifiée par une latitude géodésique, une longitude géodésique et (dans le cas tridimensionnel) une hauteur ellipsoïdale, associé à un ou plusieurs systèmes de coordonnées de référence géographiques.	Géographique Ingénierie
Sphérique	Système de coordonnées tridimensionnel avec une distance à l’origine, et deux coordonnées angulaires. À ne pas confondre avec le système de coordonnées ellipsoïdales (la sphère est un ellipsoïde dégénéré). Ce système est illustré dans la Figure 23 qui montre un exemple de coordonnées tridimensionnelles géographiques (λ , ϕ , h).	Géocentrique Ingénierie
Cylindrique	Système de coordonnées en 3 dimensions constitué d’un système de coordonnées polaires et d’un axe de coordonnées perpendiculaire au plan défini par le système de coordonnées polaires.	Ingénierie
Polaire	Système de coordonnées en 2 dimensions dans lequel la position est donnée par la distance du point à l’origine (point central appelé pôle) et l’angle entre le point et la demi-droite d’origine (angle 0°) dans une direction de référence.	Ingénierie

Vertical	Système de coordonnées à une dimension utilisé pour enregistrer les hauteurs (ou profondeurs) de points sur le champ gravitationnel de la Terre.	Temporel
Linéaire	Système de coordonnées à une dimension qui se compose des points situés sur un axe unique. L'ordonnée associée est la distance de l'origine spécifiée au point le long de l'axe (par exemple on peut utiliser une ligne représentant une route pour décrire les points sur ou le long de cette route.	
Temporel	Système de coordonnées à une dimension contenant un axe de temps unique et utilisé pour décrire la position temporelle d'un point dans les unités de temps spécifiées à partir d'une origine de temps définie.	

Tableau 1. Les différents types de Systèmes de Coordonnées

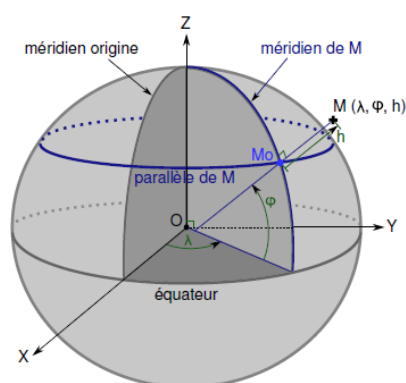


Figure 23. Système de coordonnées sphérique

De façon générale, les projets de constructions tels que traités dans le BIM utilisent un système de coordonnées Cartésien à l'intérieur d'un CRS d'ingénierie, mais sans datum (système de référence lié à la Terre, cf. section 1.3.4).

1.3.3 Le repère

Les systèmes de coordonnées sont constitués d'un ensemble ordonné d'axes ou repère. Un axe peut être classé par la façon dont il est défini, par exemple la distance par rapport à une origine (obtenue par une mesure de longueur), l'angle planaire depuis une direction de référence ou le long d'un axe de temps. Le tableau suivant (Tableau 2) répertorie les axes utilisés en fonction du système de coordonnées et du CRS associé.

Système de coordonnées	Système de coordonnées de référence	Nom des axes des systèmes de coordonnées
Cartésien	Géocentrique	X géocentrique, Y géocentrique, Z géocentrique
Sphérique	Géocentrique	Latitude sphérique (on parle aussi de latitude géocentrique), longitude sphérique, rayon géocentrique
Ellipsoïdal	Géocentrique	Latitude géodésique, longitude géodésique, hauteur ellipsoïdale
Vertical	Vertical	Hauteur du champ gravitationnel terrestre
Vertical	Vertical	Profondeur

Cartésien	Projeté	Est, Nord
Cartésien	Projeté	Ouest, Sud

Tableau 2. Aperçu des axes utilisés en fonction du système de coordonnées

Les projets de construction sont généralement conçus dans un système de coordonnées cartésien avec une définition des axes dite main droite⁴⁰, X, Y, Z. Il n’y a pas de datum, tous les axes se réfèrent à un (arbitraire) du lieu d’origine (0, 0, 0). Le seul système de coordonnées cartésiennes sous-jacente d’un modèle CAO utilisé pour les plans de construction est aussi appelé "World Coordinates System", ou WCS. Souvent, un modèle CAO d’un plan de construction a de nombreux systèmes de coordonnées locales, et tous sont convertis dans le "système de coordonnées général" unique par une matrice de transformation (3x3 pour la 2D, 4x4 pour la 3D). Ce système de coordonnées locales est appelé « Système de Coordonnées Utilisateur » (UCS pour User Coordinates System, SCU en français). Ces systèmes sont tous cartésiens, main droite, 2D ou 3D.

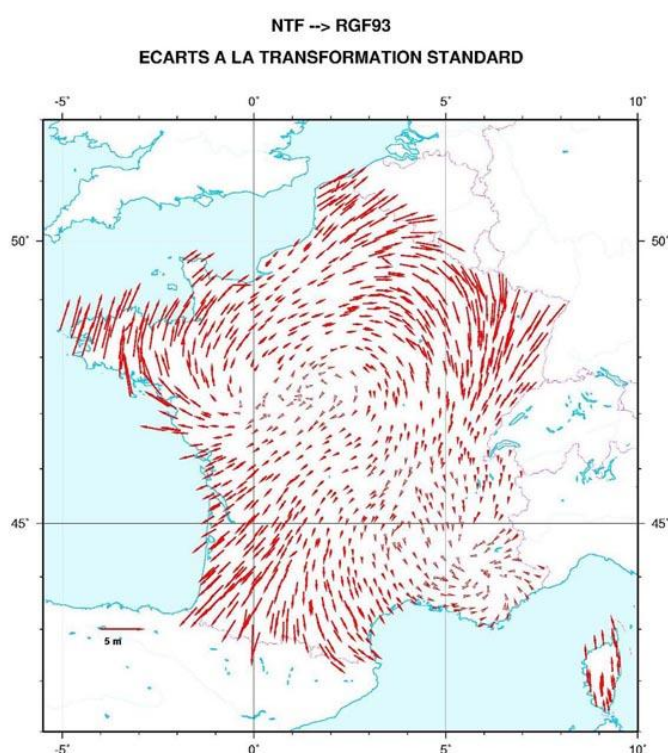


Figure 24. Écarts à la transformation standard sur les points en colocalisation NTF/RGF93

1.3.4 Définition du datum

Un datum (ou système de référence) définit la relation d’un système de coordonnées à la Terre (position de l’origine, échelle, orientation). Il est nécessaire pour créer un système de coordonnées de référence. Dans le cas d’un système de coordonnées d’ingénierie, il peut aussi être défini localement et non pas en relation à la Terre. Il y a différents types de datum, chacun ne pouvant être associé qu’à un type particulier de CRS : datum géodésique, vertical, d’image, d’ingénierie et temporel.

⁴⁰ Main gauche, main droite : Les mains sont utilisées comme moyen mnémotechnique pour se rappeler de la direction des axes : avec les doigts de la main pliés à 90° (pouce, index et majeur), on obtient respectivement les axes x, y et la direction positive de l’axe z.

Pour information, depuis décembre 2000, le système de référence légal est le RGF93 en France Métropolitaine avec comme projections (systèmes projetés) associées Lambert-93 et Coniques Conformés 9 zones⁴¹. Cependant, encore aujourd'hui, en France, de nombreuses administrations et projets utilisent encore l'ancien référentiel NTF (Nouvelle Triangulation de la France) avec les projections associées dites Lambert 4 zones (définie en 1920). La Figure 24 ci-dessus, tirée du site de l'IGN montre les écarts entre ces deux systèmes et la nécessité de calculer une grille de paramètres pour passer d'un système à l'autre. L'IGN a calculé cette grille, au pas de 0,1°, et l'a incluse dans son logiciel Circé⁴² qui convertit des coordonnées géographiques ou cartographiques d'un système de coordonnées dans un autre.

Dans certains cas, pour désigner les systèmes de coordonnées nous utiliserons les codes EPSG (European Petroleum Survey Group). Ceux-ci constituent une liste des systèmes de coordonnées géo-référencées de projection utilisés tout d'abord par le groupe de producteurs de pétrole du même nom avant d'être utilisés par de nombreux logiciels de SIG. Ces codes sont notamment utilisés dans les standards de l'OGC.

2 Modélisation Urbaine

La modélisation urbaine est à la jonction de deux mondes: celui de l'architecte qui modélise un nouveau projet de bâtiment pour l'insérer dans un site et le construire et d'autre part la personne chargée de modéliser tous les éléments d'un site urbain, avec notamment les bâtiments (Ferries et Légise, 2006).

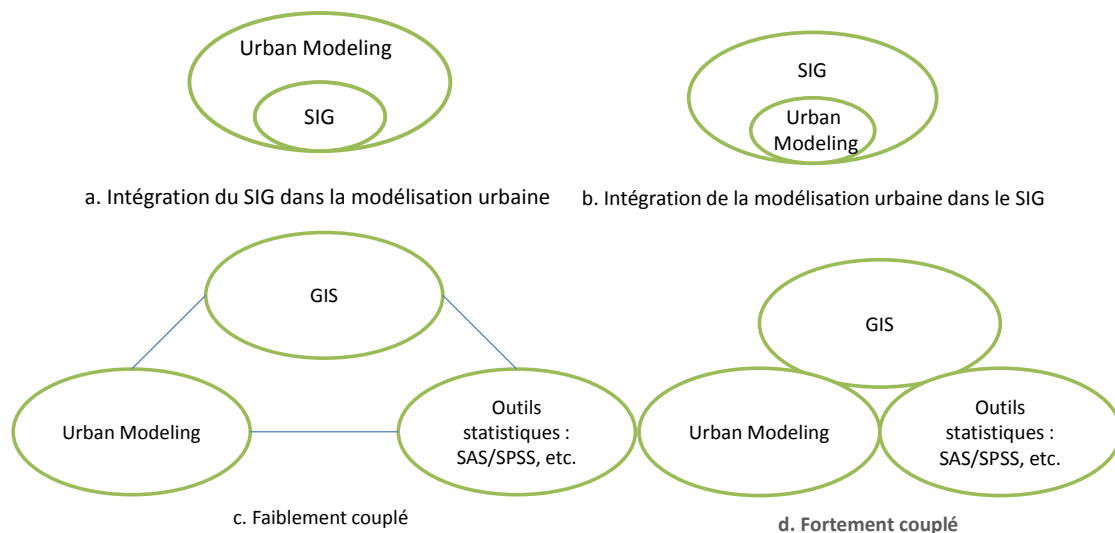


Figure 25. Approche de la modélisation urbaine par les SIG.

Durant ces dernières années, les administrations publiques, les villes et les entreprises ont montré un grand intérêt dans la construction de modèles virtuels de villes en 3D, et ce pour différents usages allant de la communication à la gestion d'un patrimoine urbain en passant par des projets d'urbanismes, d'implantation ou de simulations (bruit, pollution...) (Stoter et al., 2008). Certains projets sont axés essentiellement sur la modélisation d'objets spatiaux et géographiques représentés dans un environnement urbain, d'autres sur la visualisation d'objets sans pour autant représenter leurs données attributaires ou les relations entre eux. En effet, non seulement chaque projet de modélisation urbaine est conçu selon les différents moyens et connaissances du concepteur, mais aussi en fonction des différents objectifs. Par exemple, un architecte souhaite avoir une vision précise des bâtiments avec ses détails de façade et du terrain proche, mais une personne qui souhaite effectuer des simulations sur la propagation du bruit souhaite un modèle dont le niveau

⁴¹ Décret n°2000-1276 du 26 décembre 2000

⁴² <http://geodesie.ign.fr/index.php?page=circe>

de détail ne laisse pas apparaître les éléments dont la taille est inférieure à 50 cm puisque cela va complexifier de manière exponentielle les temps de calcul alors que les impacts de ces détails sont négligeables sur les résultats.

Il est possible par ailleurs de regrouper les techniques de rapprochement entre les SIG et les modèles urbains en quatre groupes (Sui, 1998). La Figure 25 illustre ces catégories qui sont l’intégration des SIG dans les modèles urbains (Figure 25a), l’inverse, c’est-à-dire l’intégration des modèles urbains au sein des SIG (Figure 25b), puis deux formes de couplage : faible, l’intégration des deux systèmes par l’intermédiaire d’échange de données (Figure 25c), et le couplage fort (intégration de certains modèles et fonctionnalités d’un système à l’autre, par le biais de macro par exemple)(Figure 25d).

De fait, malgré le nombre important de modèles de ville existants, il existe des problèmes communs quant à l’intégration de ces différents modèles (Mekawy et Östman, 2010) :

- des organisations différentes ;
- des schémas différents ;
- des modèles géométriques différents ;
- un manque de sémantique ;
- un manque d’interopérabilité.

Afin de pallier à ces différents problèmes, il apparaît que l’utilisation plus soutenue de normes, à la fois dans le domaine géographique, mais aussi dans celui du bâtiment, est une bonne solution. Une des limites à l’utilisation de modèles normalisés pour la description d’environnement urbain complet en 3D est qu’il n’existe pas, à l’heure actuelle, de normes ni d’outils permettant de modéliser à la fois un environnement géographique avec les mécanismes de représentations géographiques comme le fait le format ShapeFile⁴³ et la possibilité d’avoir des maquettes numériques de bâtiments aussi riches sémantiquement qu’avec les IFC. Une approche alors largement abordée, à la fois par les chercheurs et les professionnels de ces deux domaines, consiste à coupler ces deux normes pour obtenir un tel système.

Le concept de base est la représentation des objets à l’aide de modèles géométriques et sémantiques. C’est le couplage de ces deux types de modèles qui donne la représentation des objets spatiaux dans l’environnement complet : d’un côté, le modèle sémantique permet de définir des classes d’objets et des relations pour les éléments spatiaux et les entités du modèle ; de l’autre côté, le modèle géométrique s’occupe de la représentation géométrique et des relations spatiales entre les objets spatiaux et leurs propriétés (Kolbe et Gröger, 2004). S’il existe de nombreux modèles géométriques différents pour diverses applications dans ces deux domaines, relativement peu de modèles sémantiques ont été développés. D’après (Isikdag et Zlatanova, 2009), cela est dû à la complexité de modéliser l’environnement spatial et ses objets dans différents scénarii et changements sur différentes zones et constructions.

Les normes citées ci-dessus font partie des modèles sémantiques les plus répandus, chacune dans leur domaine. Elles permettent, en étant combinées, de modéliser à la fois les espaces extérieur et intérieur d’un environnement urbain. Cependant, il n’est pas si facile de coupler ces deux domaines, et transférer des données d’un modèle à un autre, sur le plan à la fois géométrique et sémantique, est soumis à certaines difficultés (Mekawy et Östman, 2010) :

- La culture dans chaque domaine n’est pas la même et les modèles SIG et BIM sont développés dans des objectifs différents et visent des centres d’intérêt différents, même si une convergence existe. Il en résulte des barrières technologiques entre les deux mondes.

⁴³ Format initialement développé par ESRI pour ses logiciels commerciaux, devenus standard de fait, et largement utilisé par les logiciels libres.

- La plupart des modèles BIM existants sont parfaitement adaptés à l'intégration si le modèle est créé à partir de zéro lors de la phase de conception. La plupart des bâtiments ont cependant été construits il y a plusieurs années et il est difficile de faire des relevés clairs et complets de ceux-ci.
- La plupart des modèles BIM viennent de fichiers de CAO (DWG, DGN...) sans connaissance objet. Il faut alors enrichir ces modèles sémantiquement et géographiquement pour les intégrer.
- Il y a souvent différentes représentations des objets spatiaux dans différentes résolutions ou thématiques selon le type et l'échelle d'application. Si ce mécanisme de niveaux de détail pour est parfaitement maîtrisé par les SIG, il n'y a pas de norme ni de formalisme particulier sur ce point au sein des BIM. Ainsi, la maquette numérique est composée, en général, de tous les détails du modèle géométrique, chaque objet spatial pouvant même être représenté par plusieurs objets géométriques. Cela rend le processus d'automatisation de génération des niveaux de détail compliqué et gourmand en ressources, limitant ainsi les échanges possibles entre le BIM et les SIG.
- L'intégration des données SIG et BIM requiert également la plupart du temps la gestion des coordonnées. En effet, aujourd'hui les BIM sont rarement géoréférencés et lorsque c'est le cas, ils le sont dans un système de coordonnées local qui leur est propre. Il faut alors transformer les coordonnées entre les différents systèmes, depuis des systèmes locaux vers des systèmes géodésiques ou projetés.
- Enfin, les capacités des logiciels de CAO permettent de modéliser des objets avec des enveloppes paramétriques complexes, ce que ne permet pas, ou difficilement, les SIG.

Maintenant que nous avons vu les objectifs d'intégration des normes du BIM et des SIG, nous allons voir quelques solutions proposées dans cet objectif.

2.1 IFG

Le projet IFG a été initié en Norvège dans le but de fournir de l'information géographique au sein des IFC. Une description relativement complète peut être trouvée sur le site de l'IAI⁴⁴. Nous en reprenons ici les éléments importants pour notre étude.

Les motivations pour ce projet sont de pouvoir soutenir des ensembles de plus en plus grands d'exigences métier qui émergent dans les administrations tels que :

- le stockage cohérent de l'information à la fois géographique et spécifique au bâtiment, au sein d'une structure de données cohérente (rendu possible par des installations de communication entre les SIG et les systèmes AEC/FM et vice-versa) ;
- le support du développement des systèmes de contrôle de conformité des codes de construction et de planification.

Le développement des IFG a initialement prévu pour rendre le processus de soumission des plans de zonage et de construction plus efficace. Pour cela, des informations SIG sont stockées dans des maquettes de bâtiments issues de la CAO et des projets AEC/FM. Les informations SIG peuvent être de types divers comme : données de localisation sur une carte, propriétés, services utilitaires, zonage, facteurs de risques...

L'objectif du projet IFG est d'utiliser le modèle IFC comme norme pour l'échange d'informations limitées, mais significatives, entre les SIG et les systèmes AEC/FM et vice-versa. Le but est d'utiliser des entités qui existent déjà dans les vues de coordination et de vérification du code du schéma IFC afin de pouvoir utiliser les outils, techniques et possibilités déjà existantes.

Comme nous l'avons déjà évoqué, le but des IFG n'est pas de créer un SIG complet dans les IFC, mais de créer une passerelle entre ces deux mondes ; c'est un modèle qui reconnaît les modèles existants et parfaitement

⁴⁴ http://www.iai.no/ifg/Content/ifg_index.htm

adaptés à ce rôle, comme le fait GML pour les SIG. Les IFG tentent de faciliter la correspondance des éléments entre le monde du BIM et les SIG.

L’approche choisie pour créer cette passerelle entre les deux univers est d’effectuer un « chevauchement » entre les deux modèles en fournissant des entités dont les occurrences et la position dans l’espace peuvent être reconnues dans les deux mondes. Cela implique notamment de comprendre les informations provenant des projets AEC/FM dans des systèmes de projection de carte avec une précision raisonnable. Généralement, la zone de chevauchement entre plusieurs types de données issues de milieux différents est considérée cohérente dans un carré de un kilomètre de côté, centré sur le bâtiment au centre d’intérêt. Dans cet espace, l’erreur maximale, sur les bords du carré, due à la projection est estimée à quatre millimètres.

Ce projet a conduit à la création d’une norme de mappage entre la version XML des IFC et GML. Il existe d’autres approches pour coupler le BIM avec les SIG, dont une grande partie tente de convertir le modèle géométrique ou sémantique des IFC en CityGML. C’est ce que nous allons étudier dans le prochain paragraphe.

2.2 L’intégration des IFC dans CityGML

Comme nous l’avons évoqué plus tôt, chacun des modèles IFC et CityGML possèdent à la fois des mécanismes de modélisation de la géométrie et de la sémantique.

Dans les travaux que nous allons voir ici, plusieurs approches sont identifiées : celles qui consistent à ne transformer de manière unidirectionnelle que la géométrie (souvent des IFC vers CityGML comme le propose (Kolbe et al., 2009), ou des logiciels commerciaux comme IfcExplorer⁴⁵ et FME⁴⁶), celles qui traitent à la fois de la géométrie et de la sémantique (Isikdag et Zlatanova, 2009b) (Berlo, 2009), et les approches présentant des solutions pour effectuer des échanges bidirectionnels entre modèles ((Nagel et al., 2009) et (Mekawy et Östman, 2010)).

Nous allons présenter chacune des approches énoncées ci-dessus de manière un peu plus détaillée dans les paragraphes suivants, après avoir décrit le framework généralement employé.

2.3 Un framework de conversion 3D conçu par l’université technique de Berlin

La première approche qui nous intéresse, et certainement la plus populaire, est celle qui a été proposée par Thomas Kolbe à l’université technologique de Berlin pour la création de modèle de bâtiments 3D (Nagel et al., 2009). Ce framework est constitué de deux étapes (qu’il est possible de voir dans la Figure 26) :

1. Des données ou géométries de bâtiments et de zones urbaines pouvant provenir de différentes sources. Ces données, stockées dans des formats tels que X3D, DXF⁴⁷, KML⁴⁸ et COLLADA⁴⁹, sont généralement utilisées pour créer des modèles de visualisation. Ces modèles sont destinés à être intégrés avec des données sémantiques qui représentent des contextes spatiaux. L’enjeu de cette phase est la création d’un modèle spatio-sémantique qui permet aux entités sémantiques d’être liées avec des composants géométriques. Pour cette phase, CityGML a été choisi comme schéma cible.
2. Dans la seconde étape, les modèles CityGML sont utilisés pour reconstruire les modèles IFC. Bien que CityGML et les IFC ne soient pas étudiés pour travailler à la même échelle et que les ontologies derrière chaque modèle sémantique n’aient pas la même portée, ces deux modèles sont en accord sur

⁴⁵ IfcExplorer CityGML Export, http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export

⁴⁶ FME Desktop translator/convert software, <http://www.safe.com/fme/fme-technology/fme-desktop/overview/>

⁴⁷ Drawing eXchange Format, format d’échange pour les fichiers de CAO, utilisé également par les SIG.

<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=12272454&linkID=10809853>

⁴⁸ Keyhole Markup Language, <https://developers.google.com/kml/documentation/?hl=fr-FR>

⁴⁹ COLLABorative Design Activity, <http://www.khronos.org/collada/>

la notion de bâtiment et sa décomposition sémantique (Benner et al., 2005) (Isikdag et Zlatanova, 2009).

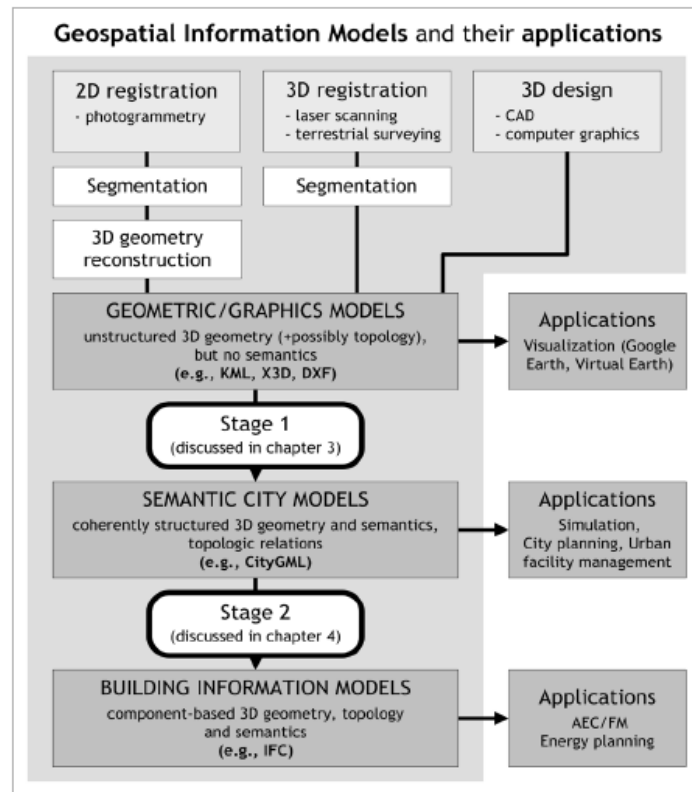


Figure 26. Processus de reconstruction en deux étapes (Nagel et al., 09)

Le framework proposé et ses principaux résultats s'appuient sur un certain nombre d'avantages potentiels pour le domaine de la modélisation de ville en 3D :

- comme CityGML et les IFC ont la même notion de la sémantique des bâtiments, ces deux systèmes peuvent offrir une base de connaissances pour un accès et des réponses rapides dans la construction de modèles de bâtiments et une recherche d'éléments du bâtiment dans les modèles de bâtis.
- Le framework supporte une interface parfaitement définie dans le milieu de la construction, étant donné que CityGML est utilisé comme modèle sémantiquement riche. Cela permet de relier et échanger des informations sans perte entre les modèles 3D visualisés (CAD et d'autres) et le modèle IFC plus sophistiqué et détaillé.

Un autre projet de recherche sur le sujet, en partenariat avec l'institut d'informatique appliquée au centre de recherche de Karlsruhe, et l'Université de sciences appliquées de l'Hochschule Karlsruhe en Allemagne, a été réalisé pour la conversion des IFC en CityGML. Le projet vise à développer un algorithme qui permet une transformation complète et automatisée de modèles de bâtiments IFC en modèles CityGML. La recherche a porté dans un premier temps sur les niveaux de détail LOD1 et LOD2 de CityGML. L'objectif de l'algorithme proposé est de créer une représentation géométriquement et sémantiquement valide de LOD1 qui peut également être appliquée pour le LOD2 (Kolbe et al., 2009).

Le LOD1 est une représentation des objets à l'échelle régionale, voire urbaine. Les caractéristiques des bâtiments à ce niveau de détail sont, d'après les auteurs :

- Modèle en bloc
- Bâtiments prismatiques
- Enveloppe extérieure approchée du bâtiment

- Toits plats
- Pas de débord de toit
- Pas de surplombs de dalle

Pour ce faire, une transformation en six étapes est décrite (Figure 27) :

1. Simplification au niveau du modèle sémantique. Cette étape est axée sur les éléments du bâtiment qui forment l’enveloppe externe (éléments verticaux, par exemple IfcSlab, élément horizontaux, par exemple IfcWall, IfcColumn, IfcBeam et élément du toit). Le résultat est une vue simplifiée du modèle de bâtiment IFC.
2. Approche 2D. Les éléments du bâtiment, horizontaux et verticaux, sont projetés selon un axe X, Y. Chaque étage est ici traité séparément. Le résultat donne un ensemble de polygones formant le contour du bâtiment.
3. Calcul des empreintes pour chaque étage. Cette étape est basée sur une union 2D des éléments constituant le contour d’un étage à l’aide d’opérations booléennes 2D. Le résultat est un ensemble d’empreintes 2D pour chaque étage.
4. Extrusion linéaire des empreintes des étages. Une extrusion de l’empreinte est faite selon l’axe Z du référentiel du bâtiment selon la hauteur de l’étage. Il en résulte une surface englobante pour chaque étage.
5. Généralisation des éléments du toit. Cette étape vise à transformer les toits en solides 3D extrudés selon le même principe que pour les étages.
6. Composition du modèle surfacique final. Pour cela, on fusionne les solides adjacents avec la même empreinte puis on identifie la surface englobante.

En parallèle de ces travaux, un logiciel commercial pour la conversion des IFC en CityGML a été développé, en 2008, au Centre de recherche de Karlsruhe. Ce logiciel, IFCExplorer, est conçu pour la conversion automatique d’un modèle IFC en CityGML dans différents niveaux de détails.

2.4 Framework pour la conversion automatique de BIM (IFC) en CityGML

Le troisième projet que nous présentons possède la même logique de conversion des IFC en CityGML. Il s’agit plus d’une discussion que d’un projet concret. L’article (Isikdag et Zlatanova,2009b) propose les bases d’un framework pour une conversion automatique des IFC en CityGML. Cet article fait valoir que deux étapes sont nécessaires à cette conversion : la transformation de l’information sémantique et la transformation des géométries. Ils affirment également que les deux étapes ne peuvent pas être effectuées séparément, puisque les deux modèles, CityGML et IFC, ont des objets et classes très divers. Ainsi, la correspondance entre les objets des deux modèles doit être faite soigneusement puisqu’un objet dans l’un des modèles peut être relié à un certain nombre d’objets dans l’autre. Ils suggèrent ainsi des opérations à effectuer pendant le processus de conversion :

- En premier lieu, une compréhension de la composition sémantique des deux modèles doit être définie par un ensemble de règles pour les classes correspondantes. Ces règles doivent être clairement définies séparément pour chaque niveau de détail de CityGML.
- Sur la base des règles construites, le mapping des classes et des objets peut être facilité par la construction d’algorithmes nécessaires à la simplification des modèles.
- Un LOD souhaité pour une application doit contrôler le processus de conversion en définissant toutes les informations qui doivent être transformées, ainsi les attributs des objets CityGML seront également construits.

Les auteurs décrivent leur approche au travers d’exemples de conversions réelles d’IFC en CityGML avec plusieurs niveaux de détail et pour les modèles sémantiques et géométriques.

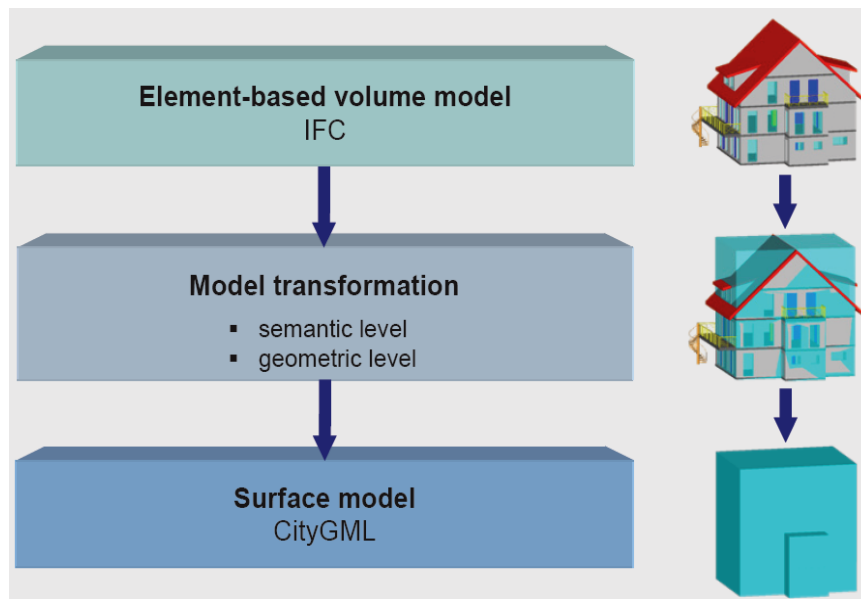


Figure 27 Transformation de modèles IFC en CityGML (Kolbe et al., 2009)

2.5 Une ADE CityGML

Un projet de recherche intéressant est celui de (Berlo et Laet, 10) qui vise à utiliser le mécanisme d'extension du modèle de données de CityGML, ADE (Application Domain Extensions), pour y intégrer les IFC. Quelques extensions existent déjà (CityGML, 2010). Cette nouvelle extension doit permettre d'intégrer la sémantique et les propriétés des IFC. L'approche est basée sur le projet open source BIMserver⁵⁰ qui devra être capable d'exporter des IFC en CityGML, incluant géométrie, sémantique et propriétés. Cette extension est appelée « GeobIM »⁵¹.

Le développement de l'extension GeobIM pour CityGML se fait sur plusieurs niveaux. D'abord, les types connus d'objets dans CityGML tels que chambre, fenêtre, porte, bâtiment... sont étendus avec des propriétés supplémentaires des IFC (par exemple hauteur et largeur des portes et fenêtres). Ensuite, le niveau suivant est l'extension de « `_AbstractBuilding` » avec une propriété supplémentaire, ce qui crée un lien vers la classe de base de l'extension, appelée `VisibleElement`.

Les auteurs ont fait le choix de ne pas implémenter la totalité des classes IFC dans l'extension. En effet, sur plus de 900 classes et propriétés pour la version actuelle des IFC, la majorité sont dédiées à la représentation géométrique, les relations et la topologie. Des recherches empiriques ont été faites afin de déterminer les classes IFC qui pourraient être utiles dans les SIG ; il en est ressorti qu'il y a environ 60 à 70 classes IFC qui pourraient être transformées en une extension de GeobIM (Berlo, 2009). Ces classes sont répertoriées dans la Figure 28. Les propriétés de ces classes peuvent être transformées en attributs CityGML. Certaines classes IFC correspondent aussi directement à un type CityGML. Par exemple, `IfcBuilding` correspond à `_Building` dans CityGML.

Un prototype de ce projet a été fait. Pour l'instant il n'exporte des IFC qu'en LoD4 et certains viewers ne sont pas compatibles avec tous les éléments exportés.

⁵⁰ <http://bimserver.org/>

⁵¹ http://www.citygmlwiki.org/index.php/CityGML_GeobIM_ADE

- IfcAnnotation
- IfcBeam
- IfcBuilding
- IfcBuildingElementComponent
- IfcBuildingElementPart
- IfcBuildingElementProxy
- IfcBuildingStorey
- IfcChamferEdgeFeature
- IfcColumn
- IfcCovering
- IfcCurtainWall
- IfcDiscreteAccessory
- IfcDistributionChamberElement
- IfcDistributionControlElement
- IfcDistributionElement
- IfcDistributionFlowElement
- IfcDistributionPort
- IfcElectricalElement
- IfcElectricDistributionPoint
- IfcElementAssembly
- IfcEnergyConversionDevice
- IfcEquipmentElement
- IfcFastener
- IfcFooting
- IfcGrid
- IfcMechanicalFastener
- IfcMember
- IfcOpeningElement
- IfcPile
- IfcPlate
- IfcProjectionElement
- IfcProxy
- IfcRailing
- IfcRamp
- IfcRampFlight
- IfcReinforcingBar
- IfcReinforcingMesh
- IfcRoof
- IfcRoundedEdgeFeature
- IfcSite
- IfcSlab
- IfcSpace
- IfcStair
- IfcStairFlight
- IfcStructuralCurveConnection
- IfcStructuralCurveMember
- IfcStructuralCurveMemberVarying
- IfcStructuralLinearAction
- IfcStructuralLinearActionVarying
- IfcStructuralPlanarAction
- IfcStructuralPlanarActionVarying
- IfcStructuralPointAction
- IfcStructuralPointConnection
- IfcStructuralPointReaction
- IfcStructuralSurfaceConnection
- IfcStructuralSurfaceMember
- IfcStructuralSurfaceMemberVarying
- IfcTransportElement
- IfcVirtualElement
- IfcFurnishingElement
- IfcWall
- IfcWallStandardCase
- IfcWindow

Figure 28. Liste de classes IFC utiles dans un contexte géographique (extrait de (Berlo, 2009))

2.6 The Unified Building Model

La dernière approche que nous allons présenter dans cet état de l’art diverge quelque peu des autres même si la finalité est identique : intégrer les IFC avec CityGML. Le projet est proposé par (Mekawy et Östman, 10). Il se base sur la notion de modèle unifié (utilisé notamment par Miguel et al., 2002) qui est défini, pour la partie théorique, comme un modèle de surensemble étendu pour contenir tous les éléments et objets à la fois des IFC mais aussi de CityGML. Ce modèle se pose comme un modèle intermédiaire pour relier les objets entre ces deux normes. Cette approche de modèle unifié permet donc une conversion bidirectionnelle entre les IFC et CityGML plus poussée que les modèles précédents.

Le modèle unifié, appelé Unified Building Model (UBM) est basé sur le concept des ontologies de référence⁵². Les auteurs proposent alors un modèle de bâtiment intermédiaire qui comprend les concepts des modèles de bâtiments des IFC et de CityGML. Les étapes pour la construction de l’UBM sont décrites dans la Figure 29 et détaillée ci-dessous (Mekawy et Östman, 2010) :

- 1) Élicitation⁵³ du modèle de bâtiment IFC. Comme il n’existe pas de modèle de bâtiments IFC universel (Kolbe, 2009), des chercheurs comme (Benner et al., 2005) et (Kolbe et al., 2009) ont fourni des modèles UML informels pour les IFC en fonction de leur compréhension de la norme. De la même façon, les auteurs de l’UBM construisent leur propre modèle IFC, en sélectionnant classes et propriétés qu’ils veulent intégrer.
- 2) Développement de l’UBM. Durant cette phase, les éléments communs aux ontologies des IFC et CityGML sont identifiés afin d’en faire une correspondance directe ou indirecte.
- 3) Conversion bidirectionnelle entre les modèles de bâtiments IFC et CityGML à travers l’UBM. Cette phase détermine les relations entre les objets et concepts combinés.
- 4) Identification des contraintes sous la forme de règles pour la transformation des informations entre les IFC et CityGML depuis l’ontologie de référence.

⁵² Une ontologie de référence est basée sur le développement d’une terminologie tierce plus expressive qui sert d’intermédiaire pour l’interopérabilité entre deux domaines intégrés.

⁵³ En knowledge management, « l’élitication » est l’action d’aider un expert à formaliser ses connaissances.

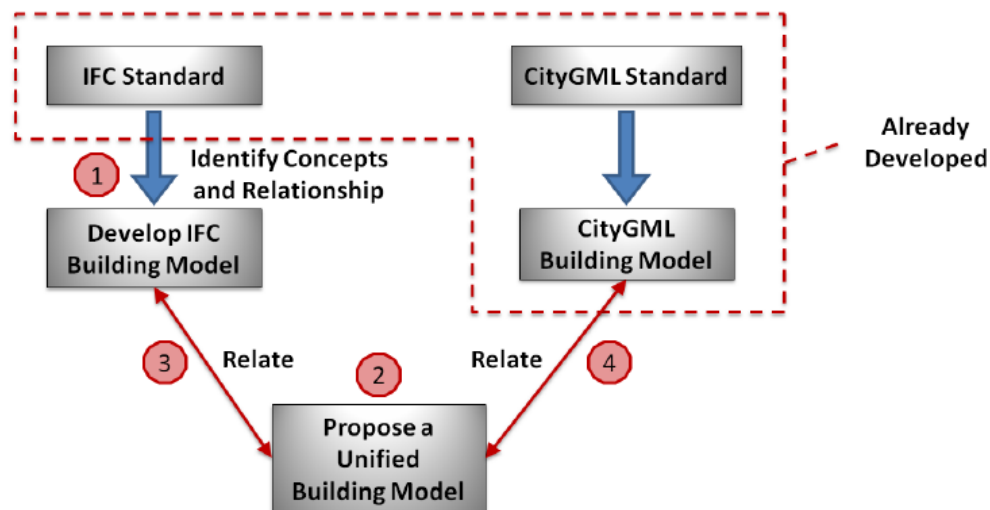


Figure 29. Les étapes pour développer un UBM

Pour construire l'UBM, les classes avec leurs concepts ont été initialement rassemblées à partir de deux modèles, mais leurs relations ont été omises (les modèles choisis sont des modèles de bâtiments particuliers ce qui fait que l'UBM ne regroupe pas l'ensemble des classes et concepts de ces deux normes). Ensuite, les concepts qui se chevauchent ont été fusionnés et de nouveaux objets ont été créés de telle sorte que les objets à la fois intérieurs et extérieurs soient représentés. Enfin, les relations entre les objets ont été reconstruites pour produire l'UBM. Le modèle est également développé dans différents LOD qui correspondent à ceux de CityGML.

2.7 Limites et conclusion

Le domaine de recherche pour l'intégration des IFC et CityGML est un sujet d'actualité. De nombreuses équipes sont identifiées et travaillent, tant dans le domaine public que dans des laboratoires privés, pour résoudre les contraintes de cette intégration. La tendance pour résoudre cette problématique est une approche de conversion unidirectionnelle, avec un accent porté sur les géométries (la plupart du temps des IFC vers CityGML et pas l'inverse) (Kolbe et al., 2009), (Nagel et al., 2009) ou des extensions à CityGML pour enrichir sa sémantique afin de faciliter l'import d'IFC (Berlo et Laat, 2010). Il existe aussi des propositions pour développer des frameworks qui peuvent être utilisés pour effectuer une communication bidirectionnelle entre les deux modèles (Mekawy et Östman, 2010). Néanmoins, cette approche par framework n'est pas toujours homogène (Isikdag et Zlatanova, 2009b).

On remarque en particulier que toutes ces approches sont basées sur les normes IFC et CityGML exclusivement. Aucun de ces travaux n'est extensible vers d'autres formats tels DWG ou DXF par exemple. De plus, malgré l'existence de plusieurs méthodes d'intégration, la plupart des travaux dans ce domaine ont porté sur la conversion des IFC vers CityGML, ce qui est plus simple que l'inverse. Ainsi, tous les efforts précédents ne contribuent que partiellement à la conception d'un framework pour une conversion d'ordre purement sémantique et géométrique, nécessaire néanmoins à une intégration complète et une conversion bidirectionnelle entre CityGML et les IFC (Isikdag & Zlatanova, 2009b)(Nagel, 2009)(Mekawy et Östman, 2010). Une des approches qui semble la plus aboutie est celle de (Mekawy et Östman, 2012). Cependant, le projet est encore jeune et plusieurs limites existent. L'UBM ne traite que les bâtiments (seuls les modèles de bâtiments IFC et CityGML sont étudiés en tant que point de départ). Le modèle est construit sur un nombre limité de modèles de bâtiments, et les procédés de conversion ne sont pas construits ou vérifiés par des outils d'ontologies. Le modèle gagnerait en flexibilité et possibilités de modélisation et de vérification de modèles (cohérence et consistance) à être créé à partir d'un vocabulaire commun et indépendant en termes de

structuration de hiérarchie de classe, comme ceux issus des outils du web sémantique ou de langage comme OWL ou C-DMF.

Nous avons donc, d’un côté, le monde du BIM qui voit les SIG comme un moyen de concevoir et d’intégrer les données du BIM dans un environnement spatial, et de l’autre côté, le BIM est vu comme une source de données essentielle pour la construction des environnements par les utilisateurs de SIG.

Dans le chapitre contexte, nous avons présenté les travaux antérieurs et la norme IFC. Maintenant, nous devons étudier plus en avant les SIG et les normes associées pour répondre à nos objectifs.

3 La 3D dans les SIG

L’idée de construire un SIG avec des représentations à trois dimensions part du constat que nous vivons dans un monde à trois dimensions et que la représentation traditionnelle des données géographiques en projection est, dans certains domaines, source d’imprécision et provoque des pertes de données (suppression des objets volumiques, impossibilité de représenter correctement les falaises et les surplombs, bâtiments fortement dégénérés...). Il en découle que certaines applications sont impossibles à réaliser (propagation du bruit dans l’espace, guidage d’engins militaires...) ou bien donnent des résultats incorrects en raison de la non-conformité des données avec le monde réel (calcul de distances entre bâtiments et lignes électriques, par exemple) (De La Losa, 2000). Or, à l’ère du numérique et de la 3D, non seulement logicielle, mais aussi matérielle, les données géographiques peuvent être tridimensionnelles et s’affranchir des contraintes de la représentation bidimensionnelle. La 3D permet notamment de représenter des réseaux souterrains complexes (eaux, électricité, gaz...), de faire des calculs d’inter-visibilité ou de trajectoires plus précis que la 2D et de rendre plus accessible la donnée géographique aux personnes qui ne sont pas familières avec celles-ci.

Dans la thèse de (Poupeau, 2008), le SIG 3D est présenté comme étant « un système capable de modéliser, représenter, gérer, manipuler, analyser et supporter des décisions fondées sur l’information associée à un phénomène 3D » et doit constituer une plateforme susceptible d’intégrer des données modélisées différemment, qui sont d’origine et de dimension variées, et, au-delà de cette intégration, de les analyser, c’est-à-dire extraire de l’information utile pour supporter des décisions. Cependant les SIG 3D actuels sont le plus souvent restreints à de simples viewers-3D dotés de quelques fonctionnalités. En effet, le développement des SIG 3D est freiné par (selon (Poupeau, 2008) et (Abdul-Rahman, 2007) :

- la difficulté d’acquérir des objets à la géométrie complexe ;
- l’incapacité de gérer l’intégration de données dont la géométrie et la topologie sont modélisées de manières différentes ;
- les problèmes de (re)mise en cohérence des données et de mise en relation des objets. Les modèles de description des relations entre objets, fondés sur des principes de topologie, nécessitent une reconstruction précise de la géométrie des objets. Si deux objets ne sont pas parfaitement mis en relation, certains calculs risquent d’être erronés ;
- le manque d’outils d’analyse spatiale et de requêtes 3D ;
- les structures de données, puisque bien qu’existant de nombreuses structures à la fois pour les données 2,5D et 3D, chacune possède ses points forts et ses faiblesses ;
- les modèles de données : les données spatiales peuvent être modélisées de différentes manières, chaque modèle devant être capable de décrire les relations entre les données de telle sorte que l’information puisse être générée à partir de celles-ci.

3.1 Premiers pas vers la troisième dimension

Le premier pas vers la troisième dimension dans les bases de données géographiques a consisté à ajouter un attribut altitude sur chacun des points des bases de données en deux dimensions (Oosterom, 1994). Le résultat

est appelé 2,5D et non 3D, car il subsiste encore de fortes contraintes liées au fait que l'altitude soit fonction de x et de y . On ne peut en effet pas superposer plusieurs points. Pour représenter le terrain à une échelle suffisamment petite (échelle inférieure au 1 : 100 000), la 2,5D peut être jugée satisfaisante puisque l'on arrive à négliger les falaises et autres surplombs grâce au processus de généralisation. Cependant, la 2,5D présente l'avantage de pouvoir être gérée au moyen de modèles bidimensionnels puisque l'altitude est stockée comme un attribut alphanumérique quelconque. Les SIG 2D n'ont donc subi aucune transformation pour pouvoir gérer cette troisième dimension (mis à part l'élaboration d'un outil de visualisation à effets tridimensionnels).

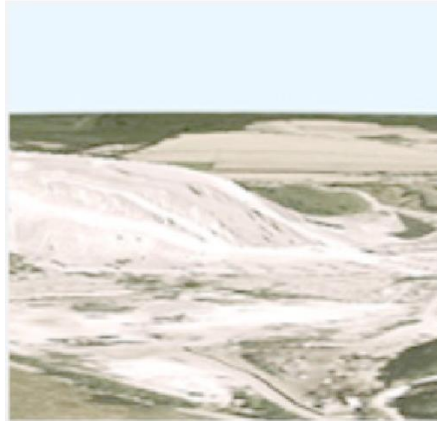


Figure 30. Exemple de MNT habillé avec une orthophoto pour plus de réalisme (IGN).

La modélisation du terrain est une problématique à part entière. Il existe de nombreux moyens de représenter un terrain, dont les représentations raster et vecteur. Dans le cas des MNT (Modèles Numériques de Terrain) Raster (on parle aussi de matrice d'altitude), on possède une répartition régulière de points qui définit un maillage de la surface du terrain, les dimensions de la maille définissant la résolution spatiale planimétrique du MNT (plus l'espacement des points est serré, plus la résolution est grande et plus le MNT est fin et riche en détails topographiques). Les MNT vecteur, sont en fait une triangulation de la surface du terrain. On l'utilisait déjà à l'époque des premiers SIG vectoriels dans les années 70. Les semis de points de mesure utilisés dans ce type de MNT sont souvent irréguliers et la densité varie en fonction de la complexité du relief et de la précision recherchée. Chaque point est relié à deux voisins pour former un réseau de triangles qui ne doit laisser apparaître aucun trou et aucune superposition de triangles. On parle également de TIN (Triangular Irregular Network). La Figure 30 montre une capture d'écran provenant du Géoportail⁵⁴ présentant un modèle numérique de terrain texturé avec une orthophotographie.

La construction de ces reliefs 2,5D est soit entièrement intégrée aux SIG (Gestionnaire de MNT de GeoConcept 3D⁵⁵, par exemple, ou bien ArcGIS 3D Analyst d'ESRI⁵⁶), soit c'est un logiciel à part entière pour la visualisation comme Leveller de Daylon Graphics⁵⁷, Vistapro Renderer de MonkeyByte⁵⁸, ou Terragen de PlanetSide Software⁵⁹ (création d'un relief à partir d'une image composée de pixel où la couleur détermine une altitude). Lorsque le module 3D (2,5D) est intégré au SIG, certaines applications peuvent être proposées : calculs de pentes ou zones visibles à partir de points. Les résultats de ces calculs dépendent bien sûr des données, et comme ces données ne sont qu'en 2,5D, les zones visibles peuvent être assez différentes de ce qui peut être

⁵⁴ Le Géoportail est un portail Web public appartenant à l'IGN et permettant l'accès à des services de recherche et de visualisation de données géographiques ou géolocalisée pour l'ensemble du territoire français. www.geoportail.fr

⁵⁵ <http://www.geoconcept.com/3D-for-GeoConcept.html>

⁵⁶ <http://www.esri.com/software/arcgis/extensions/3danalyst/index.html>

⁵⁷ <http://www.daylongraphics.com/products/leveller/>

⁵⁸ <http://store.monkeybyte.com/index.php/tools/vistapro-renderer-download-pc.html>

⁵⁹ <http://www.planetside.co.uk/>

constaté dans la réalité (les bâtiments et forêts ne sont généralement pas pris en compte, car seul le terrain est utilisé).

L’IGN constitue depuis plusieurs années une base de données en 2,5D : la Base de Données Topographique (BD Topo). Cette base de données regroupe différents thèmes (voies de communication routière, hydrographie, bâtiments, végétation, limites administratives...) pour des échelles allant du 1 : 5 000 au 1 : 50 000.

Les applications sont diverses, certaines s’appliquent à de vastes MNT alors que d’autres sont ponctuelles. On peut définir également le modèle numérique d’élévation qui pour chaque point de coordonnées (x, y) donne l’altitude maximale, c’est-à-dire un modèle qui représente à la fois le sol et le sur-sol (objets 3D).

3.2 La topologie en 3D

Les modèles topologiques 3D sont souvent une simple extension des modèles 2D existant. Ainsi leur base est composée par les primitives géométrico topologiques que sont le nœud, l’arc et la face auxquels on ajoute une nouvelle primitive, le volume, délimité par un ensemble de faces. Depuis les années 90, plusieurs propositions de modèles topologiques sont ainsi apparues (Molenaar, 1990), (recommandations de l’ISO (ISO/TC211, notamment ISO 19107)), (Zeitouni, 1995) qui a travaillé sur la notion de topologie sémantique (placement des objets les uns par rapports aux autres et non plus par rapport à un référentiel).

3.2.1 La topologie dans les IFC et CityGML

Dans le cas où l’on souhaite construire un Système d’Information Géographique qui se base sur des normes existantes, la topologie est souvent une des possibilités offertes et ne nécessite pas forcément une prise en compte spécifique dans la définition du système comme il a été fait dans les travaux de recherche à l’IGN. Nous allons voir rapidement comment l’aspect topologique des modèles de bâtiments construits à partir de la norme IFC se manifeste, puis nous ferons de même avec les modèles urbains basés sur la norme CityGML.

Commençons l’analyse par la gestion de la topologie dans les IFC. Tout d’abord, précisons un peu ce que nous attendons de la topologie dans le bâtiment. D’après l’article de (Chena et al., 2004), il est possible de définir cinq catégories de relations topologiques sur les éléments et espaces d’un bâtiment :

- un composant est adjacent à un autre (adjacence),
- les composants sont disjoints (séparation),
- un composant en contient un autre (contenance),
- un composant en intersecte un autre (intersection),
- un composant est connecté à un autre (connectivité)

Il est possible de créer des représentations topologiques des bâtiments, mais ces relations ne sont pas toutes explicitées dans la norme IFC. On peut citer en exemple pour les relations existantes, la notion d’adjacence pour les pièces, déterminée par les objets Space Boundaries. Le schéma de la Figure 31 montre un extrait des classes IFC permettant de définir une représentation topologique à l’aide notamment de nœud (IfcVertex), d’arcs (IfcEdge) et de face (IfcFace) (le diagramme vient d’un article de (Clemen et Gründig, 2006)). La définition de cette représentation topologique est tirée de la norme ISO/IS 10303-42:1994 à laquelle des adaptations ont été apportées (convention de nommage, adaptation des entités STEP pour la prise en compte de l’héritage multiple et l’héritage non-exclusif, omission de l’attribut nom sur la représentation de l’objet...) (voir les spécifications IFC sur le sujet⁶⁰).

Comme pour les IFC, la norme CityGML s’appuie sur un modèle de représentation géométrico-topologique (OpenGIS, 2008). Les propriétés spatiales des éléments de CityGML sont représentées par un sous-ensemble d’objets issus du modèle géométrique de GML 3 (basé sur la norme ISO 19107 permettant de décrire et

⁶⁰ <http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm>, section IFCTOPOLOGYRESOURCE

manipuler les caractéristiques spatiales d'un phénomène géographique en 3D). On peut ainsi définir de façon explicite à la fois les représentations géométriques, mais aussi topologiques. On y retrouve plusieurs classes pour définir les points, courbes, surfaces et solides nécessaires à ces représentations. On peut ensuite définir, à partir de ces géométries simples, des géométries combinées qui peuvent être une agrégation, complexes ou composites, chacune devant respecter un certain degré de topologie. Ainsi, les relations spatiales entre composants d'une agrégation de géométries n'ont aucune restriction topologique (ils peuvent être disjoints, se chevaucher, se toucher, ou être séparés), alors que les géométries complexes sont topologiquement structurées (chaque partie doit être disjointe des autres, sans se chevaucher et sont autorisées à toucher ou partager des parties sur la frontière des éléments). Les géométries composites sont des complexes particuliers qui ne peuvent contenir que des éléments de même dimension, disjoints, mais connectés à la frontière.

CityGML prévoit une modélisation explicite de la topologie, par exemple le partage d'objets de la géométrie entre éléments ou autre géométrie. Une partie de l'espace est représentée seulement une fois par un objet géométrique et est référencé par d'autres éléments ou géométries complexes qui sont définis ou bordés par la même géométrie : la redondance est évitée et les relations topologiques explicites entre les parties conservées. Pour implémenter la topologie, CityGML utilise le concept XML des XLinks fourni par GML. Ainsi, chaque objet géométrique qui pourrait être partagé est associé à un identifiant unique qui peut être référencé par une propriété géométrique GML en utilisant l'attribut href. L'inconvénient de ce concept est le fait que, contrairement au modèle topologique de GML3, la navigation entre objets topologiquement liés ne peut se faire immédiatement que dans un sens (de l'agrégation vers le composant).

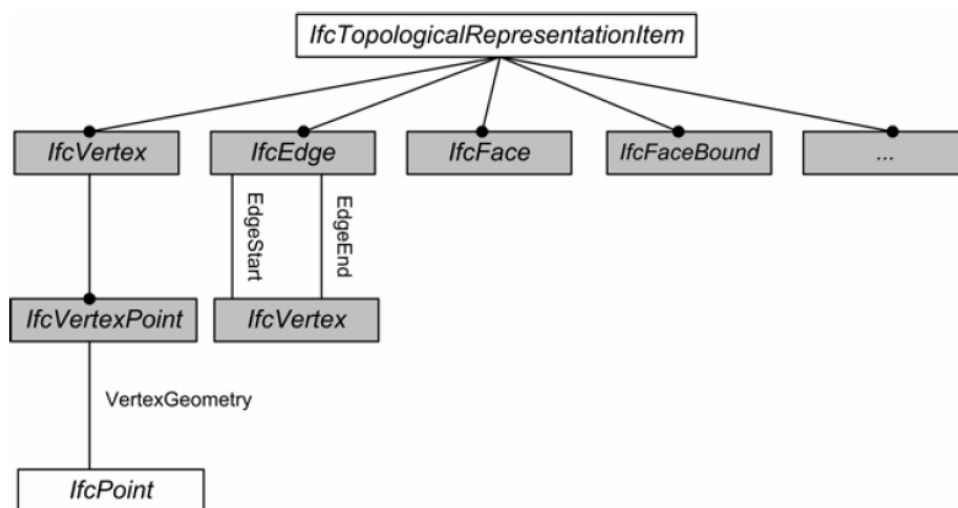


Figure 31. La topologie dans les IFC (extrait de (Clemen et Gründig, 2006))

3.2.2 Gestion d'un ensemble de données important

Cette partie va présenter quelques techniques d'optimisation simples de rendus de scènes 3D. Elle est basée en partie sur l'article de (Baxter, 2000) et vise à optimiser l'interactivité lors de l'affichage de grandes scènes urbaines en trois dimensions. En effet, les modèles 3D que nous sommes capables de générer aujourd'hui pour représenter de vastes scènes dépassent les capacités de la plupart des cartes graphiques, même si leur puissance ne cesse d'augmenter. Il faut donc optimiser la phase de rendu pour obtenir une interactivité satisfaisante lors du parcours de ces scènes. On parle bien sûr ici de visualisation en temps réel. Il en est de même pour les différents calculs qui sont rendus difficiles. Il est alors nécessaire de travailler avec des données simplifiées. Une des techniques d'optimisation de l'ergonomie et du confort de l'utilisateur consiste d'ailleurs à pré-calculer des scènes et afficher un résultat tronqué afin d'obtenir une bonne interactivité. Cependant, cette façon de procéder limite les interactions que l'utilisateur peut espérer avec la scène (on peut par exemple pré-calculer un quartier de ville avec de nombreux bâtiments et objets urbains en 3D et limiter l'affichage au

quartier avec ces objets particuliers ; mais si l’utilisateur souhaite visualiser le quartier voisin, ajouter ou supprimer des objets, il faudra calculer à nouveau la scène pour prendre en compte ces interactions).

Revenons maintenant à notre problématique initiale. L’optimisation que nous recherchons passe par une réduction du nombre de triangles (ou polygones) qu’on affiche simultanément dans une scène 3D. Les techniques de réduction existantes partent souvent d’un postulat simple : on ne dessine que ce que l’on peut voir, c’est-à-dire qu’on ne dessine pas les détails que l’on ne peut voir, et l’on ne dessine pas les objets que l’on ne voit pas. Ceci entraîne l’utilisation de deux techniques que nous allons décrire par la suite, les niveaux de détail et les techniques de culling.

3.2.3 Niveaux de détails (Levels of Details)

La technique des niveaux de détail consiste à calculer une série d’approximation des objets avec de moins en moins de détails, et par conséquent de polygones. Le plus souvent les LoD doivent être pré-calculés car les algorithmes sont coûteux en temps. Il existe deux manières d’utiliser les LoD : soit de manière discrète (on possède alors un ensemble d’approximations), soit de façon continue (on stocke l’opération de simplification et recalcule la meilleure approximation à chaque image). La Figure 32 nous montre un exemple de plusieurs LoD discrets calculés sur une sphère. La réduction du nombre d’arêtes est significative et pourtant la représentation de l’objet n’est pas affectée si les coefficients d’éloignement sont bien configurés (un objet placé loin peut être simplifié).

Visual impact comparisons and measurements






Image					
Vertices	~5500	~2880	~1580	~670	140
Notes	Maximum detail, for closeups.				Minimum detail, very far objects.

Figure 32. Exemple de niveaux de détail discrets sur une sphère

Le mécanisme classique des LoD présenté ci-dessus peut être complété par une autre technique de LoD, les H-LoD (Hierarchical Levels of Detail). La description que nous allons en faire ici est tirée de l’article de (Erikson et al., 2001). Le but des H-LoD est d’accélérer l’affichage de scènes massives statiques ou dynamiques en utilisant une simplification hiérarchique. Contrairement aux autres LoD conventionnels, les H-LoD sont générés en simplifiant des portions séparées de la scène afin de créer un ensemble simplifié et plus fidèle que des approximations importantes d’objets isolés. Le principe de fonctionnement des H-LoD est relativement simple : on se base sur la structure du graphe de scène contenant tous les éléments géométriques à afficher. Le H-LoD va alors associer pour chaque nœud du graphe de scène une représentation spécifique qui englobe toute sa descendance. L’avantage de cette technique est qu’il est possible de couper des portions entières de la scène en affichant le H-LoD approprié d’un nœud (il n’est pas nécessaire de charger les nœuds fils). La Figure 33 illustre ce fonctionnement en montrant à la fois les H-LoD de la scène (Original Model) et les LoD discrets classiques de chaque objet.

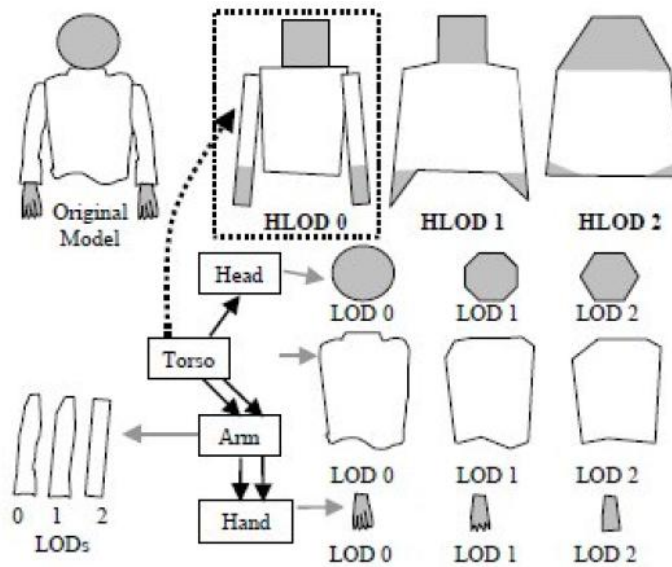


Figure 33. Fonctionnement des niveaux de détails hiérarchiques

Si l'on compare encore une fois l'implémentation de ces techniques dans les IFC et CityGML, on remarque que les IFC ne prévoient pour l'instant aucun mécanisme de normalisation des LoD, le besoin au niveau des bâtiments ne s'en faisant pas forcément ressentir. Par contre CityGML définit clairement cinq niveaux de détails à utiliser pour la représentation des scènes (illustrés dans la Figure 34) :

- **Niveau 0** : Ce niveau correspond le plus souvent à un modèle numérique de terrain en 2,5D, avec éventuellement l'application de photos aérienne. Le « sur-sol » n'est pas représenté dans ce niveau. Il permet la représentation de régions par exemple.
- **Niveau 1** : Ce niveau de détail des modèles de bâtiments peut être créé par l'extrusion de leur simple contour. Les toits sont plats et les murs ne sont pas texturés. L'échelle d'affichage de ce LoD est la ville.
- **Niveau 2** : Ce niveau ajoute des détails sur les toitures et peu appliquer des textures aux bâtiments. Ces textures peuvent être génériques ou provenir de photos des façades des bâtiments pour un rendu plus réaliste. La végétation peut également être représentée à ce niveau qui sera affiché à l'échelle des quartiers de ville.
- **Niveau 3** : Avec ce LoD on peut représenter les éléments architecturaux des bâtiments. Ainsi, les toits et les murs seront détaillés, les balcons, baies et projections apparaîtront sur les bâtiments en 3D (et non plus comme simple photo comme c'était le cas en LoD2). Des textures haute définition peuvent être appliquées aux façades. La végétation ainsi que les objets urbains sont des composants de ce LoD. Ce niveau de détail peut difficilement être généré automatiquement et sera donc produit par des logiciels de CAO. Ce LoD est affiché pour représenter l'extérieur d'un bâtiment.
- **Niveau 4** : Ce dernier niveau de détail complète le précédent en modélisant la structure intérieure des bâtiments. Par exemple, un immeuble est composé de pièces, de portes, d'escaliers et de mobilier. Ce LoD sert à représenter l'architecture interne des bâtiments.

Ces deux derniers niveaux de détail sont ceux qui composent la maquette numérique d'un bâtiment, ou BIM. Ils sont souvent créés depuis des logiciels d'architectes (hors SIG). Le format IFC permet de produire des BIM normalisées et indépendantes de l'application qui les a produites. Elle est complémentaire avec CityGML : la maquette IFC du bâtiment est très complète et contient de nombreuses informations (murs, toiture, portes, fenêtres, équipements...) là où CityGML ne représentera que la forme externe des objets selon ses différents niveaux de définition. À l'inverse, au niveau du site, les IFC représentent (de manière facultative) vaguement la

forme du terrain alors que CityGML permet l’affichage des éléments composant un site urbain. Les deux porteurs de ces normes, BuildingSmart pour les IFC et OGC pour CityGML, travaillent ensemble à l’interopérabilité des maquettes numériques des bâtiments et des territoires.

3.2.4 Le découpage (culling)

Bien que les techniques de niveaux de détail améliorent grandement les performances de rendu des scènes, tous les objets, même simplifiés n’ont pas besoin d’être affichés ni même chargés en mémoire. Les techniques de culling sont justement utilisées pour déterminer les objets à charger et constituent une autre technique de réduction de polygones. Il existe plusieurs types de culling :

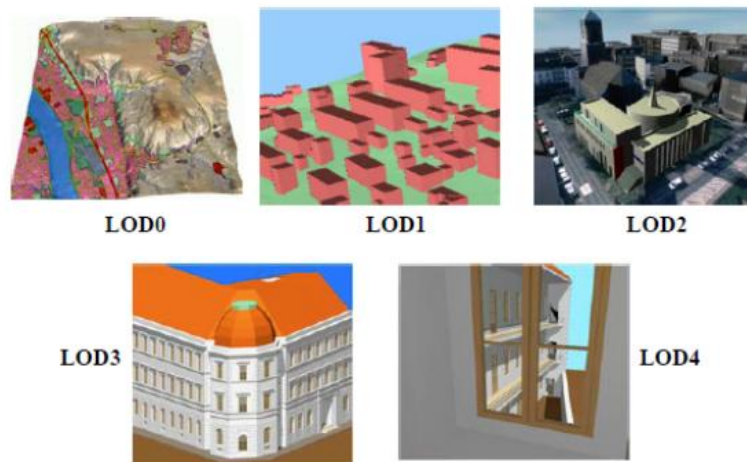


Figure 34. Les LoD dans CityGML

- le frustum culling : le plus simple et le plus couramment utilisé. On définit une zone de visibilité (dont la forme la plus courante est une pyramide tronquée) et l’on affiche dans la scène uniquement ce qui est dans cette zone. Cette technique est très efficace lorsque la scène est organisée de manière hiérarchique puisqu’il est alors possible de couper des branches entières du graphe de scène.
- l’occlusion culling vise à déterminer les objets qui ne sont pas visibles, car cachés par d’autres objets. C’est un algorithme relativement difficile à mettre en place dans les scènes globales (un extérieur par exemple). En effet, pour qu’un objet soit coupé de la scène, il doit être entièrement caché par un (ou plusieurs) autre(s). Il y a alors deux approches possibles : soit on regarde si un objet est caché au moment de le charger, soit on regarde quels objets en cachent d’autres. Quelle que soit la technique utilisée, l’occlusion culling est généralement plutôt coûteux en termes de performances, mais pour certains types de scènes avec de forts taux d’occlusion il peut se révéler d’un gain non négligeable.
- le portal culling est un type d’occlusion culling particulier. Il vise à définir un réseau de cellules reliées par des portails. Reste à définir par la suite les cellules visibles depuis un portail pour un point de vue donné. Cette technique est très efficace dans les scènes d’intérieur puisque le plus souvent les murs des pièces cachent une grande partie des autres objets extérieurs à cette pièce.

4 Normes.

La gestion technique de patrimoine urbain passe par la modélisation de l’information urbaine. Pour atteindre cet objectif, il faut combiner différents domaines de modélisation liés à l’urbain, aux villes, à l’information géographique et aux bâtiments. Cependant, il existe de nombreuses différences entre toutes ces spécialités, que ce soit au niveau de la terminologie, des normes d’information et de modélisation, de l’échelle d’intérêt, et

de la nature technique des problèmes à résoudre. Ainsi, coupler l'information de ces différents domaines pour créer un système qui répondrait à nos attentes va nécessairement créer des problèmes au niveau de l'échange d'information, ou interopérabilité. Les problèmes d'hétérogénéité représentent l'une des entraves majeures à l'interopérabilité. Ce genre de problème ne date pas d'hier. En effet, en étudiant les travaux de (Sheth, 1998) et (Nicolle, 2005), on peut distinguer cinq niveaux d'hétérogénéité entre les systèmes, en fonction des besoins d'interopérabilité plus ou moins complexes qu'ils exigent. Le premier type d'hétérogénéité, apparu dans les années 70, est celle au niveau système : comment résoudre les problèmes de communication entre matériels et systèmes d'exploitation différents. C'est l'avènement des protocoles internet qui a apporté une réponse. L'hétérogénéité syntaxique soulève, elle, les problèmes relatifs aux langages de manipulation de données. Puis ce sont posés les problèmes de l'intégration de schémas pour permettre la mise en commun d'informations provenant de sources diverses (hétérogénéité schématique) et la traduction de modèles de données (schémas, requêtes et données) pour éliminer l'hétérogénéité sémantique de différentes représentations (hétérogénéité structurelle). Comme cette présentation le suggère, l'hétérogénéité sémantique est aujourd'hui l'un des problèmes d'interopérabilité qui pose le plus de problèmes. De nombreux travaux sont effectués dans ce domaine, et particulièrement dans des thématiques liées à l'information géographique, par exemple dans (Mignard et Nicolle, 2011), (Li, 2011) et (Huang, 2011). Celle-ci est décrite par Amit Sheth comme la possibilité de relier le contenu et la représentation des sources d'information à des entités et des concepts du monde réel. Idéalement, des systèmes sémantiquement interopérables pourraient répondre de façon unifiée à des requêtes de haut niveau portant sur les concepts et ne nécessitant pas la connaissance de la structure ou du schéma de la base de données.

Dans le domaine de l'information géographique de nombreuses normes ont été proposées pour répondre au problème de l'hétérogénéité. Les normes constituent un premier moyen de lutter contre les problèmes d'hétérogénéité rencontrés dans la résolution de notre problématique. Nous donnons ici un aperçu non exhaustif des normes que nous sommes amenés à rencontrer lors de nos travaux sur l'information géographique. Ces normes, approuvées ou en cours d'élaboration, nous montrent en particulier les tendances à suivre pour élaborer un système d'information fiable et, par la suite, interopérable. Nous allons de fait nous intéresser aux normes et standards utilisés dans la gestion et la modélisation des connaissances du bâtiment, des villes, de l'information géographique et du temps.

Plusieurs organisations, consortiums d'industriels et communautés sont impliqués dans le développement des normes relatives à la modélisation urbaine.

- ISO/TC 211⁶¹ – Information géographique/géomatique est en charge des normes pour l'information géospatiale ;
- L'OGC se concentre sur les normes pour les services géographiques;
- buildingSMART Alliance travaille sur le développement de normes pour la construction et la gestion technique de patrimoine urbain;
- Le Web3D⁶² Consortium est concerné par les standards d'échange des données 3D à travers Internet;
- Open Design Alliance cherche à rendre interopérables les données CAO par l'intermédiaire d'API (Application Programming Interface, Interface de programmation des applications) libres (par exemple DWG, DGN...).
- Khronos Group⁶³ crée des standards ouverts pour la création et l'accélération du calcul parallèle et des applications graphiques;

⁶¹ International Organisation for Standardization / Technical Committee 211, <http://www.isotc211.org/>

⁶² <http://www.web3d.org/>

⁶³ <http://www.khronos.org/>

- ISO/TC 204⁶⁴ – Normalisation des systèmes de commande, de communication et d’information dans le domaine des transports de surface urbains et ruraux.
- Dans le domaine de l’information géographique et spatiale, nous retrouvons aussi, au niveau national, l’AFNOR⁶⁵ et le CEN⁶⁶ TC 287 au niveau européen.

Présentons plus en avant les deux principaux acteurs de la normalisation géographique. Depuis 1995, l’ISO/TC 211 qui élabore des normes internationales pour les données, et l’OGC, un consortium d’industriels qui développe des spécifications d’interface ordinateur, sont devenus des protagonistes de premier plan sur la scène internationale géographique. Par la suite, l’ISO/TC 211 et l’OGC ont formé un groupe de coordination des développements afin d’avoir une base commune et minimiser les chevauchements techniques. D’un côté l’OGC soumet des spécifications pour normalisation ISO via l’ISO/TC 211. L’intérêt de cette démarche est, qu’en tant qu’industriel, l’OGC a un programme d’essais et de conformité pour les normes qu’il développe. De l’autre côté, il existe un autre programme d’interopérabilité de l’OGC qui permet le développement des normes par un prototypage rapide. Nous allons maintenant décrire les différentes normes et aspects techniques que nous voulons prendre en considération pour la suite. Cette partie va s’articuler en deux temps : la description des normes de méthodologies ISO que l’on trouve dans le comité technique 211, puis les normes de spécification utilisant ces normes telles que GML, CityGML et les IFC.

4.1 Présentation des normes géographiques

Cette partie présente les normes qui ont guidé nos travaux de recherche afin d’aboutir à une solution la plus interopérable possible. Nous nous sommes en effet aperçus que, dans le milieu de l’information géographique, cette notion d’interopérabilité est omniprésente. En témoignent les nombreux organismes qui travaillent à cette cause et l’importance du groupe de travail sur l’information géographique au sein de l’ISO (Laurini, 2001). Le comité technique ISO 211 (ISO/TC 211) a développé la famille de normes ISO 19100, soit une quarantaine de normes et spécifications techniques dans le domaine de l’information géographique. Ce groupe de travail qui regroupe de nombreux pays a pour mission de développer et déployer ces normes à l’international. Le champ d’action de ISO/TC 211, tel que défini dans (ISO/TC 211, 2009a) est le suivant: Normalisation dans le domaine de l’information géographique numérique.

Ce travail vise à établir un ensemble structuré de normes pour les informations concernant des objets ou des phénomènes qui sont directement ou indirectement associés à un emplacement lié à la Terre. Ces normes peuvent spécifier, pour l’information géographique, des méthodes, outils et services pour la gestion des données (y compris la définition et la description), l’acquisition, le traitement, l’analyse, l’accès, la présentation et le transfert de ces données sous forme numérique / électronique entre les différents utilisateurs, systèmes et lieux. Les travaux doivent être liés aux normes appropriées des technologies de l’information et des données lorsque cela est possible, et fournir un cadre pour le développement des applications spécifiques à un domaine utilisant des données géographiques.

Les objectifs généraux de l’ISO / TC 211 sont les suivants:

- accroître la compréhension et l’utilisation de l’information géographique;
- accroître la disponibilité, l’accès, l’intégration et le partage de l’information géographique;
- promouvoir l’utilisation efficace (faire les choses justes), efficiente (faire juste les choses) et économique de l’information géographique numérique et du matériel et logiciels associés;
- contribuer à une approche unifiée pour aborder les problèmes mondiaux écologique et humanitaire.

⁶⁴ <http://itsa.org/industryforums/isotc204>

⁶⁵ Association Française de NORmalisation

⁶⁶ Comité Européen de Normalisation

Il nous semble donc opportun de respecter ces normes pour nos travaux. Les normes sont rangées en familles extensibles et modulaires. La Figure 35, extraite de (CNIG, 2006) nous donne un aperçu de cette organisation. Ce schéma n'est pas complet puisqu'il date de 2006 et que des normes ont été ajoutées ultérieurement. Nous allons à présent introduire les normes qui ont le plus d'intérêts pour nos travaux.

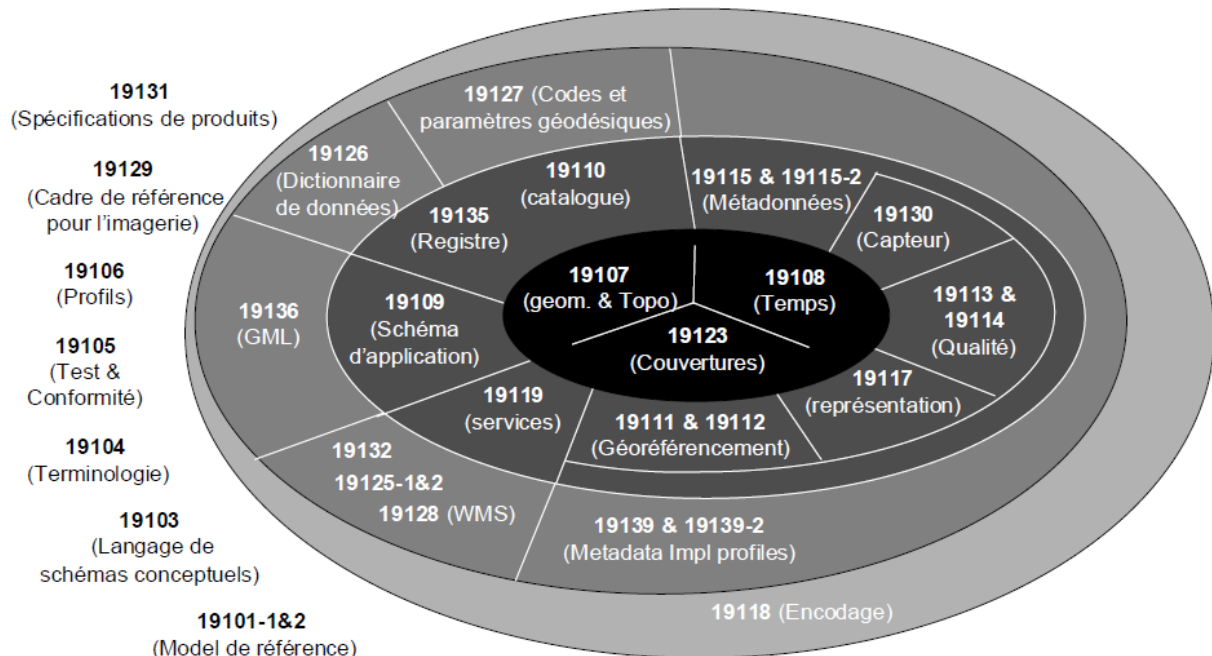


Figure 35. Les normes et spécifications techniques de l'ISO/TC211 (CNIG, 2006)

4.1.1 Les normes sur les ontologies

Nous allons commencer par lister les normes qui nous semblent intéressantes, puis dans un second temps la manière dont elles pourraient nous servir. Bien sûr, cette liste n'est pas exhaustive, dû au nombre de normes qu'il faudrait traiter, déjà rien que dans ce comité technique (56⁶⁷).

Aujourd'hui, les ontologies sont partie intégrante des bases de données et des systèmes. Cela permet une interopérabilité complète sur le Web grâce à des technologies et des concepts du Web Sémantique. La première série de normes qui nous intéressent concerne les ontologies. Les travaux en relation avec les ontologies et plus globalement le Web Sémantique dans les groupes de travail d'ISO/TC211 sont nombreux. Dans le document (ISO/TC 211, 2009b) une classification en est faite selon les aspects terminologies, règles et méthodes pour la description du contenu, description du contenu, schémas, services localisés, format commun. Nous allons lister ici celles qui nous semblent le plus importantes.

- ISO 19107:2003 - Schéma spatial
- ISO 19108:2002 - Schéma temporel
- ISO 19110:2005 - Méthodologie de catalogage des entités
- ISO 19115 - Métadonnées
- ISO 19136 :2007 - Langage de balisage en géographie (GML pour Geographic Markup Language)

⁶⁷ Source : http://www.iso.org/iso/fr/iso_technical_committee.html?commid=54904

Deux projets ont été inscrits à l’ISO fin 2010 qui méritent, dans le cadre de nos travaux, une attention particulière :

- ISO/DTS 19150-1 - Ontologie - Partie 1 : Structure
- ISO/NP19150-2 - Ontologie - Partie 2 : Règles pour le développement d’ontologies dans le langage d’ontologie Web (OWL)

À ce titre, la définition des règles et lignes directrices dans OWL pour les ontologies traitant de l’information géographique est une condition préalable. Il faudra ensuite assurer la cohérence entre les ontologies dans le domaine de l’information géographique et celles des autres domaines, ce qui permettra de simplifier le développement d’applications basées sur les ontologies. Enfin, il faudra les relier aux données affichées sur le Web provenant de sources et communautés diverses.

4.1.2 ISO 19110 - Méthodologie de catalogage des entités

Les entités géographiques sont des phénomènes du monde réel associées à un lieu sur la Terre, à l’égard desquelles des données sont recueillies, tenues à jour et diffusées. Les catalogues d’entités qui définissent les types d’entités, leurs fonctions, leurs attributs et leurs associations représentés sous forme de données géographiques sont indispensables pour convertir les données en informations exploitables. De tels catalogues d’entités favorisent la diffusion, le partage et l’utilisation des données géographiques en facilitant la compréhension du contenu et de la signification des données. À moins que les fournisseurs et les utilisateurs de données géographiques ne partagent la compréhension des types de phénomènes du monde réel représentés par les données, les utilisateurs ne pourront pas juger si les données fournies conviennent à leurs besoins.

La disponibilité de catalogues d’entités normalisées pouvant être utilisés à maintes reprises réduira les coûts d’acquisition des données et simplifiera le processus de spécification des produits pour les ensembles de données géographiques.

La présente norme internationale ISO/TS 19110:2005 (actuellement en révision) fournit un cadre normalisé d’organisation et de présentation de la classification des phénomènes du monde réel dans un ensemble de données géographiques. Tout ensemble de données géographiques constitue une abstraction grandement simplifiée et réduite d’un monde complexe et divers. Un catalogue de types d’entités ne peut jamais saisir la richesse de la réalité géographique. Il convient toutefois qu’il présente l’abstraction particulière représentée dans un ensemble de données déterminé d’une façon claire et précise et sous une forme facilement compréhensible et accessible aux utilisateurs des données.

Il existe deux niveaux d’entités géographiques : les occurrences d’entité et les types d’entité. Au niveau de l’occurrence, une entité géographique est représentée comme un phénomène discret qui est associé à des coordonnées géographiques et temporelles et peut être caractérisé par un symbole graphique particulier. Ces occurrences d’entité individuelles sont regroupées dans des classes ayant des caractéristiques communes : les types d’entité. Il est reconnu que l’information géographique est perçue de manière subjective et que son contenu dépend des besoins d’applications particulières. Les besoins d’applications particulières déterminent la manière dont les occurrences d’entité sont regroupées en types d’entité à l’intérieur d’un schéma de classification particulier.

La norme ISO 19109 *Information géographique — Règles de schéma d’application* spécifie comment les données doivent être organisées afin de refléter les besoins particuliers d’applications présentant des exigences similaires en matière de données.

Il est à noter que la description complète du contenu et de la structure d’un ensemble de données géographiques est donnée par le schéma d’application développé conformément à l’ISO 19109. Le catalogue des entités définit la signification des types d’entités et leurs attributs d’entité, fonctions d’entité et associations d’entité associés contenus dans le schéma d’application.

Les critères de collecte servant à identifier les phénomènes individuels du monde réel et à les représenter en tant qu'occurrences d'entité dans un ensemble de données ne sont pas spécifiés dans la présente Norme internationale. Du fait qu'ils ne sont pas inclus dans les normes, il est recommandé d'inclure ces critères de collecte séparément dans la spécification de produit de chaque ensemble de données.

Organiser de manière normalisée les informations d'un catalogue d'entités n'aboutira pas automatiquement à une harmonisation des applications ou à leur interopérabilité. Dans les situations où les classifications des entités sont différentes, la présente norme peut au moins servir à clarifier les différences et, de cette façon, aider à éviter les erreurs qui résulteraient de leur non-prise en compte. La norme peut également être utilisée comme cadre standard pour harmoniser les catalogues d'entité existants dont les domaines se chevauchent.

La norme ISO 19110 permet d'atteindre un niveau de granularité fin pour décrire des données géographiques qui composent un lot de données. Elle est utilisée au sein de la norme plus globale ISO 19115 qui décrit l'ensemble des métadonnées (auteurs, mode de diffusion, système de projection...).

4.1.3 ISO 19150-1 - Ontologie – Partie 1 : Structure

Cette norme constitue un nouvel élément de travail depuis fin 2010 (ISO/TC 211, 2010). Une nouvelle norme en création commence par une proposition de nouvel élément de travail (New Work Item Proposal, NWIP ou NP). Il faut ensuite environ 2 à 3 ans de travail avant sa publication.

Afin de bénéficier des développements résultant du Web Sémantique et de contribuer à l'interopérabilité de l'information géographique avec d'autres sources d'information, il devient essentiel que les modèles et les schémas des applications d'information géographiques UML de l'ISO/TC 211 soient fournis sous une forme qui est compatible avec les technologies du Web Sémantique. En tant que telle, la définition d'une structure qui spécifie l'ensemble des normes nécessaires pour faire face à l'interopérabilité sémantique en utilisant les ontologies et leurs dépendances est nécessaire. Cette partie s'appuie essentiellement sur les résumés des études de l'ISO19150 et les résultats de l'équipe du projet.

Les normes ISO pour l'information géographique (c'est-à-dire la série de normes de l'ISO/TC 211) fournissent une structure cohérente pour la description de l'information géographique en termes de schéma d'application basé sur les éléments. Ces éléments de base de la géométrie, des métadonnées, des services et autres sont bien définis. Toutefois, les normes ISO pour l'information géographique autorisent une grande liberté dans le développement et la production de données afin d'établir des catalogues d'objets variés pour mieux décrire le monde réel. Cette liberté est importante puisqu'elle permet aux normes d'être appliquées dans de nombreuses situations. En contrepartie, il est plus difficile de trouver et à la fois intégrer des données géographiques.

Le Web Sémantique a introduit le web des données. Il pourrait être considéré comme une énorme base de données ouverte dans le monde entier que les gens peuvent interroger selon leurs besoins, leur compréhension, et leur niveau d'abstraction des phénomènes du monde réel ou des événements et obtenir des réponses précises, détaillées, et appropriées, de la même façon dont les gens communiquent entre eux. Cette approche implique des capacités de raisonnement reposant sur des bases de connaissances, c'est-à-dire les ontologies. Le Web Sémantique apporte de nouvelles possibilités dans le domaine de l'information géographique pour implanter une nouvelle génération de normes qui puisse bénéficier du Web Sémantique pour réaliser l'interopérabilité sémantique de l'information géographique.

Les ontologies consistent en une représentation formelle des phénomènes d'un univers de discours (c'est-à-dire les connaissances représentables) avec un vocabulaire sous-jacent, y compris les définitions et les axiomes. Nous obtenons alors un sens voulu explicite, et la description des phénomènes et de leurs interrelations. Les données provenant de différentes disciplines, y compris l'information géographique, peuvent être intégrées et contribuer à la résolution de problèmes spécifiques (par exemple déversement de pétrole), ou plus globaux (par exemple, le changement climatique). Cette spécification technique ISO 19150-1 définit une structure de haut niveau qui encadre les normes portant spécifiquement sur la sémantique de l'information géographique par le biais d'ontologies. Les autres parties de cette structure sont les suivantes (Brodeur, 2010):

- ISO/NP 19150-2, Information géographique - Ontologie - Partie 2: Règles pour le développement des ontologies dans le langage d’ontologie Web (OWL), définit les règles et directives pour l’élaboration d’ontologies OWL, comprenant une correspondance entre des éléments de diagramme de classe UML et OWL ainsi que des règles pour décrire des schémas d’application en OWL;
- ISO 19150-3, Information géographique - Ontologie - Partie 3: les opérateurs sémantiques, définit des opérateurs sémantiques de proximité entre les concepts, ce qui vient compléter les opérateurs géométriques et temporels;
- ISO 19150-4, Information géographique - Ontologie - Partie 4: ontologie de services, identifie la structure d’une ontologie de services et définit la description des services Web pour l’information géographique dans un langage ontologique;
- ISO 19150-5, Information géographique - Ontologie - Partie 5: Registre de l’ontologie de domaine, définit internationalement le registre des ontologies de domaine de l’information géographique et de leur maintenance et fait le lien entre les concepts d’ontologies d’applications spécifiques dans un domaine donné;
- ISO 19150-6, Information géographique - Ontologie - Partie 6: Registre de l’ontologie de service, définit internationalement le registre des ontologies de services d’information géographique et de leur maintenance.

Ces parties sont complétées par les ontologies harmonisées de l’ISO/TC 211 constituant un ensemble d’ontologies OWL qui traduit et complète les modèles harmonisés de l’ISO/TC 211 développés en UML.

Après avoir regardé de plus près l’intérêt de l’ISO pour la normalisation des ontologies géographiques, intéressons-nous plus en détail à la norme consacrée aux schémas spatiaux.

4.1.4 ISO19107 - Schéma spatial

Cette norme est un ensemble de modèles conceptuels spatiaux. On l’utilise pour décrire et manipuler des données spatiales 3D ainsi que leurs caractéristiques.

Nous allons commencer cette partie par une petite définition, celle des features géographiques. C’est une notion que l’on retrouve dans de nombreuses spécifications ISO, dont notamment ISO19107 et ISO 19136 (GML). Nous parlerons parfois d’*éléments* ou d’*entités* géographiques en français. Dans ces spécifications⁶⁸, le feature est décrit comme « *une abstraction d’un phénomène du monde réel ; c’est un feature géographique s’il est associé à un emplacement relatif à la Terre* ». Une entité peut être caractérisée par ses propriétés géométriques et non géométriques. Par exemple, une route (l’entité) peut être décrite par un identifiant, son état, sa largeur, et la géométrie de son axe qui sont autant de propriétés. Les entités d’un même type sont regroupées en types d’entités. Ainsi, l’autoroute « A31 » et la « RN274 » sont deux entités du type d’entités « ROUTE ».

Dans cette norme, les caractéristiques spatiales sont décrites à l’aide d’attributs que l’on peut associer soit à un objet géométrique (dérivé de la classe GM_Object comme GM_POINT, GM_CURVE, GM_SURFACE...), soit à un objet topologique (dérivé de la classe TP_Object tels que TP_NODE, TP_EDGE, TP_FACE...), tel que décrit dans la Figure 36. Il existe également des opérations sur les données (intersection...) par le biais d’opérateurs

⁶⁸ ISO 19107 : http://www.iso.org/iso/catalogue_detail.htm?csnumber=26012

spatiaux. Ce sont des fonctions permettant de manipuler, faire des requêtes, créer, modifier ou supprimer des objets spatiaux.

La partie géométrie des objets porte sur les informations de positionnement, de taille, d'orientation et de forme du phénomène géographique, ainsi que sur des informations de dimension de l'espace dans lequel il évolue. La particularité de cette norme est que le système de coordonnées de référence choisi pour définir la position de l'objet influe sur la géométrie de l'objet. C'est d'ailleurs la seule partie de l'information géographique définissant un phénomène qui est modifiée quand le système géodésique ou le système de coordonnées est modifié. Ainsi, l'objet est traité différemment en fonction du système de coordonnées dans lequel il a été défini (François et al., 2010).

La topologie est la partie de l'information géographique qui ne varie pas lorsque l'espace est déformé de manière élastique et continue ; par exemple, si l'on se réfère à la remarque ci-dessus pour la géométrie, la topologie, elle, est invariante par transformation de coordonnées. Dans le contexte de l'information géographique, la topologie est utilisée, entre autres, pour décrire les relations de connectivité (comme l'inclusion et l'adhérence entre les objets, utilisant également la notion de voisinage) entre les différents éléments d'un graphe à n dimensions. Les relations de topologie entre des éléments géographiques peuvent être recalculées à partir de la géométrie.

4.1.5 ISO 19108 - Schéma temporel

L'ISO 19108 est un standard international qui définit les attributs, opérations, associations et métadonnées en relation avec l'information temporelle. C'est la généralisation des SIG et des applications informatiques qui ont mené à conduire des études sur les données géospatiales dans de nombreux domaines. L'information géographique comprend aussi bien les données spatiales en 3 dimensions que nous venons de voir, mais également des données temporelles, utilisées dans des applications de simulation par exemple. Le temps représente une donnée du monde physique et de nombreux concepts décrits dans les spécifications de cette norme sont également applicables hors du domaine de la géographie. L'intention du comité technique 211 de l'ISO n'est pas de développer des normes indépendantes dédiées à la gestion du temps, mais de décrire les données temporelles utilisées pour l'information géographique.

Ce schéma temporel fournit des types similaires à ceux de la géométrie et de la topologie dans la norme ISO 19107, respectivement *TM_GeometricPrimitive* et *TM_TopologicalPrimitive* (Brodeur, 2000).

4.1.6 ISO19136 - GML

L'un des résultats de la collaboration entre l'ISO/TC 211 et l'OGC est la publication d'une norme centrée sur les aspects d'implémentation de l'information géospatiale 2D et 3D : ISO 19136, correspondant au standard de l'OGC GML 3.2.1 de l'OGC pour l'information géospatiale 2D et 3D, publié en 2007. Il est utilisé pour encoder, manipuler, échanger et stocker de l'information géographique, par la description de schémas d'application (Cox et al., 2004). GML est un codage XML conforme à l'ISO 19118 :2011 qui spécifie les exigences pour la définition des règles de codage à utiliser pour l'échange de données conformes à l'ensemble de Normes internationales relatives à l'information géographique de l'ISO/TC 211^{69,70}. Du fait d'être un standard de l'OGC, GML est interopérable avec les spécifications OpenGIS® de l'OGC telles que Web Map Service (WMS) ou Web Feature Service (WFS). GML inclut à la fois la définition des propriétés spatiales et non-spatiales des éléments géographiques.

⁶⁹ ISO 19136 : http://www.iso.org/iso/catalogue_detail.htm?csnumber=32554

⁷⁰ ISO 19118 : http://www.iso.org/iso/catalogue_detail.htm?csnumber=44212

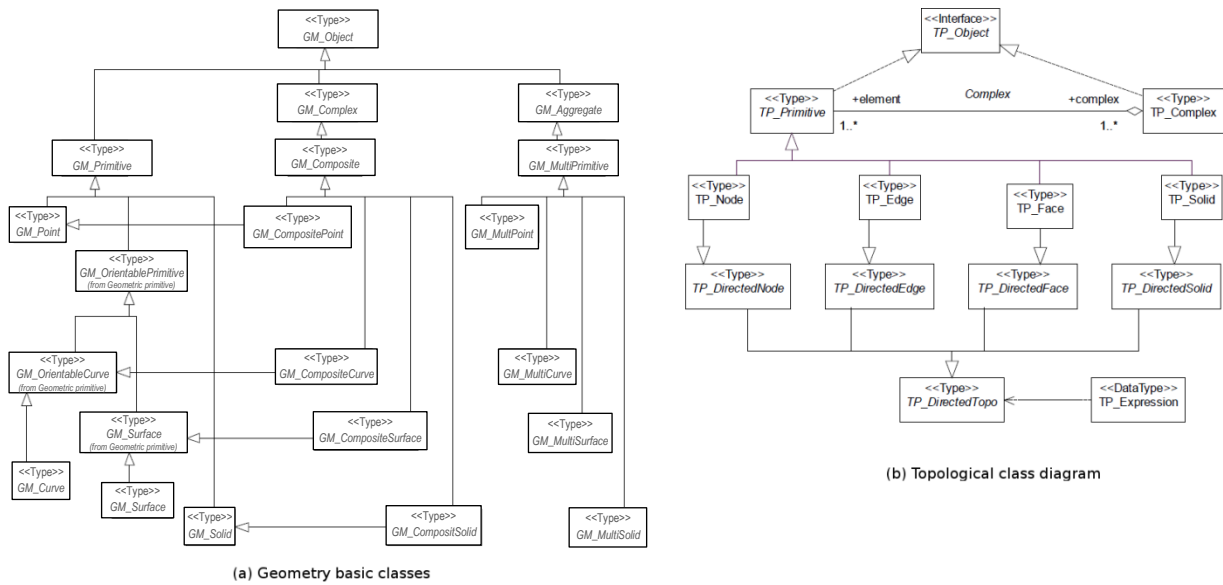


Figure 36. Les schémas conceptuels de l’ISO 19107

Dans les spécifications de GML, les features géographiques sont une notion importante et récurrente de la modélisation de l’information géographique. La définition est la même que celle utilisée dans la description de l’ISO 19107, le modèle géométrique de GML étant conforme à cette norme. Un feature correspond à la façon qu’utilise GML pour référer une chose à laquelle nous nous intéressons.

Comme nous l’avons évoqué plus tôt, les types d’entités d’une application ou domaine d’application sont précisés dans un schéma d’application. Un schéma d’application GML est spécifié dans un schéma XML qui importe le schéma GML. Il peut être construit soit par l’adhésion aux règles de schémas GML (article 21 de la spécification GML) pour la création d’un schéma d’application GML directement dans un schéma XML, soit en adhérant aux règles spécifiées dans la norme ISO 19109 pour les schémas d’application en UML, et conformément à la fois aux contraintes sur ces schémas et à la fois aux règles pour faire la correspondance aux schémas d’application GML (ISO/TC 211, 2009a). Il y a des extensions spécialisées en fonction de communautés par exemple CityGML ou SensorML^{71,72}. GML permet de définir des entités géographiques spécifiques à un domaine applicatif particulier grâce à l’utilisation de ces schémas d’application. Dans le domaine touristique par exemple, un schéma d’application contiendra les définitions des éléments géographiques tels qu’un musée ou un monument. Une autre notion importante dans GML est celle des profils. Les profils sont des restrictions logiques de GML qui peuvent être exprimées par un document, un schéma XML ou les deux. Ils servent à améliorer l’interopérabilité et réduire l’ambiguïté en n’autorisant qu’un sous-ensemble de GML. Nous pouvons citer en exemple les profils définis dans les spécifications GML et publiés ou proposés pour une utilisation publique :

- Point Profile, un profil pour les applications avec des géométries de points qui ne nécessitent pas la grammaire complète de GML ;
- GML Simple Features profile qui supporte les requêtes d’entités vectorielles et les transactions. On ne peut, en particulier, travailler dans ce profil qu’avec les trois géométries de base que sont les points, les lignes et les polygones ;
- Un profil GML pour GMLJP2 (GML en JPEG 2000) (Hayat et al., 2007)

⁷¹ SensorML sur la page de l’OGC : <http://www.opengeospatial.org/standards/sensorml>

⁷² Une liste de schémas connus et publiquement accessibles peut être trouvée à l’adresse suivante : http://en.wikipedia.org/wiki/GML_Application_Schemas

La partie géométrie occupe une place importante dans GML. C'est elle qui permet de réaliser la représentation visuelle du fichier, généralement sous forme d'une carte. Le schéma permet aussi de décrire un système géodésique en détail. Cela permet de pouvoir effectuer d'éventuelles conversions d'un système à un autre. Dans GML, une géométrie simple est caractérisée par des coordonnées en deux dimensions et une interpolation linéaire des courbes. GML fournit des éléments géométriques pour les géométries suivantes : *Point*, *LineString*, *LinearRing*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon* et *MultiGeometry*. En dehors des collections, la forme géométrique la plus complexe définie dans GML est le polygone. L'entité *Polygon* représente une aire délimitée par une frontière. Cette dernière est décrite par la propriété *exterior* dont la valeur est un élément *<LinearRing>* définissant les points de la frontière extérieure. La Figure 37 nous montre la structure de ces éléments géographiques. Ils ne sont pas tous complètement décrits afin de garder de la lisibilité dans le diagramme. De plus, en GML, les noms des éléments appelés *Exterior Boundary* et *Interior Boundary* sont *outerBoundaryIs* et *innerBoundaryIs*.

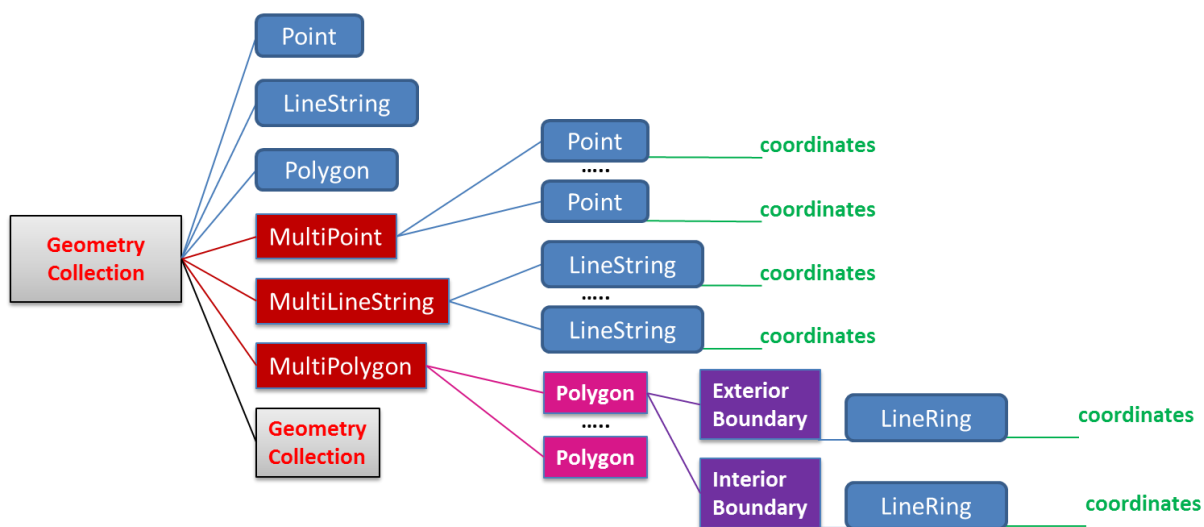


Figure 37. Structure de l'élément *geometricCollection* en GML

Les coordonnées des géométries dans GML sont décrites au moyen des balises *<coordinates>*, *<posList>* et *<pos>*. Elles sont encodées sous forme d'un ensemble de tuples séparés par des espaces. Les deux coordonnées (*x* et *y*) du tuple sont elles-mêmes séparées par des espaces. L'élément *<posList>* n'utilise que des espaces comme séparateur. L'élément *<pos>* ne peut contenir qu'un seul tuple de coordonnées. Les coordonnées d'une forme géométrique peuvent alors être décrites par l'élément *<coordinates>*, l'élément *<posList>*, ou une séquence d'éléments *<pos>*. Il existe également un élément *<Box>* permettant de délimiter une étendue en n'en spécifiant que les extrémités.

Les éléments géométriques possèdent, outre un identifiant global unique éventuel, un attribut *srsName* contenant le nom du système de coordonnées de référence utilisé. En effet, le langage GML ne spécifie pas de système de référence par défaut. Il faut donc le spécifier au moyen de cet attribut. Sa valeur doit être une URI pointant vers la définition d'un système de coordonnées de référence. Dans les collections géométriques, seul l'élément parent doit être doté d'un tel attribut.

L'exemple ci-dessous (Script 8) illustre l'utilisation des éléments *<coordinates>* et *<pos>* pour spécifier les coordonnées d'un point. L'attribut *srsName* fait référence au système de référence EPSG:4171 qui correspond au réseau géodésique français 1993 (RGF93) avec des coordonnées sous forme de (latitude, longitude) exprimées en degrés, le méridien de Greenwich étant le méridien de référence.

```
<Point gid="P1" srsName="http://www.opengis.net/gml/srs/epsg.xml#4171">
```

```

<gml:coordinates>13.37, 71.16</gml:coordinates>
</gml:Point>

<Point gid="P2" srsName="http://www.opengis.net/gml/srs/epsg.xml#4171">
<gml:pos>52.1 43.84</gml:pos>
</gml:Point>
    
```

Script 8. Exemple de définition de géométries avec GML.

Nous pouvons également évoquer la topologie spatiale dans GML. Celle-ci est basée sur la description des liens de connexion entre les primitives (*Node, Edge, Face, and TopoSolid*). Contrairement à ce que nous venons de voir pour la géométrie, la topologie n’utilise pas de coordonnées. Cependant, les primitives de la topologie GML peuvent avoir une représentation géométrique (avec respectivement *Point, Curve, Surface, and Solid*) en utilisant les objets et propriétés GML *TopologyPrimitive/property/GeometricPrimitive* (OGC, 2006).

Enfin, la temporalité étant une notion répandue dans les SIG et sur laquelle de plus en plus d’intérêts sont portés, il n’est pas étonnant de la voir apparaître dans GML3. Il y a un schéma (temporal.xsd schéma) qui définit les objets GML temporels clés : *TimePeriod* et *TimeInstant*. Suivant les recommandations de la norme ISO 8601

Pour résumer cette partie, nous pouvons retenir que les différents éléments du langage que nous venons de voir reposent sur des schémas applicatifs qui vont composer les modules de GML. La Figure 38 nous montre les relations entre GML et les autres normes ISO dont nous avons parlé plus tôt.

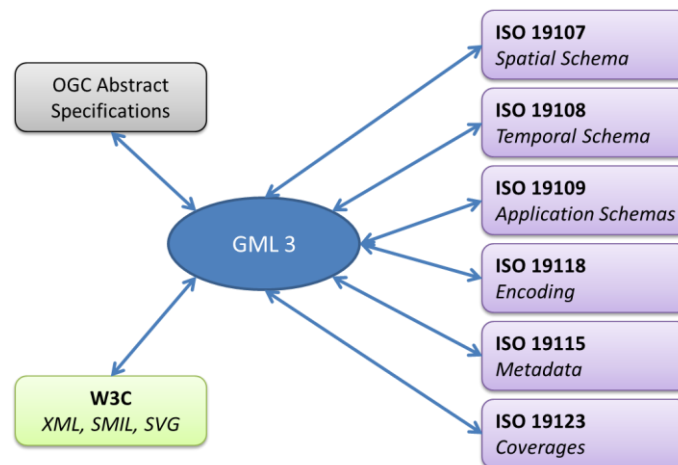


Figure 38. GML et les normes

4.2 Conclusion

Les différentes normes que nous venons d’étudier apportent une réponse partielle à la problématique de l’hétérogénéité rencontrée dans l’objectif de nos travaux. Une fois que l’on a fait le tour des normes intéressantes dans le domaine de la modélisation géographique, il est intéressant de voir la manière dont on peut les utiliser afin de construire une solution la plus respectueuse possible. Pour cela, nous allons utiliser des langages de description des données du bâtiment et de l’information de la ville qui respectent, dans leur domaine, tout ou partie de ces normes : IFC et CityGML.

On peut faire une petite remarque sur le fait que, malgré les efforts effectués pour apporter une normalisation sur les ontologies dans le domaine géographique, il n’y a pas de collaboration directe entre le comité technique ISO/TC 211 et le W3C, alors que ce dernier collabore avec d’autres comités techniques de l’ISO (comme ISO/IEC JTC 1 pour les technologies de l’information par exemple) (Lopez et al., 2011). Par contre, pour nuancer cette

remarque on peut noter le fait que l'OGC fait partie depuis 2007 du W3C et crée ainsi un lien indirect avec le comité technique pour l'élaboration de ces normes.

Pour conclure sur cette partie, nous pouvons citer quelques normes dont nous n'avons pas parlé, mais que nous prenons tout de même en considération. Il y a par exemple la norme ISO 19111:2007 (Système de références spatiales par coordonnées) qui va nous servir pour la définition et l'intégration des systèmes de référence géodésiques dont nous avons besoin pour notre plateforme. Il y a également la norme ISO 8601 :2004 (Éléments de données et formats d'échange – Échange d'information – Représentation de la date et de l'heure) qui va nous intéresser. En effet, dans l'objectif d'internationaliser la plateforme Active3D, il est important de respecter une norme internationale dans l'écriture des dates et de l'heure afin d'éviter tout risque de confusion dans les communications internationales dû au grand nombre de notations nationales différentes. De plus, la flexibilité de cette norme nous permet de ne pas utiliser tous ces éléments, ceux-ci étant répartis en six niveaux de granularité (par exemple 2012-12-21T01:00:00-06:00 correspond au 21 décembre 2012, à 1h du matin heure normale de l'est de l'Amérique Centrale (soit 6 heures de décalage) et 2012-12-21T07:00:00Z correspond au même instant).

5 La multireprésentation dans les SIG

La multireprésentation des données dans les SIG est un vaste domaine et englobe plusieurs significations. Nous allons présenter celles qui nous semblent les plus pertinentes dans notre étude. De manière générale, la multireprésentation désigne la multitude des aspects sous lesquels la réalité géographique peut intéresser les personnes à un instant donné. Une même portion de territoire est perçue de façons très différentes par des gestionnaires de patrimoines immobiliers, des économistes, des agriculteurs, ou encore des touristes planifiant leurs vacances. La quantité et la nature de l'information, la précision et le mode de représentation cartographique varient selon le profil (Moisuc et al., 2004). Par exemple, la représentation d'un site urbain peut demander beaucoup de détail pour un architecte créant un nouveau projet (forme et emplacement exact, utilisation, nature du terrain...) et une description plus sommaire pour une personne travaillant dans le domaine des risques hydrologiques (inondabilité, vulnérabilité).

On peut distinguer deux approches dans le phénomène de la multireprésentation : les sources et les applications. Les principales sources de représentations multiples dans les SIG sont la prise en compte d'échelles multiples, la prise en compte de différents points de vue et la prise en compte de temps (Moisuc et al., 2004). Parmi les types d'applications multireprésentations (Devogele, 1997) en a cités sept que nous pouvons compléter avec des applications apparues ces dernières années comme la 3D et les ontologies (Mao, 2011) (Culot et al., 2003):

- les trois premiers types répondent aux besoins des utilisateurs : cartographie électronique, analyse, navigation, 3D et ontologies
- les quatre suivants sont plus spécifiques aux producteurs de bases de données géographiques : contrôle de cohérence, dérivation de bases de données ayant une représentation hétérogène, propagation des mises à jour et serveur de données multireprésentation.

Nous allons étudier plus en détail ces différentes approches.

5.1 La multireprésentation en cartographie

La gestion des représentations en cartographie est essentiellement liée à trois types d'information : la gestion de la résolution, la gestion des thèmes et les informations de mise à jour. Pour pouvoir générer des cartes de différentes échelles, il est nécessaire de stocker des données correspondant à plusieurs résolutions : une résolution par intervalle d'échelle à représenter, et ceci pour chaque territoire étudié lorsque cela est nécessaire. Dans le cas des cartes papier, cela revient à stocker posséder plusieurs cartes d'une même zone avec des représentations différentes et faire soi-même le rapprochement entre ces cartes. L'idéal serait d'être

capable de stocker uniquement les données correspondant au niveau de détail le plus fin, et d’en déduire, par généralisation, les données moins détaillées au moment de la requête. Cependant, le processus de généralisation cartographique (obtention des cartes à des échelles plus petites à partir des cartes à des échelles plus grandes) n’est pas entièrement automatisable et de plus, dans les cas où cela s’avère possible, il s’agit d’un processus très coûteux en temps de calcul (Davis et Laender, 1999). La solution largement utilisée aujourd’hui consiste à stocker explicitement plusieurs résolutions de données pour une même zone. Cela peut se faire de deux manières : en réalisant une base de données par résolution ou en utilisant une seule base stockant plusieurs résolutions. La Figure 39 montre un exemple de différentes représentations d’une même zone selon différentes échelles (généralisation cartographique). Les données proviennent du Géoportail et les échelles ne sont pas forcément habituelles comparées aux versions des cartes papier que l’on a l’habitude de manipuler. Ainsi, plusieurs représentations des mêmes données à des échelles différentes et selon différents thèmes et à différentes époques doivent pouvoir être gérées. Les critères de sélection cartographique pour afficher les informations cohérentes sont, d’après (Devogele, 1997) :

- Les circonstances d’utilisation
- La densité d’information
- La catégorie de l’usager
- L’intérêt
- La date
- La distance entre les objets.

Ces critères sont décrits dans notre architecture de modélisation d’environnement urbain sous la forme de contexte d’utilisation, explicitement (circonstance, date) ou implicitement (densité, distance).

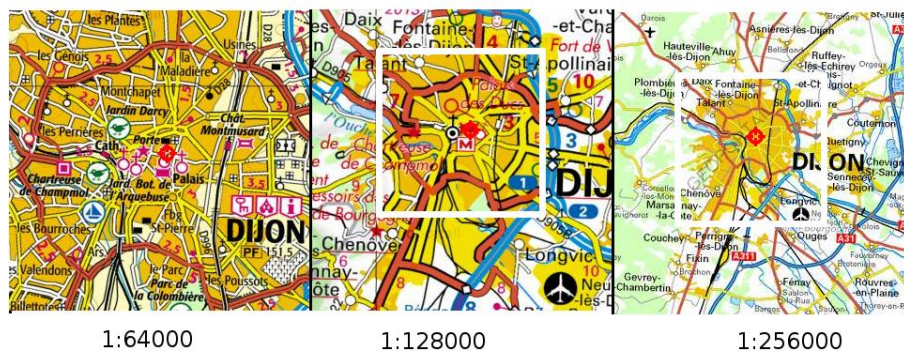


Figure 39. Exemple de généralisation cartographique (données issues du Géoportail)

Une majorité de SIG stockent l’information par résolution. Un des problèmes rencontrés pour notre étude est qu’en passant d’une résolution à l’autre, il n’y a aucune liaison entre l’identité des objets représentés (manque de sémantique). Cela pose certains problèmes pour les utilisateurs finaux (une analyse combinant plusieurs résolutions devient très difficile), mais aussi pour les concepteurs, en termes de propagation des mises à jour et de maintien de la cohérence de données. Si l’on prend en exemple une parcelle stockée dans plusieurs bases de données sur laquelle on souhaite construire un ouvrage, ce changement doit être introduit dans chaque base de données. De plus, le mécanisme de vérification de la propagation des modifications et de vérifications de leur cohérence est complexe à mettre en place.

Une autre solution, plus efficace et plus expressive, consiste à stocker une seule instance dans la base de données pour chaque objet du monde réel. Un SIG fondé sur une telle approche permet un plus de flexibilité et d’efficacité, mais il doit s’appuyer sur une structure de données capable de résoudre plusieurs problèmes (Moisuc et al., 2004) :

- La gestion de géométries multiples

- la gestion de plusieurs niveaux d'abstraction et de la visibilité des objets ;
- la gestion de multiples niveaux d'abstraction dans la partie attributaire des objets ;
- l'introduction du temps dans les SGBD a pour but de représenter la réalité en différents points de l'axe temporel ;
- la gestion de points de vue multiples.

5.2 La multireprésentation dans les ontologies

Nous commencerons par présenter dans ce paragraphe ce qui définit une ontologie, et notamment le concept d'ontologie géographique, un cas particulier des ontologies dynamiques. Puis nous présenterons un peu plus en détail le modèle MADS, une approche qui a été utilisée pour la multireprésentation sémantique. Enfin, nous essaierons de mettre en avant les concepts que nous pouvons réutiliser pour la construction d'une ontologie dynamique appliquée à la description d'un UIM.

5.2.1 Ontologie géographique / évolutive

Il existe de fait différents types d'ontologies dont nous pouvons trouver une rapide description dans l'article (Cullot et al., 2003). Cet article traite plus particulièrement des ontologies dans le domaine géographique. Il y est notamment expliqué que le terme d'ontologie peut désigner, entre autres et de façon *concrète*, une liste de termes reliés entre eux par des liens sémantiques tels que synonymie, hyponymie ou hyperonymie, ce qui définit un « thésaurus » (ontologie de haut niveau). Mais il peut également désigner une structure plus complexe s'apparentant à un schéma *conceptuel* de base de données, que l'on appelle « ontologie descriptive ». C'est avec ce type d'ontologie que l'on peut construire des ontologies de domaines ou de tâches en utilisant des concepts et des relations sémantiquement riches, ceux-ci pouvant être représentés dans des langages à base de logiques de description (Wache et al., 2001).

Il reste maintenant à définir la manière de conceptualiser les informations qui, au sein de l'ontologie, évoluent en fonction du temps et de l'espace. Ces informations ne sont pas naturelles à modéliser (par exemple il n'existe pas à l'origine de facilité pour modéliser des informations spatiales en Logique de Description (LD)) malgré le besoin qui s'en fait ressentir, notamment dans les domaines des systèmes d'information géographique. Nous allons nous baser pour ce faire sur une ontologie d'application, dite géographique, où la notion d'espace est la plus naturelle. Commençons par définir les besoins inhérents en termes de conceptualisation des ontologies géographiques. Leur nature d'ontologie d'application en font des structures composées qui recouvrent des ontologies de domaines et d'activités géographiques, basés sur un haut niveau qui permet de décrire spécifiquement des concepts qui caractérisent l'espace (comme le point, la ligne...) et où ceux-ci sont spatialisés, c'est-à-dire localisés dans l'espace et souvent le temps. Les besoins pour de telles ontologies sont, d'après (Cullot et al., 2003) :

- de pouvoir définir la spatialité à l'aide de types de données spatiales variées (ligne, point, surface simple...), de types d'objets spatiaux (i.e. d'objets ayant des attributs spatiaux), de relations spatiales comme les relations topologiques et/ou des champs continus (raster).
- définir intentionnellement des concepts spatiaux à l'aide d'axiomes contenant des prédicats spatiaux,
- raisonner sur la spatialité des instances, i.e. inférer à partir des relations spatiales décrites l'ensemble des relations spatiales valides.

Le prochain paragraphe présente une approche qui permet de résoudre la problématique de la conceptualisation de la spatialité dans des ontologies comme celle décrite ci-dessus ; il s'agit de MADS, un

modèle conceptuel relationnel qui donne les bases de la multireprésentation sémantique évoquée dans nos objectifs.

5.2.2 MADS et ses applications aux ontologies géographiques

Le modèle de données MADS (Spaccapietra et al., 2002) vise à définir une approche pour manipuler des bases de données géographiques dans lesquelles une même information peut avoir plusieurs représentations en fonction du temps et de l’espace.

L’élaboration du modèle conceptuel MADS a été guidée par les objectifs suivants, détaillés dans (Parent et al., 1998) :

- l’orthogonalité des dimensions structurelle, spatiale et temporelle (c’est-à-dire que le choix d’un mode de représentation pour l’une des dimensions n’influence pas les autres)
- la possibilité de décrire des relations topologiques ou temporelles entre les entités de manière explicite,
- une définition formelle des concepts tels l’agrégation spatiale ou temporelle, la généralisation/spécialisation et les mécanismes d’héritage associés,
- la possibilité de décrire des champs spatiaux continus,
- la provision de types spatiaux génériques au-delà des types de base, des notations visuelles intuitives.

MADS est un modèle conceptuel relationnel (entité - association) étendu aux concepts principaux de l’approche orientée objet (structures complexes, héritage et méthodes). Il propose un ensemble de quatre composantes pour résoudre ces différentes problématiques (structurelle, spatiale, temporelle et multireprésentation). Nous allons voir comment l’on peut intégrer ce mécanisme de représentation au sein d’une ontologie.

L’extension de MADS décrite dans (Vangenot, 2004) permet à un phénomène d’avoir plusieurs représentations au sein d’une même base de données. La notion de multireprésentation est apparue dans le domaine des Systèmes d’Information Géographique, dans le but de gérer plusieurs représentations différentes des mêmes objets géographiques selon différents points de vue ou selon différents niveaux de résolution. Dans le domaine de la modélisation d’information, le problème de la multireprésentation est connu depuis longtemps au travers de l’utilisation des vues dans les bases de données pour personnaliser l’accès d’un groupe d’utilisateurs. On peut supposer qu’une des origines de la représentation multiple vient du caractère subjectif de la modélisation des bases de données. Mais l’information peut également être une entité sujette à différentes représentations selon le contexte dans lequel elle est considérée.

Dans MADS, les représentations multiples sont utilisées dans deux cas d’utilisation : le point de vue et le contexte multirésolution. Un point de vue matérialise un intérêt spécifique dans la gestion des données et détermine une représentation de ces données lui correspondant. Il permet de sélectionner un sous-ensemble de la base avec des éléments intéressants pour un groupe d’utilisateurs en se basant sur les types d’objets ou d’associations et leurs propriétés. La résolution, quant à elle, permet d’ajuster de façon dynamique l’accès aux données afin de sélectionner les représentations appropriées à un certain niveau de détail. Les facettes de multireprésentation, le point de vue et la résolution sont introduits dans le modèle de données sous la forme d’estampilles de représentation. Pour gérer ces deux mode de sélection, on peut définir des estampilles <point de vue, résolution> pour les différents éléments, où le domaine de valeur pour le point de vue est discret, énuméré et non ordonné et celui de la résolution est continu, linéaire et ordonné (de façon similaire au temps).

MADS propose deux approches pour décrire les représentations multiples, chaque représentation étant identifiée par le mécanisme des estampilles. On peut donc soit regrouper toutes les représentations dans une structure unique (approche par intégration de représentation), soit effectuer une mise en correspondance des représentations à l'aide d'associations dites de correspondance (les liens traditionnels de multi-instanciation, les liens is-a et may-be, l'association d'agrégation, qui traduit la décomposition d'un objet en ses composants, l'association d'identité, qui lie des objets de même identité, et l'association SetToSet, qui relie deux ensembles d'objets décrivant collectivement le même objet du monde réel).

L'adaptation des principes de MADS aux ontologies géographiques a donné lieu à plusieurs travaux de recherche (Cullot et al., 2002), (Sotnykova et al., 2005) et (Benslimane et al., 2003). Ceux-ci visent essentiellement à recréer la technique d'estampillage des données pour représenter les différents contextes. Le contexte est utilisé pour définir des profils utilisateurs dans l'ontologie et permettre la sélection d'une sous-ontologie (i.e. un sous ensemble de l'ontologie).

Dans un premier temps, le principe consiste à définir, pour un concept, un ou plusieurs contextes, ou chaque représentation d'un concept est estampillée. Le Script 9 nous donne un exemple de définition d'un concept avec deux contextes. Ainsi, pour définir un concept, on donne en paramètre l'ensemble des estampilles, puis pour la définition des attributs, on indique une liste d'estampilles auxquelles ils se rapportent.

```
s1, s2 sont des estampilles qui correspondent respectivement aux contextes
Road trafic et Driving school
```

```
Type Vehicle (s1, s2)
s1 : Speed (1,1) : number,
s1 : VehicleType (1,1) : string,
s1, s2 : LicencePlate (1,1) : string,
s1, s2 : AssurancePolicy (1,1) : string,
s1: RegistrationDate (1,1) : date,
s2 : RegistrationDate (1,1) : string,
s1: Driver (1,1) : string,
s2 : CarModel (1,1) : string
```

Script 9. Exemple de définition d'un concept estampillé, d'après (Benslimane et al., 2003)

Ensuite, pour définir la multireprésentation des concepts complexes (dérivés de concepts atomiques par l'application de constructeurs des logiques de descriptions), il a été fait le choix d'étendre la grammaire des LD classiques (i.e. exprimé avec le langage de description *ALCN*, défini dans (Baader et al. 2003)) avec des constructeurs contextuels. Ils ont été définis dans le script suivant (Script 10) :

```
On considère s1,...,sm un ensemble d'estampilles, R un rôle et C et D des
concepts complexes contextuels que l'on peut créer au moyen de la syntaxe
suivante :
```

```
C, D → ∃ s1,...,sm R.C (quantification existentielle contextuelle)
| ∀ s1,...,sm R.C (restriction de valeur contextuelle)
| (≤ s1,...,sm nR) (cardinalité maximale d'un rôle contextuel)
| (≥ s1,...,sm nR) (cardinalité minimale d'un rôle contextuel)
| C ∩ s1,...,sm D (conjonction contextuelle)
```

Script 10. Extension de la grammaire des Logiques de Description pour la multireprésentation sémantique

Une partie des travaux cités plus tôt est plus orientée sur la définition de la temporalité et de la spatialité et vise à étendre les logiques de descriptions. Pour se faire, il a été proposé de séparer les ensembles d'objets définis en deux groupes : celui des objets abstraits, et celui des objets concrets (comme les nombres, chaînes,

les spatiaux ou les temporels). Les objets abstraits peuvent être liés à d’autres objets abstraits par des rôles abstraits ou à des objets concrets par des rôles concrets. Le but est donc de définir des rôles concrets particuliers entre les objets pour définir la spatio-temporalité. En appliquant ses rôles à un domaine, il est possible de décrire les relations topologiques et de synchronisation entre les objets (rôles *hasArea* et *hasDuration*). Le domaine concret T pour l’aspect temporel est basé sur les intervalles de temps et sur les relations de Allen (Allen, 1983) (*précède, succède, égale, rencontre, chevauche, pendant, commence et finit*), le domaine concret spatial S est lui défini pour les polygones par l’utilisation de prédicats de base lié à la topologie (*Disjonction, adjacence, recouvrement, inclusion stricte et tangentielle, égalité*), présentés dans la Figure 40.

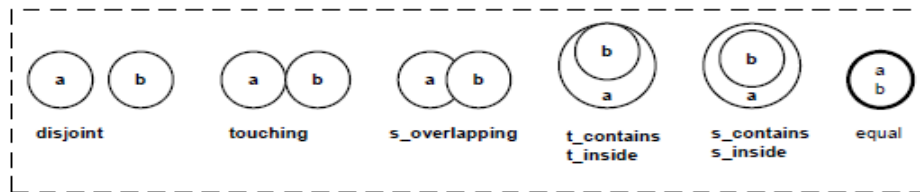


Figure 40. Relations topologiques

Pour illustrer la définition d’objets spatio-temporels nous pouvons reprendre un exemple tiré de (Sotnykova et al. 2005) qui vise à définir la notion de Musée comme un ensemble de monuments. Pour cela, on décrit le fait qu’un Musée est spatialement connecté à des monuments, et que les horaires d’ouverture se chevauchent. On définit alors un premier prédicat spatial, *connected*, comme la disjonction des différents prédicats spatiaux de base, un rôle, *spatial_connected*, basé sur le prédicat que l’on vient de définir, et un rôle temporel, *duration_overlaps*, comme dans le Script 11.

```
connected ≡ touching v s_overlapping v t_contains v t_inside v s_contains v
s_inside v equal ;
spatial_connected ≡ ∃(hasArea) (hasArea).connected ;
duration_overlaps ≡ ∃(hasDuration) (hasDuration).overlaps ;
MuseumMonument ⊆ Museum Π
    ∃spatial_connected.Monument Π
    ∃duration_overlaps.Monument ;
```

Script 11. Exemple de relations topologiques

5.2.3 La multireprésentation en 3D

Pour conclure la partie de l’état de l’art sur la multireprésentation dans les SIG, nous allons étudier les mécanismes existant dans le domaine de la 3D. Nous avons déjà abordé certains mécanismes sans les citer explicitement lorsque nous avons parlé des niveaux de détails. Ainsi, nous avons défini à la fois des mécanismes de représentation multiple par simplification des géométries en fonction de la distance (LoD de CityGML), et à la fois sur le regroupement d’entités avec les H-LoD.

Si CityGML semble être une approche complète pour modéliser l’environnement urbain d’un point de vue géométrique, topologique et sémantique, il n’est pas fait pour l’affichage des données comme le souligne (Kolbe, 2009) : « CityGML est complémentaire avec les normes de visualisation comme X3D ou KML. Alors que ces modèles sont axés sur la présentation, le comportement et l’interaction des modèles 3D, CityGML s’occupe principalement de l’échange d’informations en milieu urbain à l’aide d’objets 3D ».

Certains travaux comme (Mao, 2011) viennent pallier ce manque en proposant des mécanismes de multi-représentation 3D permettant de créer des arbres d'objets urbains (CityTree) afin d'adapter des mécanismes de généralisation classique vues plus tôt (le regroupement d'objets par exemple). Il est ainsi possible de calculer des représentations pour des nœuds de l'arbre qui sont sélectionnés en fonction du point de vue de la caméra. Dans l'arbre, les objets sont regroupés par bloc géographiquement proche, par exemple un ensemble de bâtiments séparés par une route. Cette méthode, purement géométrique, permet d'optimiser le rendu d'un fichier CityGML et d'obtenir des résultats satisfaisants lors du rendu de larges scènes.

Enfin, le dernier mécanisme de multireprésentation que nous allons présenter est un classique des logiciels de CAO 3D : l'association de représentations 2D et 3D aux entités (Cruz et Nicolle, 2006)(Nebiker, 2002). Ainsi, dans Active3D par exemple, chaque objet est représenté par sa boîte englobante 3D, une représentation 2D et sa représentation réelle 3D. Certains systèmes comme DILAS (Nebiker, 2002) calculent les deux premières représentations depuis le modèle complet en 3D, et ne représente que le contour des objets en 2D, d'autres, comme dans les IFC, peuvent associer une représentation 2D complète et indépendante de la 3D. Les deux premières représentations (boîte englobante et empreinte 2D) sont utilisées essentiellement pour accélérer les requêtes géométriques.

6 Conclusion

Le projet SIGA3D qui s'inscrit dans la continuité du projet Active3D a pour but de compléter l'architecture de modélisation des bâtiments existante au sein de la plateforme Active3D afin de pouvoir effectuer une modélisation d'objets urbains. Ces objets urbains sont des éléments extérieurs au bâtiment tels que les réseaux physiques (route, réseau gaz, téléphone...) et leurs composants (valve, tuyaux, coude...). L'un des objectifs de cette extension est le respect des normes que nous utilisons telles les IFC pour le bâtiment. Pour la modélisation extérieure du bâtiment, d'autres normes issues notamment de l'Open Geospatial Consortium apparaissent actuellement et sont en constante évolution. Pour reprendre les propositions de schéma de (Ferries et Légise, 2006) présentés au tout début de ce chapitre, nous pouvons dans le même esprit proposer une architecture faiblement couplée (Figure 41). Cette architecture permettra à un méta-niveau de représenter les meilleures propriétés de chacun des domaines pour construire une plateforme de gestion technique de patrimoine urbain cohérente.

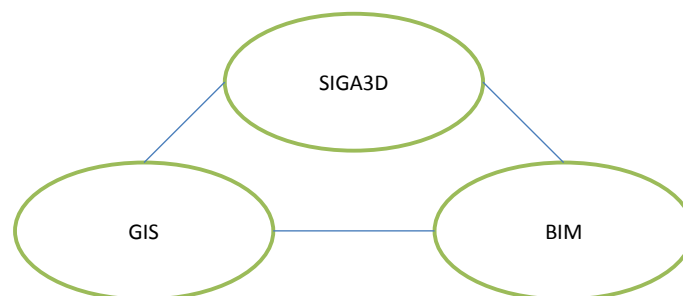


Figure 41. Couplage GIS et BIM avec SIGA3D

L’ensemble des objectifs du projet SIGA3D présenté en introduction de cette thèse (Figure 1) peuvent être regroupés par catégories : la *géométrie*, la *sémantique*, la *spatialité* et la *temporalité*, les *connexions*, et enfin la *topologie*, à la fois spatiale et temporelle.

Le Tableau 3 présente une synthèse des besoins auxquels répondent les différents systèmes de représentation de connaissances. Les lignes correspondent aux domaines sur lesquels nous devons agir, et les colonnes rappellent les différentes familles de systèmes de modélisation. Pour chaque caractéristique et système de modélisation, trois valeurs sont possibles. Un cercle rouge signifie que le système de modélisation ne répond pas à une caractéristique, ou du moins de manière trop basique pour l’objectif visé. Un cercle vert informe que le système répond de manière exhaustive à la caractéristique. Un cercle orange indique que le système ne répond pas directement aux besoins, mais propose un cadre qui permettrait d’y répondre. Le tableau est inspiré à la fois des travaux de (Vanlande, 2007) et de (Kolbe, 2009).

	<i>Intelligence Artificielle</i>	<i>3D</i>		<i>Génie logiciel</i>	<i>Web</i>	<i>SIG</i>				<i>BIM</i>	
	Logiques de description	X3D	Collada	MADS	RDF/OWL	Modèles 3D géométriques	Modèles topologiques	KML	CityGML	IFC	C-DMF
<i>Sémantique</i>	●	●	●	●	●	●	●	●	●	●	●
<i>Géométrie</i>	●	●	●	●	●	●	●	●	●	●	●
<i>Spatialité</i>	●	●	●	●	●	●	●	●	●	●	●
<i>Temporalité</i>	●	●	●	●	●	●	●	●	●	●	●
<i>Connexions</i>	●	●	●	●	●	●	●	●	●	●	●
<i>Topologie</i>	●	●	●	●	●	●	●	●	●	●	●

Tableau 3. Synthèse des approches par catégories

La première remarque qui s’impose est qu’aucun des formalismes étudiés ne propose une réponse globale à l’ensemble des besoins du projet. Il apparaît alors que la solution finale sera basée en partie sur un assemblage de certains d’entre eux. Les logiques de description, les graphes conceptuels de Sowa basés sur une structure de réseau sémantique et sur la logique de premier ordre, permettent de réaliser des inférences au niveau terminologique et assertionnel. Ces modèles combinent la syntaxe pour décrire les données et les processus permettant de réaliser des inférences. Il est possible de définir des modèles de données par la description de classes et de propriétés. Ces formalismes sont utilisés dans le domaine de l’intelligence artificielle, par exemple dans le traitement des langues automatiques.

X3D est un standard libre de format de fichier et d’architecture d’exécution pour représenter et échanger des scènes et des objets 3D avec XML (ISO/IEC 19775-1:2008). C’est une norme ISO qui fournit un mécanisme pour le stockage, la récupération et la lecture en temps réel de contenus graphiques intégrés dans des applications, le tout dans une architecture ouverte afin de supporter un large éventail de scénarii de domaines et d’utilisateurs.

COLLADA a pour but d’établir un format de fichier d’échange pour les applications 3D interactives. Il définit un standard de schéma XML ouvert pour échanger les acquisitions numériques entre différents types d’applications logicielles graphiques qui pourraient autrement conserver leur acquisition dans des formats

incompatibles. Les documents COLLADA, qui décrivent des acquisitions numériques, sont habituellement identifiés par leur extension .dae («digital asset exchange», traduit par «échange numérique d'acquisition»⁷³). Le domaine du génie logiciel offre de nombreux outils informatiques éprouvés et des solutions pour la conception d'applications industrielles. Le génie logiciel va au-delà de nos préoccupations et s'attache entre autres, à développer des méthodes pour concevoir et gérer les applications durant leur cycle de vie. Ces formalismes proposent une partie pour décrire de l'information, mais également pour la traiter. Nous nous intéressons ici seulement à la partie modélisation. L'approche relationnelle se propose de décrire les connaissances avec des entités et des relations entre ces entités. Elle est très utilisée pour concevoir les modèles conceptuels de bases de données relationnelles. MADS est à l'origine une extension de ces systèmes qui supporte les dimensions spatiales, temporelles et topologiques dans le but de gérer la multireprésentation des données.

Dans le domaine des SIG, et plus particulièrement dans les approches de SIG 3D, nous avons retenu deux modèles principaux pour la modélisation des données : le modèle topologique que nous avons décrits plus tôt, et les modèles géométriques tels que l'on peut les retrouver dans (De La Losa, 2000) ou (Ramos, 2003). Ensuite, KML est un format en XML créée par Google, pour la visualisation dans Google Earth et peut-être généré dans Sketchup⁷⁴. Il utilise le format COLLADA. Il est de plus en plus pris en compte par les SIG (MapGuide, ArcGIS, MapInfo...). Quant à CityGML dont nous avons plusieurs fois parlé, il est souvent associé à des outils de représentations 3D comme COLLADA (Kolbe, 2009) du à ses faiblesses en modélisation géométrique. Le nombre de travaux sur ce sujet que nous avons évoqué précédemment montre l'engouement pour ce système de modélisation et l'on peut voir effectivement qu'il est, pour notre objectif, celui qui apporte le plus de réponses. Cependant, même si les possibilités offertes en termes de modélisation sémantique sont importantes, il reste aujourd'hui beaucoup de travail à réaliser pour pouvoir prendre en compte une expressivité ontologique équivalente aux IFC pour la modélisation des bâtiments. De plus, la technologie est encore très peu utilisée en milieu professionnel et le manque d'outils spécialisés basés sur ce formalisme sont aujourd'hui un frein à son adoption, du moins dans le cadre d'une architecture basée uniquement sur celui-ci.

Enfin, dans le domaine du BIM, nous ne pouvons pas passer à côté du système de modélisation C-DMF qui est à la base de la plateforme industrielle Active3D actuelle. Si ce système convient parfaitement à la modélisation des bâtiments, il trouve en revanche très vite ses limites dans la gestion d'objets urbains. Ainsi, les notions de géoréférencement, de gestion du temps et de relations topologiques sont très peu présentes (elles ne peuvent être décrites que sous forme d'étiquette dans la couche de gestion du contexte et ne possèdent aucun système de traitement dédié).

Avec ces différents éléments en notre possession, nous avons choisi une approche permettant de coupler les avantages de ces différents formalismes de modélisation au sein d'une même structure. De plus, la contrainte industrielle étant forte sur ce projet, nous avons décidé d'adapter le modèle existant pour le sortir de sa spécialisation sur les bâtiments et élargir son champ d'action à l'information géographique. Nous allons donc faire évoluer C-DMF pour en faire une architecture d'intégration sémantique. Nous entendons par là que l'architecture SIGA3D aura pour objectif de stocker l'information dans des ontologies dynamiques et évolutives, peuplées à partir non seulement de la plateforme Active3D et des outils dédiés au Facility Management, mais également à partir des autres formalismes décrits dans ce document, que l'information traitée soit liée aux bâtiments, aux objets urbains ou même à la cartographie.

⁷³ http://fr.wikipedia.org/wiki/Collaborative_Design_Activity

⁷⁴ <http://sketchup.google.com/intl/fr/>

Chapitre 4

Extension d'Active3D à la modélisation Urbaine

Résumé

L'objectif de ce chapitre est d'apporter une proposition formalisée au problème de la modélisation sémantique, graphique et géographique d'informations du bâtiment et d'objets urbains dans un contexte de gestion technique de ces informations. Cette problématique vise à élargir la portée des outils et techniques de modélisation connus dans le monde du BIM et des SIG. L'idée est de pouvoir traiter de manière uniformisée les capacités de modélisation, géométriques et sémantiques, de l'univers du bâtiment et les notions typiquement issues des SIG (environnement urbain et terrains).

Par extension des travaux de Renaud Vanlande, l'approche de SIGA3D vise à construire, d'un côté, un graphe de scène contenant les informations relatives aux géométries des objets, avec notamment la possibilité d'associer plusieurs représentations géométriques à ces objets, puis d'un autre côté, un graphe sémantique à l'aide d'une extension du framework C-DMF. Ce graphe permet d'associer, dans une ontologie, plusieurs représentations sémantiques à un seul concept. En couplant ces deux approches, nous pouvons créer des niveaux de détail basés sur le contexte de visualisation des objets. L'objectif de ces niveaux de détail contextuels est d'optimiser le rendu géométrique des scènes en créant des niveaux de détail propres à chaque utilisation qui est faite des données. De sorte, la simplification des scènes est personnalisée et ne dépend plus des seuls critères géométriques, mais également de critères sémantiques.

Plan

1	Modélisation des données et Géométrie	112
2	Modélisation des données et multireprésentation sémantique.....	114
2.1	Pre-requis	115
2.2	Les classes et propriétés	116
2.2.1	Déclaration contextuelle de classes	116
2.2.2	Déclaration contextuelle de propriétés	117
2.2.3	Les prédicats.....	118
2.2.4	Les opérateurs.....	120
3	La couche de gestion du contexte.....	129
3.1.1	Le systemGraph.....	129
3.1.2	La validité du et dans le modèle.....	132
4	Conclusion	135

Pour répondre aux contraintes présentées dans le chapitre précédent, ce chapitre propose une solution formalisée pour la résolution du problème de la modélisation sémantique, graphique et géographique d'informations du bâtiment et d'objets urbains dans un contexte de gestion technique de ces informations. Cette problématique qui vise à élargir la portée des outils et techniques de modélisation connus dans le monde du BIM et des SIG est illustrée par la Figure 42. L'idée dans cette illustration est de pouvoir traiter de manière uniformisée les capacités de modélisation, géométriques et sémantiques, de l'univers du bâtiment et les notions typiquement issues des SIG (environnement urbain et terrains). Si chacune des disciplines maîtrise parfaitement son environnement (la modélisation 3D et sémantique de bâtiments pour le BIM, la gestion des coordonnées, des niveaux de détail et d'un volume important de données pour le SIG), la convergence vers un écosystème unique reste aujourd'hui un défi. Cela est dû en particulier aux problèmes d'hétérogénéité que l'on peut rencontrer à plusieurs niveaux (syntaxique, schématique et sémantique) comme expliqués dans l'état de l'art.

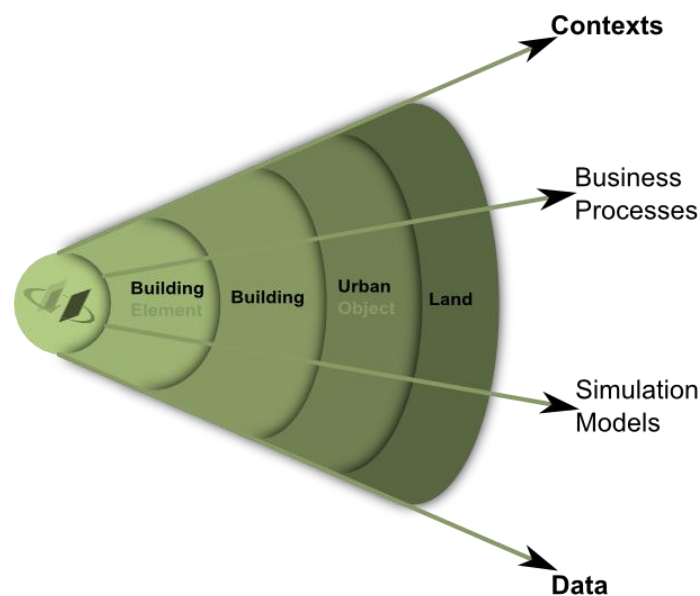


Figure 42. Convergence des méthodes BIM et SIG pour l'ouverture de la modélisation sémantique

Le couplage des trois domaines principaux traités au sein des BIM et des SIG (sémantique, 3D et géographique) nous permet d'introduire un nouveau concept d'optimisation de l'information que nous avons appelé niveaux de détail contextuels (C-LoD pour Contextual Level of Detail). Comme le montre l'état de l'art, dans chacun de ces domaines il existe traditionnellement des mécanismes spécifiques de traitement de l'information par niveau de détail : les LoD en 3D simplifient la géométrie des objets en fonction de leur distance à la caméra, la résolution dans les SIG consiste à définir des représentations correspondant à une appréciation de la densité spatiale des données, et la multireprésentation sémantique permet de structurer l'information selon des critères sémantiques (contexte d'utilisation ou contextes extérieurs comme le jour/nuit, saison...) et ainsi « filtrer » l'information utile.

L'idée de l'architecture SIGA3D est de combiner ces trois mécanismes de simplification des données afin de créer une représentation unifiée. Il existe déjà des travaux combinant différentes techniques citées ci-dessus pour optimiser l'approche d'un domaine en particulier :

- le couplage des niveaux de détail 3D avec ceux des SIG dans GML ou les SIG 3D en général (Köninger et Bartel, 1998),
- les ontologies spatiales ou géographiques (Spaccapietra et al., 2004) qui mixent la multireprésentation sémantique avec les concepts de niveaux détail géographique,

- la combinaison de la sémantique avec des représentations 3D comme dans le BIM (Active3D permet par exemple, en utilisant les IFC, de filtrer des éléments dans une scène 3D en fonction de leurs propriétés), la modélisation de bâtiment dans un sens plus général encore, comme dans les travaux de (Hu, 2008), voire même dans le sens le plus global de modélisation 3D (Spagnuolo et Falcidieno, 2009).

Ces différentes approches et leur positionnement par rapport aux domaines étudiés sont illustrés dans la Figure 43.

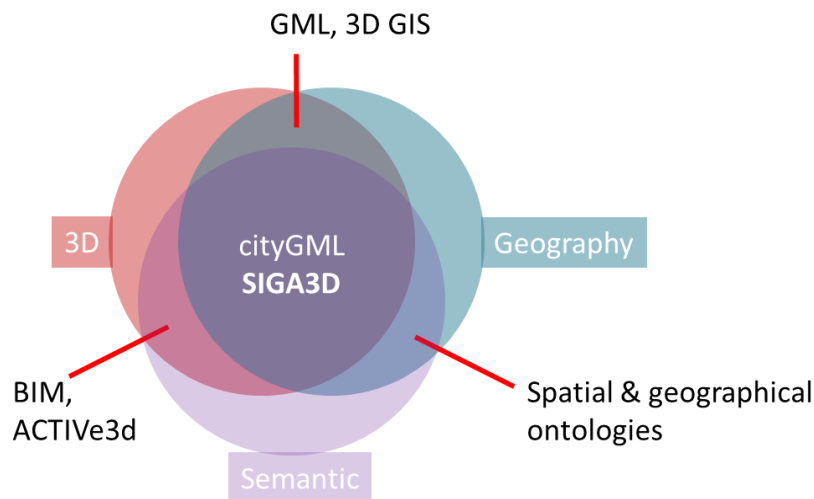


Figure 43. Positionnement des niveaux de détail contextuels de SIGA3D par rapport aux domaines de la 3D, de l'information géographique et de la gestion des connaissances

Pour concevoir une solution qui réponde à ces différents objectifs, nous avons étendu le BIM Active3D défini par Renaud Vanlande afin d'intégrer les contraintes des Systèmes d'Information Géographique et plus spécifiquement les notions de géolocalisation, de temporalité et de montée en charge. En effet, comme le nombre de données à représenter dans la scène 3D devient très volumineux (un environnement modélisé en 3D avec des objets urbains et des bâtiments IFC représentant plusieurs millions d'objets 3D), nous avons dû développer une nouvelle solution de gestion des niveaux de détail. Nous avons introduit dans notre approche une gestion de détail basé aussi sur la sémantique des objets et leur relation de pertinence avec le contexte de l'utilisateur.

L'objectif des niveaux de détail contextuels est donc d'optimiser le rendu géométrique des scènes en créant des niveaux de détail propres à chaque utilisation qui est faite des données. De sorte, la simplification des scènes est personnalisée et ne dépend plus des seuls critères géométriques, mais également de critères sémantiques. Pour ce faire, nous construisons dans un premier temps un graphe de scène contenant les informations relatives aux géométries des objets. Cette modélisation est particulière puisqu'elle donne la possibilité d'associer plusieurs représentations géométriques à ces objets. Si le mécanisme est similaire à ceux utilisés pour la gestion des LoD géométriques traditionnels, des adaptations sont néanmoins nécessaires pour pouvoir sélectionner les représentations en fonction de critères sémantiques. Dans un second temps, nous construisons un graphe sémantique à l'aide de l'architecture SIGA3D, ce qui nous permet d'associer dans une ontologie plusieurs représentations sémantiques à un seul concept. En couplant ces deux approches, nous pouvons créer nos niveaux de détail sémantiques.

Ce chapitre présente les outils que nous avons mis en place afin de construire les niveaux de détail contextuels. Cela commence par l'architecture supportant la définition de nos concepts et le moyen de leur définir plusieurs représentations. Nous parlons à ce niveau de contextes locaux (le contexte porte sur la définition du concept). Cette notion vient en complément des contextes de modèles et d'utilisation qui sont décrits dans la partie suivante (où le contexte porte sur un ensemble de contexte définissant un modèle, dans son ensemble). Enfin,

nous verrons plus en détail la construction du lien entre les représentations sémantiques et les éléments géométriques de notre graphe de scène.

La formalisation qui est faite ici est axée principalement sur la gestion du modèle de données et de la gestion du contexte que nous pouvons voir sur la Figure 42. Ces deux couches composent une architecture plus globale dont la définition complète est présentée dans le chapitre suivant. Cette architecture globale contient six couches réalisant le processus de traitement depuis l’acquisition des données issues de différentes sources, propres au BIM ou aux SIG, jusqu’à l’exploitation de celles-ci par l’utilisateur, sous forme de scènes 3D ou bien d’arbres alphanumériques (affichant l’information purement sémantique). Chacune de ces techniques de visualisation du patrimoine modélisé contient des informations sémantiques, géométriques et géographiques, et ce sous forme contextuelle (pouvant dépendre à la fois de l’environnement de l’utilisateur, mais aussi de l’information intrinsèque aux données).

L’architecture qui permet de modéliser et d’exploiter la connaissance des objets gérés est présentée dans la Figure 45. Celle-ci peut se décomposer en trois parties et est basée sur le modèle de définition de données BIM appelé C-DMF (Vanlande et al., 2008). Nous retrouvons dans l’architecture les briques CMF (Context Model Framework) et DMF qui composent l’architecture de base du BIM. Celles-ci correspondent respectivement d’une part à une architecture de graphes pour l’ontologie permettant la gestion du contexte et le versioning (CMF) ainsi que des opérateurs de graphes facilitant l’application des changements de conceptualisation, et d’autre part à une syntaxe unifiée basée sur les langages de représentation des connaissances OWL, RDFS, et de règles RuleML, SWRL, permettant l’implémentation de la brique C (DMF). Nous ajoutons à cette architecture la prise en compte d’opérateurs de gestion de l’espace et du temps, notion importante dans les SIG, ainsi que la définition des contextes locaux qui nous permettent d’effectuer une multireprésentation sémantique des données.

Les opérateurs de relations spatio-temporelles définis dans la couche des données peuvent être utilisés dans la couche de contexte pour lier certains modèles de données : on parle de synchronisation des modèles. Ainsi, il devient par exemple possible de définir l’inclusion spatiale au niveau des modèles de données entre les biefs d’un canal et ses écluses pour reprendre l’exemple cité dans l’introduction, ou l’inclusion temporelle entre deux événements (une terrasse est couverte *pendant* qu’il pleut). La Figure 44 illustre cet exemple de lien topologique entre les deux modèles de graphe avec une relation spatiale, TPP - *Tangential proper part of*, (présentée plus loin). Les relations de topologies sont définies au niveau du modèle de données et se propagent aux instances du modèle.

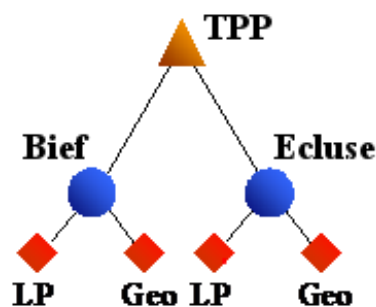


Figure 44. Exemple de relation spatiale (inclusion tangente) entre deux modèles de données

La prise en compte des notions d’espace et de temps évoqués ci-dessus est basée en partie sur les travaux de MADS (Spaccapietra et al., 2002) étudié dans l’état de l’art. Ces travaux proposent des solutions pour modéliser le temps, l’espace et permettre une multireprésentation des objets dans des structures de données adaptées, ces quatre dimensions (espace, temps, multireprésentation et structure) étant orthogonales, c’est à dire que

les choix fait dans l'une des branches n'influent en rien les autres branches. Nous ajoutons aussi la notion de concepts locaux en nous basant sur le mécanisme de multireprésentation.

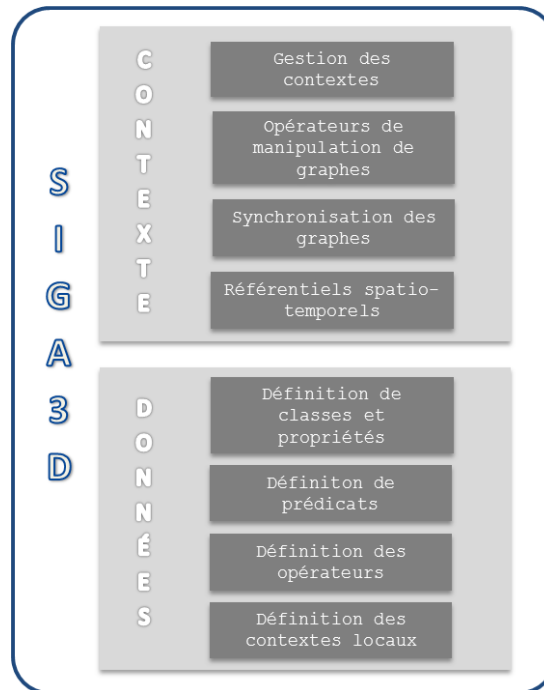


Figure 45. Détail des couches de données et de contexte de SIGA3D

1 Modélisation des données et Géométrie

Au sein de l'architecture SIGA3D, dans la couche « Données », la gestion des géométries constitue un bloc à part entière. En effet, le nom et caractéristique des choses et leur enveloppe sont deux types d'information fondamentalement différents. Il est plus facile de travailler et d'échanger de l'information sémantique que des géométries, mais afficher des listes et des tableaux n'est pas suffisant pour les intervenants qui ont besoin de situer les choses visuellement au sein d'une hiérarchie normalisée (MacPherson et Raymond, 2010).

Pour ces raisons, les données géométriques sont stockées séparément des autres données (sémantiques) afin d'alléger au maximum les bases de données et les requêtes nécessaires à l'affichage. Ainsi, les graphismes sont traités sous forme de graphe de scène, ce qui permet un chargement direct (sans transformations des données) au sein du moteur graphique de l'application. Le graphe sémantique, quant à lui, est issu de l'architecture ontologique décrite dans le paragraphe précédent.

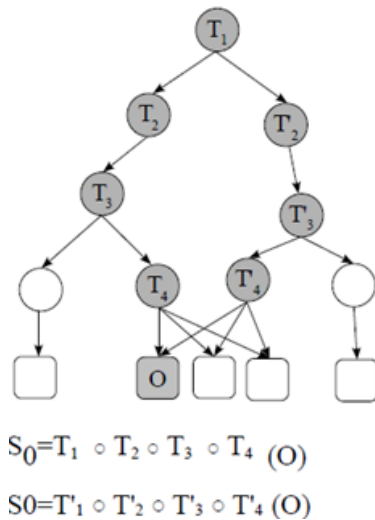


Figure 46. Exemple de graphe acyclique orienté

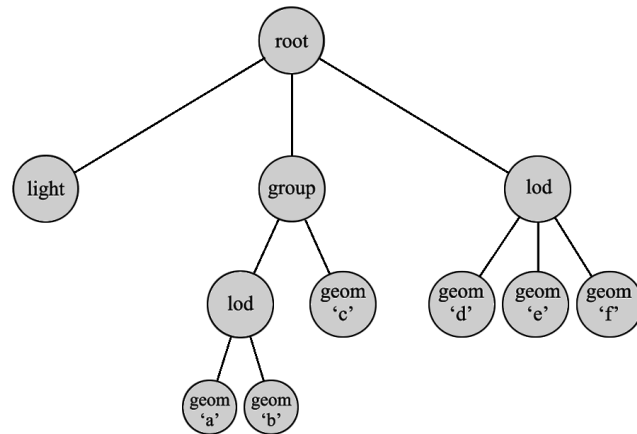


Figure 47. Exemple d'un graphe de scène avec des groupes de nœuds et niveaux de détail (source : (Germs et al., 1999))

La notion de graphe de scène est apparue dans les années 90, à l'époque où la programmation de scènes 3D demandait de solides notions en programmation avec des bibliothèques graphiques de bas niveau, afin de manipuler des entités de plus haut niveau (Huot, 2005). Le mécanisme des graphes de scènes est aujourd'hui largement utilisé dans les applications faisant de la modélisation 3D ou même 2D (jeux vidéo, visualisation scientifique, CAO...) du fait de sa compatibilité avec les principes de la programmation objet et une bonne structuration de la scène. On utilise naturellement le graphe de scène pour décrire des relations spatiales entre les objets. Le principe est d'utiliser un graphe acyclique orienté (un exemple est donné dans la Figure 46), afin d'organiser hiérarchiquement une scène 3D. Chaque nœud est un élément de la scène à représenter, celle-ci étant définie par une racine (un nœud du graphe) et un sous-graphe du graphe de scène. On obtient ainsi une simplification et une optimisation de l'ordre d'affichage pour les moteurs de rendu 3D, et les propriétés de chaque entité sont aisément transmissibles à leur descendance (position, orientation, contexte graphique...).

Les données géométriques, tout comme les autres objets nécessaires à la visualisation (telles que les sources de lumière) sont stockés dans un graphe de scène hiérarchique (Figure 47). Celui-ci est constitué de différents types de « nœuds », chacun ayant leurs propres effets et fonctionnalités. Ainsi, des données géométriques pourront être stockées dans un « nœud géométrique », et un « groupe de nœud » peut être utilisé pour regrouper plusieurs géométries (comme celles des enveloppes composées). En particulier, un « nœud niveau-de-détail » est utilisé pour stocker plusieurs représentations d'un même objet, allant des géométries les plus simples aux plus détaillées. Seulement l'une de ces représentations est visible à un point donné, en fonction du niveau de détail nécessaire à la visualisation. En outre, chaque nœud contient des données définies par l'utilisateur (comme des données attributaires thématiques).

SIGA3D reprend le concept du graphe de scène en poussant le mécanisme de multireprésentation. En effet, dans le graphe de scène de l'architecture SIGA3D, il est possible de stocker plusieurs représentations d'un même objet, sans la notion systématique de simplification des géométries que l'on trouve dans le sens traditionnel des LoD. Ainsi, il est possible d'avoir plusieurs représentations possédant un même niveau de complexité géométrique (nombres de triangles par exemple, ou représentation 2D et représentation 3D) mais dont les critères d'affichage sont différents. La difficulté dans ce cas est donc de choisir quelle représentation afficher pour un objet.

Pour cela, la structure de modélisation sémantique détaillée dans le prochain paragraphe permet, entre autres, de faire le lien entre les différentes représentations sémantiques d'un objet et les représentations géométriques du graphe de scène.

2 Modélisation des données et multireprésentation sémantique

La couche des données dans l'architecture SIGA3D est une extension de DMF qui constitue la première brique de l'architecture C-DMF (Figure 48). Ces deux briques couplées répondent aux besoins de description ontologique par la définition de règles avec un mécanisme de multireprésentation sémantique des données. La représentation multiple des données est autorisée grâce à la définition de contextes locaux sur certains opérateurs, couplée à la définition spécifique d'éléments représentant l'espace et le temps, moteurs principaux des besoins de représentation multiple.

La présentation des éléments est regroupée en quatre ensembles, la majorité des éléments pouvant être associés à un opérateur équivalent défini de manière locale par rapport à un contexte.

- Le *premier groupe* comprend les éléments de définitions de classes et de propriétés et leur correspondant associé au contexte local.
- Le *deuxième groupe* présente les éléments de définitions de variables, de prédicats unaires et binaires. Ces éléments sont utilisés pour la définition d'énoncés logiques à l'aide des opérateurs logiques présentés dans le *troisième groupe* (comme l'opérateur d'implication permettant d'établir des règles).
- Enfin, le *dernier groupe* correspond à la définition des contextes locaux.

Cette partie s'attache notamment à définir l'action des contextes locaux sur chacun des opérateurs concernés afin de comprendre leur utilité. Mais commençons par détailler le principe des contextes locaux.

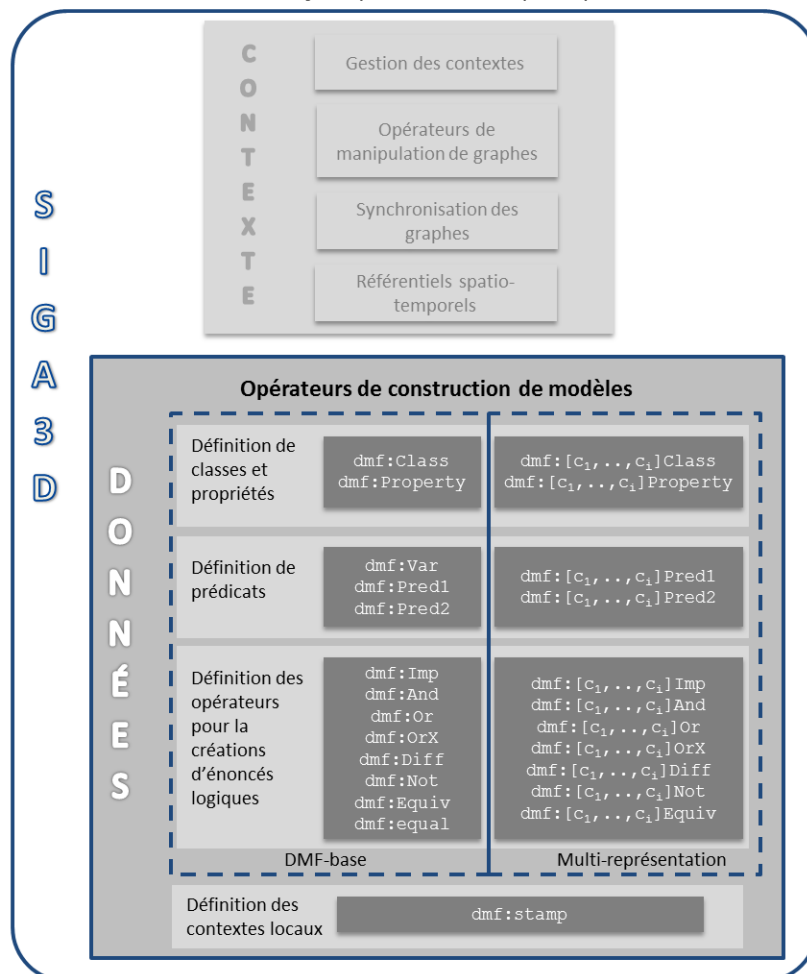


Figure 48. SIGA3D : détail de la couche de modélisation des données

2.1 Pre-requis

Les contextes locaux sur les opérateurs sont définis par la création de nouveaux opérateurs. On étend l’ensemble des opérateurs de base de C-DMF. Afin de bien différencier les opérateurs de base, ceux-ci sont référencés dans notre formalisation par l’appellation DMF-base. Dans l’approche DMF-base, il a été défini des espaces de noms par les mécanismes de RDF que nous allons réutiliser. Pour information, les espaces de nom utilisé dans DMF et repris dans cette section, sont rappelés dans le Script 12.

```
xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema#"
xmlns:dmf = "http://www.a3d.net/2007/dmf-syntax#"
```

Script 12. Définition des espaces de noms

Les espaces de noms permettent d’appeler une ressource sans utiliser l’adresse complète de la ressource. Par exemple, `rdf:type` correspond, avec la définition ci-dessus, à la ressource `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`.

La suite de ce chapitre présente l’ensemble des éléments de la couche de données. Pour chaque élément présenté, nous exposons sa définition dans la syntaxe RDF/XML avec ses propriétés, et nous présentons un exemple d’utilisation de l’élément dans le domaine du bâtiment. Une formalisation en logique de description est également donnée afin de comprendre l’effet des contextes locaux appliqués aux différents opérateurs. Cette description formelle en langage de description permet d’avoir un niveau d’abstraction suffisamment élevé pour être réutilisé par la suite dans un contexte différent de celui pour lequel l’architecture a été définie.

La consistance du modèle créé avec ce formalisme ne peut être vérifiée que par la validation de l’ensemble des modèles définis pour chaque contexte. En pratique, le contexte courant est défini à l’aide d’un et d’un seul contexte local. Par exemple, si l’on a trois contextes qui sont `dessinateur`, `gestionnaire` et `nuit`, l’utilisateur devra faire le choix du contexte courant qu’il voudra explorer dans l’ontologie. Il n’est pas possible de définir un contexte `gestionnaire+nuit` par exemple, puisque certains objets de l’ontologie peuvent posséder une représentation pour `gestionnaire`, et une représentation différente pour `nuit`. Dans ce cas, il faut définir le contexte local `gestionnaire_de_nuit` comme contexte à part entière pour lequel les concepts définis sont cohérents ; il faut précalculer la représentation, c’est-à-dire définir un contexte `gestionnaire_de_nuit` pour lequel les objets qui ont une représentation particulière sont décrits grâce aux opérateurs contextuels de la couche de données.

La liste des contextes est exclusive, c’est-à-dire qu’un ensemble de contextes ne définit pas la condition nécessaire pour définir l’opérateur, mais que l’opérateur est défini pour chacun des contextes présents dans la liste. De plus, comme en logique des prédicats, il n’est pas possible de définir plusieurs fois un même objet pour un même contexte (non-sens). Par contre il est possible de définir un objet à l’aide de DMF-base et de redéfinir ce même objet avec un opérateur contextuel. Dans ce cas le contexte est prioritaire sur la définition globale de l’opérateur dans DMF-base, c’est-à-dire que l’objet, dans ce contexte particulier, prend la valeur définie dans l’ensemble de la couche de données et qui n’est pas définie dans DMF-base.

Le contexte est utilisé comme étiquette dans les opérateurs que nous allons décrire ci-dessous. Cependant cette étiquette peut être définie à l’aide des opérateurs de DMF-base permettant ainsi de définir des étiquettes complexes. Par exemple, il est possible de définir une simple classe telle que `<dmf:Class rdf:ID='Contexte1' />` qui définit un contexte basé sur cette classe. Le but est double : cela permet de définir des contextes locaux ayant un sens (on peut par exemple expliciter la notion de `gestionnaire` à l’aide de `dmf`), et de référencer ces contextes sémantiques dans le but de leur garder ce même sens dans

différentes situations. Une des applications intéressantes de ce mécanisme est la possibilité par exemple de définir des horaires (pour déterminer l'heure d'ouverture d'un bâtiment au public).

Dans la description de l'architecture SIGA3D ci-dessous, chaque opérateur est défini en logique de description. Ainsi nous ne contraignons pas la proposition à un langage particulier. En effet, les concepts et les relations d'une ontologie peuvent être spécifiés formellement à l'aide d'un langage de Logique de Description (LD) (Baader et al., 2003), sur lequel opère un système de raisonnement. Les LD constituent une famille de formalismes de représentation des connaissances basés sur une logique du premier ordre et sur la notion de classes (concepts). Les LD fournissent plusieurs constructeurs permettant de combiner des concepts et rôles (atomiques ou composés) pour définir des concepts complexes. Dans leur grande majorité, les LD sont décidables et fournissent des services d'inférence. Une base de connaissances spécifiée en LD comprend un ensemble d'axiomes terminologiques (TBox), et un ensemble d'axiomes assertionnels (ABox), satisfaits par une interprétation \mathcal{I} .

Dans la formalisation ci-dessous, pour formaliser la multireprésentation sémantique, nous avons décidé d'utiliser une extension du langage de description logique *ALCN* avec des mécanismes tirés de la multireprésentation dans (Benslimane et al., 2003). Pour cela, nous utilisons des estampilles pour associer un contexte local à une entité simple du langage, puis nous utilisons de nouveaux opérateurs pour les concepts complexes (Tableau 4 ci-dessous). Les opérateurs que nous utilisons dans l'architecture SIGA3D peuvent être trouvés dans la logique de description appelée *SHOIN* et ses extensions.

Considérons s_1, \dots, s_m un ensemble d'estampilles, R un rôle et C et D des concepts complexes contextuels pouvant être créés selon la syntaxe suivante :

- $C, D \rightarrow \exists_{s_1, \dots, s_m} R.C$ (contextual existential quantification)
- | $\forall_{s_1, \dots, s_m} R.C$ (contextual value restriction)
- | $(\leq_{s_1, \dots, s_m} nR)$ (contextual maximum cardinality of a role)
- | $(\geq_{s_1, \dots, s_m} nR)$ (contextual minimum cardinality of a role)
- | $C \sqcap_{s_1, \dots, s_m} D$ (contextual conjunction)
- | $C \sqcup_{s_1, \dots, s_m} D$ (contextual disjunction)
- | $\neg_{s_1, \dots, s_m} C$ (contextual negation).

Tableau 4. Liste des opérateurs contextuels de logique de description.

2.2 Les classes et propriétés

Cette section présente le premier groupe d'opérateurs, les deux opérateurs permettant de créer des classes et des propriétés. Ces classes et ces propriétés pourront être utilisées dans des énoncés logiques. Dans ce cas ils sont des arguments des prédicats unaires et binaires.

2.2.1 Déclaration contextuelle de classes

L'élément `dmf: [c1, . . . , cn]Class` définit une classe. Le contexte s'applique alors à la définition de la classe. La définition de l'opérateur `dmf: [c1, . . . , cn]Class` est la suivante (Script 13):

```
<dmf: [c1, . . . , cn]Class rdf:ID='Class' />
```

Script 13. Définition de `dmf:Class` contextuel

Il est possible de définir, via la propriété `dmf:localContext`, un sous-ensemble des contextes locaux de la classe pour lesquels l'instance est valide (**Script 14**).

```
<dmf:Property rdf:ID="localContext"/>
```

Script 14. Définition de dmf:localContext

Dans le Script 15 nous définissons une classe `Batiment` et une classe `Etage` valable dans les contextes locaux c_1 à c_n pour la classe `Batiment` et c_i à c_n pour la classe `Etage`. `'batiment_1'` et `'etage_1'` sont des instances de ces classes, avec une particularité pour `etage_1` qui n’est définie que dans le contexte c_i .

```
<dmf:[c1, ..., cn]Class rdf:ID='Batiment' />
<dmf:[ci, ..., cn]Class rdf:ID='Etage' />
<Batiment rdf:ID='batiment_1' />
<Etage rdf:ID='etage_1'>
  <dmf:localContext rdf:dataType='&xsd:string'>
    ci
  </dmf:localContext>
</Etage>
```

Script 15. Exemple de description contextuelle d’une vue d’un Bâtiment

On peut traduire tout cela en langage de description comme le montre le Script 16.

```
TBox :
Batiment  $\sqsubseteq$  Tc
Etage  $\sqsubseteq$  Tc
ABox :
c1, cn : Batiment(batiment_1)
ci : Etage(etage_1)
```

Script 16. Exemple de description contextuelle d’une vue d’un Bâtiment en logique de description

2.2.2 Déclaration contextuelle de propriétés

L’élément `dmf:[c1, ..., cn]Property` définit une propriété. Le contexte de cet élément permet de définir la validité de la relation entre deux ressources ou une ressource avec une valeur, indépendamment du contexte défini pour ces ressources. Nous définissons `dmf:[c1, ..., cn]Property` comme étant de type `rdf:Property` dans le contexte donné (Script 17):

```
<dmf:[c1, ..., cn]Property rdf:ID='Property' />
```

Script 17. Définition de dmf:Property

La propriété `dmf:localContext` peut s’appliquer à l’instanciation d’une propriété. L’exemple présenté dans le Script 18 définit une propriété `contenir`. Cette propriété est utilisée pour relier `batiment_1` avec `etage_1` signifiant que le bâtiment 1 contient l’étage 1 pour le contexte c_i . Pour les autres contextes cette relation n’est pas définie.

```
<dmf:[ci]Property rdf:ID='contenir' />
<Batiment rdf:ID='batiment_1' />
<contenir rdf:resource='#etage_1' />
</Batiment>
```

```
TBox :
contenir(x, y)  $\sqsubseteq$  TR
```

```
BatimentAvecEtage ≡ Batiment  $\prod_{c_1, \dots, c_m} \forall_{c_1, \dots, c_m} \text{contenir} . \exists_{c_1, \dots, c_m} \text{Piece}$ 
ABox :
 $c_1, \dots, c_m$  :  $\text{contenir}(\text{Batiment}, \text{Piece})$ 
```

Script 18. Exemple de définition d'une propriété contextuelle "contenir" et son équivalent en logique de description

2.2.3 Les prédicats

Dans la section précédente nous avons présenté les éléments de base de la contextualisation de DMF, nous permettant ainsi de créer des classes et des propriétés dépendant d'un contexte défini de manière local. Dans cette section, nous introduisons le deuxième groupe d'éléments : les prédicats. Les prédicats sont les composants de base des énoncés logiques. En effet, ils sont utilisés pour décrire des énoncés logiques en les combinant avec des opérateurs que nous présenterons dans la section suivante.

Déclarations de prédicats contextuels

Les éléments $\text{dmf}:[c_1, \dots, c_n]\text{Pred1}$ et $\text{dmf}:[c_1, \dots, c_n]\text{Pred2}$ permettent de définir des prédicats contextuels, respectivement unaire et binaire. Leur définition est présentée dans le Script 19. Les prédicats contextuels sont utilisés pour définir des relations contextuelles (prédicats binaires) entre des concepts atomiques (prédicats unaires) pouvant dépendre d'un contexte local. Les variables utilisées dans la définition des prédicats sont celles de DMF-base. En effet, les variables n'ayant de sens qu'au travers de leurs occurrences déterminées par les prédicats qui les utilisent, définir un contexte sur celles-ci n'a pas de sens puisqu'il est implicitement dépendant du contexte local des prédicats.

```
<Class rdf:ID="Pred1"/>
<Class rdf:ID="Pred2"/>
```

Script 19. Définition de prédicats dmf

Les propriétés de ces éléments sont les mêmes que l'on retrouve pour les opérateurs de DMF-base et sont présentées dans le Script 20.

```
<dmf:Property rdf:ID="arg1"/>
<dmf:Property rdf:ID="arg2"/>
<dmf:Property rdf:ID="classPred"/>
<dmf:Property rdf:ID="propertyPred"/>
<dmf:Property rdf:ID="minCard"/>
<dmf:Property rdf:ID="maxCard"/>
```

Script 20. Définition des propriétés des prédicats

Un prédicat unaire possède les propriétés dmf:arg1 (une variable) et dmf:classPred (qui lie la variable à une classe de type `classPred`). Un prédicat binaire possède les deux propriétés dmf:arg1 et dmf:arg2 (deux variables) ainsi que la propriété dmf:propertyPred (qui lie la variable à une propriété de type `propertyPred`). Il est également possible d'y associer les deux propriétés dmf:minCard , dmf:maxCard pour définir la cardinalité minimale et maximale du nombre de valeurs possibles pour une propriété. La multireprésentation sémantique associée à ces prédicats permet de moduler ces propriétés en fonction du contexte local courant. Par exemple si l'on définit deux classes `Etage_avec_representation` et `Etage_sans_representation` valides dans deux contextes locaux différents `gestionnaire` et `utilisateur`, ainsi qu'un prédicat unaire qui rassemble les étages d'un bâtiment, on peut définir le prédicat contextuel `Etages` comme suit (Script 21):

```

<dmf:[gestionnaire,utilisateur]Class      rdf:ID='Etage' />
<dmf:Var rdf:ID='_x' />
<dmf:[gestionnaire]Pred1      rdf:ID='Etages'>
  <dmf:classPred rdf:resource='#Etage_avec_représentation' />
  <dmf:arg1      rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[utilisateur]Pred1 rdf:ID='Etages'>
  <dmf:classPred rdf:resource='#Etage_sans_représentation' />
  <dmf:arg1      rdf:resource='#_x' />
</dmf:Pred1>

```

Script 21. Exemple de description d'un étage avec des prédicats contextuels

Dans le contexte *gestionnaire*, le prédicat unaire *Etage(?x)* désigne l'ensemble des ressources de type *Etage_avec_représentation*. Dans le contexte *utilisateur*, le même prédicat unaire *Etage(?x)* désigne l'ensemble des ressources de type *Etage_sans_représentation*. L'équivalent en logique de description des prédicats unaires contextuels est décrit dans le cas général par un exemple dans le Script 22.

```

TBox :
Etages  $\sqsubseteq T_C$ 
x  $\sqsubseteq T_x$ 
Pred1  $\models_{c_1, \dots, c_m} x : Etages$ 

```

Script 22. Correspondance des prédicats unaires contextuels avec la logique de description

Dans le Script 22, la notation $\text{Pred1} \models_{c_1, \dots, c_m} x : \text{Etages}$ signifie que *Pred1* est une conséquence logique de la formule $x : \text{Etages}$ dans les contextes locaux listés, c'est-à-dire que toute interprétation terminologique qui satisfait *Pred1* satisfait également $x : \text{Etages}$ dans ces contextes particuliers. Le Script 23 présente un exemple d'utilisation des prédicats pour la définition de contraintes de cardinalités en fonction d'un contexte local. On modélise les expressions $[1, 3]\text{contenir}(?x, ?y)$ et $[1, n]\text{contenir}(?x, ?y)$ qui désignent l'ensemble des triplets dont le sujet *x* possède entre 1 et *n* ou entre 1 et 3 propriétés *contenir*, en fonction du contexte d'utilisation (*avant_1990* et *après_1990*).

```

<dmf:Property rdf:ID='contenir' />
<dmf:Var rdf:ID='_x' />
<dmf:Var rdf:ID='_y' />
<dmf:[avant_1990]Pred2 rdf:ID='pred2_0'>
  <dmf:propertyPred      rdf:resource='#contenir' />
  <dmf:arg1              rdf:resource='#_x' />
  <dmf:arg2              rdf:resource='#_y' />
  <dmf:minCard rdf:dataType='&xsd:string'>1</dmf:minCard>
  <dmf:maxCard rdf:dataType='&xsd:string'>3</dmf:maxCard>
</dmf:Pred2>
<dmf:[après_1990]Pred2 rdf:ID='pred2_0'>
  <dmf:propertyPred      rdf:resource='#contenir' />
  <dmf:arg1              rdf:resource='#_x' />
  <dmf:arg2              rdf:resource='#_y' />
  <dmf:minCard rdf:dataType='&xsd:string'>1</dmf:minCard>
  <dmf:maxCard rdf:dataType='&xsd:string'>n</dmf:maxCard>
</dmf:Pred2>

```

Script 23. Exemple de définition de la multi représentation des cardinalités avec les prédicats en fonction d'un contexte local

La traduction des prédicats binaires en logique de description est montrée dans le Script 24, avec et sans restriction de cardinalité.

```
Sans restriction de cardinalité
TBox :
contains  $\sqsubseteq T_R$ 
 $x, y \sqsubseteq T_x$ 
Predb_1  $\models$  contenir ( $x, y$ )

Avec restriction de cardinalité
TBox :
contenir  $\sqsubseteq T_R$ 
 $x, y \sqsubseteq T_x$ 
Predb_1  $\models \forall s_1, \dots, s_m x. \text{contenir}(x, y) \sqcap_{s_1, \dots, s_m} \geq_{s_1, \dots, s_m} 1 \text{ contenir}(x, y) \sqcap_{s_1, \dots, s_m} \leq_{s_1, \dots, s_m} n \text{ contenir}(x, y)$ 
```

Script 24. Équivalence des prédicats binaires contextuels en logique de description, avec et sans restriction de cardinalité

2.2.4 Les opérateurs

Dans la section précédente, nous avons vu l'utilité et la manière de déclarer des prédicats contextuels. C'est l'un des mécanismes centraux de la contextualisation locale des données. Nous allons maintenant voir comment nous pouvons utiliser les opérateurs contextuels, qui, combinés aux prédicats (de DMF-base ou associés aux contextes locaux), nous permettent d'établir des énoncés logiques.

2.2.4.1 Opérateur contextuel d'implication

Cet élément désigne l'opérateur d'implication associé à un ou plusieurs contextes locaux d'utilisation. C'est l'élément principal pour l'établissement d'énoncés logiques. Un opérateur d'implication est composé d'une condition et d'une conséquence. La définition de l'opérateur d'implication est donnée dans le Script 25.

```
<Class rdf:ID="Imp"/>
```

Script 25. Définition de dmf:Imp contextuel

Le Script 26 décrit les deux propriétés de l'opérateur `dmf:Imp`, `dmf:argCond` (condition) et `dmf:argCons` (conséquence) qui peuvent être utilisées dans l'opérateur contextuel `dmf:[c1, ..., cn] Imp`. Ce nouvel opérateur permet, en fonction des contextes locaux existants, de définir des implications logiques spécifiques à chacun des contextes. Par exemple, en fonction de l'heure de la journée (contexte `jour` et `nuit`), et de la profession d'une personne (`employé` et `veilleur`), on peut définir le fait que tout individu de type `Bureau` est un individu de type `Zone` ou bien de type `Zone_A_Risque` (`Bureau(?x) → Zone(?x)` ou `Bureau(?x) → Zone_A_Risque(?x)`). Cette expression se traduit à l'aide des opérateurs contextuels par le bloc XML suivant :

```
<dmf:Class rdf:ID='Zone' />
<dmf:[veilleur]Class rdf:ID='Zone_A_Risque' />
<dmf:Class rdf:ID='Bureau' />
<dmf:Var rdf:ID='_x' />
<dmf:Pred1 rdf:ID='pred1_0'>
  <dmf:classPred rdf:resource='#Zone' />
```

```

    <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[veilleur]Pred1 rdf:ID='pred1_0v'>
  <dmf:classPred rdf:resource='#Zone_A_Risque' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:Pred1 rdf:ID='pred1_1'>
  <dmf:classPred rdf:resource='#Bureau' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:Imp rdf:ID='imp_0'>
  <dmf:argCond rdf:resource='#pred1_1' />
  <dmf:argCons rdf:resource='#pred1_0' />
</dmf:Imp>
<dmf:[veilleur,nuit]Imp rdf:ID='imp_0'>
  <dmf:argCond rdf:resource='#pred1_1' />
  <dmf:argCons rdf:resource='#pred1_0v' />
</dmf:Imp>

```

Script 26. Exemple de l’utilisation de $\text{dmf} : [C_1, \dots, C_n] \text{Imp}$

Toutes les interprétations possibles de l’opérateur d’implication peuvent être contextualisées. Le Script 27 montre la définition et l’impact du contexte sur celles-ci en langage de description.

Subsomption de propriétés $p1(?x, ?y) \rightarrow p2(?x, ?y)$

TBox :

S_1, \dots, S_m : contenirMobilier \sqsubseteq contenir

Propriétés de Domain et Range telle que $p(?x, ?y) \rightarrow A(?x)$ et $p(?x, ?y) \rightarrow B(?y)$

TBox :

BatContenantEtage \sqsubseteq Bat $\prod_{s_1, \dots, s_m} \forall_{s_1, \dots, s_m} \text{contenirEtage} . \exists_{s_1, \dots, s_m} \text{Etage}$

Transitivité $p(?x, ?y) \sqcap p(?y, ?z) \rightarrow p(?x, ?z)$

TBox :

EtageAvEtageInferieur \sqsubseteq Etage $\prod_{s_1, \dots, s_m} \forall_{s_1, \dots, s_m} \text{avoirEtageInferieur+} . \exists_{s_1, \dots, s_m} \text{Etage}$

Symétrie : $p(?x, ?y) \rightarrow p(?y, ?x)$

TBox :

etreAuMemeEtage $\sqsubseteq T_R$

$x, y \sqsubseteq T_x$

$\text{Pred1} \models_{s_1, \dots, s_m} \text{etreAuMemeEtage} (x, y)$

$\text{Pred1}^- \models_{s_1, \dots, s_m} \text{etreAuMemeEtage} (y, x)$

$\text{Pred1}^- \sqsubseteq_{s_1, \dots, s_m} \text{Pred1}$

Propriété fonctionnelle : $p(?x, ?y) \wedge p(?x, ?z) \rightarrow ?y=?z$

TBox :

avoirEtageSuivant $\sqsubseteq T_R$

avoirEtageSuivant $\sqsubseteq T_R$

$x, y, z \sqsubseteq T_X$

Pred1 \models avoirEtageSuivant(x, y)

Pred2 \models avoirEtageSuivant(x, z)

$y=z \sqsubseteq_{s1, \dots, sm} \text{Pred1} \sqcap_{s1, \dots, sm} \text{Pred2}$

Propriété inverse : $p1(?x, ?y) \rightarrow p2(?y, ?x)$

TBox :

avoirEtageSuivant $\sqsubseteq T_R$

avoirEtagePrecedent $\sqsubseteq T_R$

$x, y \sqsubseteq T_X$

Pred1 \models avoirEtageSuivant(x, y)

Pred2 \models avoirEtagePrecedent(y, x)

Pred1 $\sqsubseteq_{s1, \dots, sm} \text{Pred2}$

Subsorption de classes $A(?x) \rightarrow B(?x)$

TBox :

Zone $\sqsubseteq T_C$

Bureau $\sqsubseteq T_C$

$x \sqsubseteq T_X$

Pred1 $\models x : \text{Zone}$

Pred2 $\models x : \text{Bureau}$

Pred2 $\sqsubseteq_{s1, \dots, sm} \text{Pred1}$

Propriété fonctionnelle inverse : $p(?x, ?y) \wedge p(?z, ?y) \rightarrow ?x=?z$

TBox :

avoirEtageSuivant $\sqsubseteq T_R$

avoirEtagePrecedent $\sqsubseteq T_R$

$x, y, z \sqsubseteq T_X$

Pred1 \models avoirEtageSuivant(x, y)

Pred2 \models avoirEtagePrecedent(z, y)

$x=z \sqsubseteq_{s1, \dots, sm} \text{Pred1} \sqcap_{s1, \dots, sm} \text{Pred2}$

Restriction de type quantification universelle (AllValuesFrom) : $A(?x) \wedge p(?y, ?z) \rightarrow B(?y)$

TBox :

Maison $\sqsubseteq T_C$

avoirEtageSuivant $\sqsubseteq T_R$

EtageMaison $\sqsubseteq T_C$

```

x, y, z ⊆ Tx
Pred1 ⊢ Maison(x)
Pred2 ⊢ avoirEtageSuivant(y, z)
Pred3 ⊢ MaisonAvecEtages(y)
Pred3 ⊆s1, ..., sm Pred1 ∏s1, ..., sm ∑s1, ..., sm Pred2.Pred1

```

Restriction de cardinalité : $A(?x) \rightarrow [i, j]p(?x, ?y)$

```

TBox :
contenir ⊆ TR
Batiment ⊆ TC
x, y ⊆ Tx
Pred1 ⊢ Batiment(x)
Pred2 ⊢ ∑s1, ..., sm x.contenir(x, y) ∏s1, ..., sm ≥s1, ..., sm 1 contenir(x, y) ∏s1, ..., sm ≤s1, ..., sm n contenir(x, y)
Pred2 ⊆ Pred1

```

Contrainte de valeur : $A(?x) \rightarrow p(?x, valeur)$

```

TBox :
avoirCouleur ⊆ TR
BatimentNoir ⊆ TC
Couleur ⊆ TC
x, y ⊆ Tx
Couleur(noir)
Pred1 ⊢ BatimentNoir(x)
Pred2 ⊢s1, ..., sm avoirCouleur(x, 'noir')
Pred2 ⊆s1, ..., sm Pred1

```

Script 27. Définition du contexte local appliqué à l’opérateur $dmf : [c_1, \dots, c_n] Imp$ pour chaque cas d’utilisation possible.

Dans le Script 27, la propriété `avoirEtageInferieur+` désigne une propriété transitive en langage de description (partie transitivité). La notation `Pred1-` est juste une convention de nommage pour définir la symétrie d’une propriété (dans la partie symétrie du script).

2.2.4.2 Opérateur d’intersection

L’élément $dmf : [c_1, \dots, c_n] And$ désigne l’opérateur d’intersection contextuel. Il réalise l’intersection entre plusieurs ensembles dont le contexte peut être différent. Le contexte appliqué à cet opérateur signifie à la fois l’existence de l’opérateur pour la liste de contextes donnés et la sélection des représentations propres à ce contexte pour les prédicats qui définissent cet opérateur. L’ensemble d’individus résultat d’une intersection contextuelle est l’ensemble des ressources RDF qui répondent vrai à tous les arguments de l’intersection dans chacun des contextes donnés. Cet élément possède une propriété, $dmf:args$, dont la valeur de type `rdf:List` permet de définir la liste des prédicats dont on veut effectuer l’intersection.

Le Script 28 présente un exemple de syntaxe et d’utilisation de l’opérateur d’intersection contextuel. Il sélectionne les ressources qui sont à la fois un `lycée` et un `collège` dans un contexte d’une gestion des biens par une `Région`, en opposition au contexte de l’État.

```

<dmf:[Région,Etat]Class rdf:ID='Lycee' />
<dmf:[Région,Etat]Class rdf:ID='College' />
<dmf:Var rdf:ID='_x' />
<dmf:[Région,Etat]Pred1 rdf:ID='pred1_2'>
  <dmf:classPred rdf:resource='#Lycee' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région]Pred1 rdf:ID='pred1_3'>
  <dmf:classPred rdf:resource='#College' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région,Etat]And rdf:ID='And_0'>
  <dmf:args rdf:parseType='Collection'>
    <dmf:Pred rdf:about='pred1_2' />
    <dmf:Pred rdf:about='pred1_3' />
  </dmf:args>
</dmf:And>

```

Script 28. Exemple d'utilisation contextuelle de l'opérateur `dmf:And`

Dans cet exemple, nous avons trois cas d'interprétations possibles pour la définition de l'opérateur `And_0`. Si l'on est dans une situation générale, c'est-à-dire sans contexte courant particulier défini, cet opérateur n'existe pas. Si l'on est dans un contexte de gestion des biens par une région, alors l'opérateur définit une intersection entre les collèges et les lycées. Dans le cas où l'on est dans un contexte de gestion par l'état, l'opérateur ne définit que l'ensemble des lycées puisqu'il n'y a pas de représentation connue pour les collèges dans ce contexte. Le Script 29 définit cet opérateur en logique de description.

```

TBox:
Collège ⊆ Tc
Lycée ⊆ Tc
x ⊆ Tx
Predu_1 ⊨ x:Collège
Predu_2 ⊨ x:Lycée
Predu_1 ⊓s1, ..., sm Predu_2

```

Script 29. L'opérateur contextuel `dmf[c1, .., cn]And`

2.2.4.3 Opérateur d'union

L'opérateur d'union contextuel réalise l'union entre plusieurs ensembles en sélectionnant pour chacun d'eux la représentation pertinente pour la liste des contextes donnés. L'ensemble des individus résultat d'une union contextuelle est l'ensemble des ressources RDF qui répondent vrai à au moins un argument de l'union dans n'importe lequel des contextes donnés. Comme l'opérateur d'intersection, l'opérateur d'union possède la propriété `dmf:args`. Le Script 30 montre un exemple d'utilisation de cet opérateur contextuel. On peut ainsi définir un ensemble de zones d'un bâtiment à explorer. Le résultat de cette union est l'ensemble des ressources contenant des bureaux, des couloirs et des sanitaires dans le cas d'une région, ou l'ensemble des ressources contenant des bureaux et des sanitaires pour l'état.

```

<dmf:Class rdf:ID='Bureau' />
<dmf:Class rdf:ID='Couloir' />
<dmf:Class rdf:ID='Sanitaire' />
<dmf:Var rdf:ID='_x' />

```

```

<dmf:[Région,Etat]Pred1 rdf:ID='pred1_1'>
  <dmf:classPred rdf:resource='#Bureau' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région,Etat]Pred1 rdf:ID='pred1_2'>
  <dmf:classPred rdf:resource='#Couloir' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région,Etat]Pred1 rdf:ID='pred1_3'>
  <dmf:classPred rdf:resource='#Sanitaire' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région]Or rdf:ID='Or_0'>
  <dmf:args rdf:parseType='Collection'>
    <dmf:Pred rdf:about='pred_1' />
    <dmf:Pred rdf:about='pred_2' />
  </dmf:args>
</dmf:And>
<dmf:[Etat]Or rdf:ID='Or_0'>
  <dmf:args rdf:parseType='Collection'>
    <dmf:Pred rdf:about='pred_3' />
  </dmf:args>
</dmf:And>

```

Script 30. Exemple d’utilisation de $\text{dmf} : [c_1, \dots, c_n] \text{Or}$

La formalisation de cet exemple en logique de description est donnée dans le Script 31.

```

TBox :
Bureau  $\sqsubseteq T_c$ 
Couloir  $\sqsubseteq T_c$ 
Sanitaire  $\sqsubseteq T_c$ 
x  $\sqsubseteq T_x$ 

Predu_1  $\models x$ : Bureau
Predu_2  $\models x$ : Couloir
Predu_3  $\models x$ : Sanitaire
Predu_1  $\sqcup_{s1, \dots, sm}$  Predu_2  $\sqcup_{s1, \dots, sm}$  Predu_3  $\models x$ .Bureau  $\sqcup_{s1, \dots, sm}$  x.Couloir
 $\sqcup_{s1, \dots, sm}$  x.Sanitaire

```

Script 31. L’opérateur contextuel $\text{dmf} : [c_1, \dots, c_n] \text{Or}$

2.2.4.4 Opérateur de disjonction exclusive

L’opérateur de disjonction exclusive contextuel garantit qu’un individu membre d’une classe ne peut pas être simultanément l’instance d’une autre classe au sein d’un même contexte local. Il possède comme les deux opérateurs précédents la propriété $\text{dmf} : \text{args}$. Le Script 32 définit un ensemble ou un élément est de type Porte ou Fenêtre, mais pas les deux, dans le contexte Région et dans le contexte État. Cependant, il est possible d’avoir des éléments qui soient Porte dans le contexte Région et Fenêtre dans le contexte État ou vice versa.

```

<dmf:Class rdf:ID='Porte' />
<dmf:Class rdf:ID='Fenetre' />
<dmf:Var rdf:ID='_x' />
<dmf:Pred1 rdf:ID='pred1_6'>
  <dmf:classPred rdf:resource='#Porte' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:Pred1 rdf:ID='pred1_7'>
  <dmf:classPred rdf:resource='#Fenetre' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>
<dmf:[Région, État]OrX rdf:ID='OrX_0'>
  <dmf:args rdf:parsetype='Collection'>
    <dmf:Pred rdf:about='pred_6' />
    <dmf:Pred rdf:about='pred_7' />
  </dmf:args>
</dmf:And>

```

Script 32. Exemple d'utilisation de dmf:OrX

Le Script 33 est la traduction du Script 32 en logique de description.

```

TBox:
Door ⊆ TC
Window ⊆ TC
x ⊆ TX
Predu_1 ⊨ x:Door
Predu_2 ⊨ x:Window
Predu_1 ⊆ ¬s1,...,sm Predu_2 ⊨s1,...,sm x.Door ⊆ ¬s1,...,sm x.Window

```

Script 33. L'opérateur contextuel dmf:[s1,...,sm]OrX

2.2.4.5 Opérateur de négation

L'opérateur de négation contextuel possède la propriété `dmf:arg`. On peut ainsi désigner, comme dans l'exemple ci-dessous, un ensemble d'individus n'appartenant pas à un ensemble donné. Les scripts suivants (Script 34 et Script 35) nous montrent l'utilisation de l'opérateur `dmf[c1, cn]Not` où l'on reprend l'exemple utilisé dans la description de l'opérateur de prédicat unaire, le premier prédicat regroupant les classes de type étage avec représentation, le second, définit avec l'opérateur de négation contextuel, regroupant les autres classes. Comme pour les autres opérateurs, celui-ci ne peut être utilisé que sur des classes définies dans le même contexte local.

```

<dmf:Pred1 rdf:ID='Etages'>
  <dmf:classPred rdf:resource='#Etage_avec_representation' />
  <dmf:arg1 rdf:resource='#_x' />
</dmf:Pred1>

<dmf:[dessinateur]Not rdf:ID='not_0'>
  <dmf:arg rdf:resource='#Etages' />
</dmf:Pred1>

```

Script 34. Exemple d'utilisation de dmf:Not

```
TBox :
Office  $\sqsubseteq$  Tc
x  $\sqsubseteq$  Tx
Predu_1  $\models$  x:Etages_avec_representation
 $\neg_{s_1, \dots, s_m}$  Predu_2  $\models_{s_1, \dots, s_m} \neg_{s_1, \dots, s_m} x$  :Etages_avec_representation
```

Script 35. L’opérateur contextuel dmf:[s₁,...,s_m]Not

2.2.4.6 Opérateur de différence

Cet opérateur est l’opérateur contextuel de différence. Il définit une différence entre deux ensembles d’individus au sein d’un même contexte local. Il va essentiellement permettre de définir des groupes d’individus dépendant d’un contexte d’utilisation à partir de groupe d’individus défini de manière globale.

Il possède deux propriétés dmf:arg1 et dmf:arg2 dont les valeurs sont des prédicats.

Le Script 36 désigne, à partir du prédicat pred1_0 défini de façon globale, toutes les ouvertures qui ne sont pas des portes: Ouverture(?x) - Porte(?x) dans un contexte dessinateur. Le Script 37 désigne le même ensemble en logique de description.

```
<dmf:Pred1 rdf:ID = 'pred1_0'>
  <dmf:classPred rdf:resource = '#Ouverture' />
  <dmf:arg1 rdf:resource = '#_x' />
</dmf:Pred1>
<dmf:Pred1 rdf:ID = 'pred1_1'>
  <dmf:classPred rdf:resource = '#Porte' />
  <dmf:arg1 rdf:resource = '#_x' />
</dmf:Pred1>
<dmf:[dessinateur]Diff rdf:ID = 'diff_0'>
  <dmf:arg1 rdf:resource = '#pred1_0' />
  <dmf:arg2 rdf:resource = '#pred1_1' />
</dmf:Diff>
```

Script 36. Exemple d’utilisation de dmf:Diff

```
dmf:[s1, ..., sm]Diff défini la différence contextuelle :
TBox :
Door  $\sqsubseteq$  Tc
Window  $\sqsubseteq$  Tc
x  $\sqsubseteq$  Tx
Predu_1  $\models$  x : Ouverture
Predu_2  $\models$  x : Porte
Predu_1- $\neg_{s_1, \dots, s_m}$  Predu_2  $\models_{s_1, \dots, s_m} \forall_{s_1, \dots, s_m} x \sqsubseteq \neg_{s_1, \dots, s_m} (Ouverture \sqcap_{s_1, \dots, s_m} Porte)$ 
```

Script 37. L’opérateur contextuel dmf:[s₁,...,s_m]Diff

2.2.4.7 Opérateur de spatialité

L’opérateur dmf:spatialEntity vient ajouter une nouvelle dimension à l’architecture SIGA3D en permettant de lier des géométries à des concepts de l’ontologie décrite à l’aide des précédents opérateurs. Sa déclinaison contextuelle, dmf:[s₁, ..., s_m]spatialEntity, permet notamment de lier des représentations

d'un objet dans le graphe de scène SIGA3D à des concepts dans un contexte local particulier, réalisant ainsi la définition des niveaux de détail contextuels.

`dmf:[s1, ..., sm]spatialEntity` permet d'associer à n'importe quel objet, définit à l'aide des opérateurs décrits ci-dessus, une représentation. Il existe déjà un lien entre le graphe de scène et le graphe de données sémantique de SIGA3D par l'intermédiaire d'identifiants uniques. Ce mécanisme est décrit plus en détail dans la prochaine section concernant la définition du contexte, au niveau du méta-contexte. C'est un enrichissement de ce qui existe aujourd'hui avec les IFC et DWG dans l'application puisqu'avant il n'y avait pas de graphe de scène, on allait chercher les géométries uniquement dans le fichier d'origine après son chargement en mémoire. Cet opérateur permet à la fois de définir de nouvelles représentations pour un objet et de faire l'association d'une ressource graphique existante avec un nouveau concept ou concept contextualisé. Le lien avec l'objet est effectué à l'aide de ces identifiants globaux que l'on récupère ou calcule, puis chaque représentation géométrique de l'objet lié est alors identifiée afin de pouvoir être sélectionnée. Le stockage et la sélection de ces objets sont similaires à la gestion traditionnelle des niveaux de détail dans les graphes de scène, mais les conditions d'affichage d'une géométrie plutôt qu'une autre est effectuée selon les contextes locaux définis sémantiquement.

L'exemple formel de l'élément `dmf:spatialEntity` est simple puisqu'il ne fait que référencer des géométries. Cependant, celles-ci peuvent provenir du graphe de scène, comme expliqué précédemment, ou bien d'une autre source de données.

```
TBox :
URI ⊆ Tc
ABox :
s1, ..., si : URI (http://a3d.net/Representation.gml)
sj, ..., sm : URI (http://a3d.net/Representation2.mesh)
```

Script 38. L'opérateur contextuel `dmf:[s1,...,sm]spatialEntity`

Le Script 38 illustre l'utilisation de l'élément `dmf :spatialEntity` de deux manières : en liant un fichier ou l'on pourra trouver la géométrie à utiliser pour les contextes 1 à i, et un maillage dans le graphe de scène associé (l'association entre le graphe de scène à utiliser et le modèle de données est faite dans la partie contextualisation). La plupart du temps, c'est un lien vers une entité d'un fichier IFC ou GML/CityGML qui est utilisé.

On peut utiliser cet opérateur pour définir ou associer une représentation normalisée à chaque version d'un concept décrit, et ce dans sa globalité ou bien pour chaque contexte local. On peut également associer (mapper) des représentations issues de sources différentes à un même objet en définissant les conditions d'accès à une représentation en particulier (c'est-à-dire dans quel contexte local une représentation que l'on souhaiterait associer au concept est valide).

Les géométries fournies à cet endroit doivent être rattachées à un système de coordonnées terrestre. Les données pouvant provenir de sources diverses, il n'y a pas d'obligations quant à un système géodésique particulier. L'harmonisation des systèmes de référence s'effectue au niveau de la couche du contexte. C'est ainsi que lors de la phase de visualisation, et à l'aide des éléments fournis sur ces systèmes, une transformation de coordonnées est effectuée. Ces transformations pouvant induire plus ou moins de distorsions dans les représentations concernées, il est néanmoins préférable, en usage, d'uniformiser les systèmes de coordonnées. Le mécanisme de synchronisation spatiale décrit dans la couche de gestion du contexte des modèles est utilisé en partie pour pallier au problème de distorsion en décrivant des propriétés entre objets qui ne dépendent pas du système de coordonnées utilisé (principe de la topologie).

2.2.4.8 La description du temps

La description du temps est un cas particulier puisqu’on ne l’utilise, au niveau du modèle de données, que pour décrire les contextes locaux. Il n’y a donc pas d’opérateur de contextualisation du temps au niveau de ce modèle de données. De plus, n’importe quel système de temps peut être utilisé à ce niveau (à condition de le connaître), un système de référence temporel étant défini dans la couche CMF de l’architecture SIGA3D. La transformation du temps d’un système à un autre est alors effectuée lors de la visualisation des données, tout comme pour la définition et l’exploitation des données spatiales. La définition du temps est faite dans une forme standard, inspirée de OWL-Time (Hobbs et Pan, 2006), et utilise les propriétés et champs suivants : *unitType*, année, mois, semaine, jour, heure, minute, and seconde. La propriété obligatoire *unitType* précise le type d’unité temporel utilisé pour la description des éléments temporels (*TemporalEntity*) et son domaine de définition est l’unité temporelle (*TemporalUnit*) définie comme suit : $TemporalUnit \equiv \{unitSecond, unitMinute, unitHour, unitDay, unitWeek, unitMonth, unitYear\}$.

D’un côté on possède une connaissance sur la modélisation sémantique d’un ensemble de données du bâtiment, permettant d’avoir une multireprésentation des données au niveau du modèle au travers de la définition de vues personnalisées, et de profil utilisateur. De l’autre côté le monde du SIG maîtrise parfaitement la multireprésentation géométrique des données. On peut, grâce à la première couche de notre architecture décrite ci-dessus, retrouver les différentes possibilités offertes par les différents domaines, et aller plus loin en les combinant.

3 La couche de gestion du contexte

Après avoir présenté les différents composants de la couche de définition des données et le mécanisme de création des contextes locaux, nous allons présenter la couche de gestion des modèles de données et des contextes. La Figure 49 propose un aperçu et un positionnement de cette brique. La partie CS-DMF (Contextualisation and Synchronisation of Data Model Framework) est celle que nous allons décrire.

Cette couche est composée de quatre blocs principaux. Le premier bloc décrit le contexte pour chaque graphique de DMF. Le second bloc définit un ensemble d’opérateurs de manipulation de graphe pour faciliter l’écriture et limiter la redondance des données dans la gestion du contexte. Le troisième bloc définit un graphe spécial contenant des informations qui décrivent plus précisément l’aspect géographique du modèle de données en définissant des relations spatiales et temporelles entre les données décrites. Enfin, le dernier bloc indique où sont gérés les référentiels liés au traitement de l’espace et du temps dans notre architecture.

3.1.1 Le systemGraph

La gestion du contexte dans cette architecture se fait à l’aide d’un graphe spécial appelé *SystemGraph*. Un *SystemGraph* est un graphe ou un ensemble de graphes défini en utilisant les opérateurs de manipulation de graphes (couche centrale de la Figure 49). L’utilisation de ces opérateurs simplifie l’évolution des connaissances du modèle. Ainsi, plutôt que de stocker, pour chaque modification du modèle, une nouvelle version de l’information, la couche de gestion du contexte stocke les modifications sous forme d’opérations sur les graphes. Le *SystemGraph* peut être décrit comme dans le Script 39 ci-dessous :

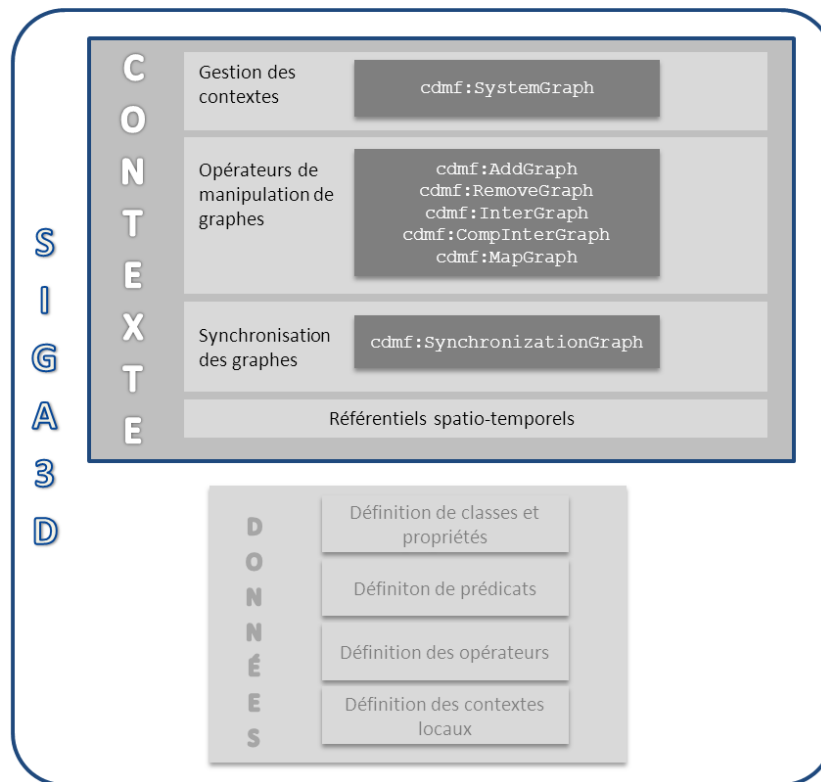


Figure 49. SIGA3D : détail de la couche de modélisation du contexte

TBox:

```

SystemGraph, SystemModelGraph, Subject, Label, Action, Graph, AddGraph,
RemoveGraph, MapGraph, InterGraph, SynchronizationGraph  $\sqsubseteq T_c$ 
model, of, name, action, graph, add, remove, synchronizationGraph,
reference_frame, timezone, crs  $\sqsubseteq T_r$ 
 $x, y \sqsubseteq T_x$ 
model(x, y)  $\models$  SystemGraph(x)  $\sqcap$  =1SystemModelGraph(y)
of(x, y)  $\models$  SystemGraph(x)  $\sqcap$  Subject(y)
name(x, y)  $\models$  Subject(x)  $\sqcap$  Label(y)
action(x, y)  $\models$  SystemGraph(x)  $\sqcap$   $\leq 2$  Action(x)
add(x, y)  $\models$  Action(x)  $\sqcap$  AddGraph(y)
remove(x, y)  $\models$  Action(x)  $\sqcap$  Graph(x)  $\sqcap$  RemoveGraph(y)
map(x, y)  $\models$  Action(x)  $\sqcap$  MapGraph(y)
inter(x, y)  $\models$  Action(x)  $\sqcap$  InterGraph(y)
graph(x, y)  $\models$  SystemGraph(x)  $\sqcap$   $\leq 1$ Graph(y)
synchronizationGraph(x, y)  $\models$  SystemGraph(x)  $\sqcap$  SynchronisationGraph(y)
reference_frame  $\models$  SystemGraph(x)  $\sqcap$   $\leq 2$  Subject(y)
timezone(x, y)  $\models$  Subject(x)  $\sqcap$  Label(y)
crs(x, y)  $\models$  Subject(x)  $\sqcap$  Label(y)

```

Script 39. Définition du SystemGraph de la couche de gestion des contextes de la brique CS-DMF

Nous pouvons voir dans cette définition quelques éléments intéressants :

- *cdmf:graph* lie les graphes et les données. Celles-ci sont décrites à l'aide de la brique données de notre architecture. Le graphe lié à cet opérateur peut être une combinaison de plusieurs graphes de données, définit à l'aide des opérateurs *AddGraph* (union de graphes), *RemoveGraph* (Différence de

graphes), *InterGraph* (Intersection de graphes), *CompInterGraph* (Intersection composée de graphes) and *MapGraph* (Correspondance de graphes)

- *cdmf:of* représente le contexte à proprement parler. Cette propriété définit une liste de ressources représentant le contexte d’accès aux données.
- *cdmf:model* définit pour un *SystemGraph* le modèle de données utilisé. On y retrouve les types d’éléments qui apparaîtront dans le graphe.
- *cdmf:action* définit les droits d’accès (lecture, écriture, suppression) pour l’utilisateur. Si aucune action n’est définie, seule la visualisation est possible.
- *cdmf:synchronizationGraph* définit une liste de graphe dépendant d’un modèle spécial qui permet de décrire toute sorte de relations spatiales et temporelles entre les modèles.
- *cdmf:reference_frame* définit un fuseau horaire et un système de coordonnées de référence utilisés pour la description du modèle de données associé au *SystemGraph*. Ces valeurs sont valides pour toutes les données associées au graphe, ce qui nécessite un mécanisme de transformation des coordonnées lorsque la source de données est décrite dans un système de référence différent de celui du modèle.

Cette liste de propriété du *SystemGraph* introduit le graphe spécial *SynchronizationGraph*. Le modèle duquel dépend ce graphe contient les relations spatio-temporelles. Les relations spatiales sont définies d’après le formalisme RCC (Region Connexion Calculus) qui permet la représentation et le raisonnement spatial (Hogenboom et al., 2010). Les relations supportées sont les suivantes : *Connected* (C), *Part of* (P), *Overlaps* (O), *Externally connected to* (EC), *Disconnected from* (DC), *Discrete from* (DR), *Proper part* (PP), *Tangential proper part of* (TPP), *Non-tangential proper part of* (NTPP), *Equal to* (EQ), *Partially overlaps* (PO), *Overlaps not equal* (ONE), *Spatially related* (SR). Les relations temporelles sont basées sur les relations d’intervalles définis dans OWL-Time. Les éléments du *SynchronizationGraph* ne sont pas décrits directement dans le modèle, mais par référence aux modèles des graphes associés. En l’occurrence, ceux-ci sont liés par *AddGraph* ce qui nous permet de décrire tout type de relations spatio-temporelles entre les modèles et les données de ces modèles. Le Script 40 illustre le fonctionnement à l’aide du graphe *SyncModelGraph* :

```

ABox of SystemGraph Sg1:
  SyncModelGraph (Sg1)
    model (Sg1, Sdmf)
    of (Sg1, su1)
      Subject (su1)
        name (su1, 'Synchronization Model')
        graph (Sg1, </graph/ModelSync#>)

TBox of </graph/ModelSync#>:
  spatialRelation, temporalRelation  $\sqsubseteq T_c$ 
  spatiallySynchronizedWith, temporallySynchronizedWith  $\sqsubseteq T_r$ 
  x, y  $\sqsubseteq T_x$ 
  Predu  $\models (\forall \text{spatialRelation.synchronizedWith}(x, y) \sqcap \geq 1$ 
  spatiallySynchronizedWith (x, y))  $\sqcup (\forall$ 
  temporalRelation.synchronizedWith(x, y)  $\sqcap \geq 1$  temporallySynchronizedWith
  (x, y))

ABox of the graph </graph/ModelSync#> :
  spatialRelation (SR)
  spatialRelation (C)
  spatialRelation (DC)

```

```

spatialRelation (P)
spatialRelation (O)
spatialRelation (DR)
spatialRelation (EC)
spatialRelation (EQ)
spatialRelation (ONE)
spatialRelation (PP)
spatialRelation (PO)
spatialRelation (TPP)
spatialRelation (NTTP)

temporalRelation (intEquals)
temporalRelation (intBefore)
temporalRelation (intMeets)
temporalRelation (intOverlaps)
temporalRelation (intStarts)
temporalRelation (intDuring)
temporalRelation (intFinishes)
temporalRelation (intAfter)
temporalRelation (intMetBy)
temporalRelation (intOverlappedBy)
temporalRelation (intStartedBy)
temporalRelation (intContains)
temporalRelation (intFinishedBy)

```

Script 40. Définition du graphe spécial `SyncModelGraph` utilisé dans CS-DMF

La définition du graphe `SyncModelGraph` est basée sur celle du `SystemGraph` puisque ce modèle est un `SystemGraph` particulier. Le modèle associé à celui-ci est appelé `Sdmf`. Il fait référence à l'espace système `SpaceSystem` de DMF à partir duquel tous les nouveaux modèles sont créés. Le graphe référencé, appelé `ModelSync`, contient les relations spatiales et temporelles. Il permet de lier les modèles de données de DMF avec une ou plusieurs relations. Le modèle vérifie la cohérence des relations en permettant une seule relation de chaque type entre deux graphes DMF. Par ailleurs, le modèle ne peut pas être modifié si aucune action n'est définie pour ce graphe. Le `SynchronizationGraph` peut être créé à partir de ce modèle. De plus, comme ce graphe manipule des données spatiales et temporelles, il est nécessaire de gérer les systèmes de référence: les caractéristiques spatiales ont besoin d'un CRS et les données temporelles nécessitent la connaissance du fuseau horaire utilisé. Dans notre architecture, chaque source de données peut décrire les propriétés des objets dans n'importe quel système. Lors de l'exploitation des données, une transformation des coordonnées sera effectuée si nécessaire. Cette information est donnée dans le `SystemGraph` associé au graphe des données du DMF comme nous l'avons vu plus tôt.

Le graphe de synchronisation que nous avons introduit ici induit un nouveau comportement dans l'exploitation de notre ontologie par l'intermédiaire de la validité des objets, au niveau des concepts définis dans les graphes, ou au niveau des modèles de données. Nous allons étudier ces problèmes dans le paragraphe suivant.

3.1.2 La validité du et dans le modèle

Dans les paragraphes précédents, il est souvent fait allusion à la validité de concepts (ou la validité d'une représentation d'un concept) et du modèle de données. Ce paragraphe va expliquer ce que signifie la validité, comment elle est définie et exploitée. L'expression « modèle de données » définit ici un graphe possédant un contexte (décrit à l'aide du `SystemGraph`). Par exemple, la Figure 50 nous montre de manière graphique la définition d'un modèle de données appelé « `City district 1` ».

La validité intervient à deux niveaux : la validité de la représentation d'un concept, induite par le contexte d'utilisation du modèle de données contenant le concept (on parle de contexte courant), et la validité du modèle de données, guidée par les liens de synchronisation existant avec d'autres modèles.

Pour le premier cas, il s'agit tout simplement de déterminer, pour un concept donné qui possède plusieurs représentations, quelle représentation est valide et dans quel modèle de données. Pour cela, la seule référence du contexte local utilisé pour définir la représentation concernée dans le contexte du modèle de données (`cdmf:of`) permet de sélectionner cette représentation pour ce modèle de données. Cela explique les limitations imposées lors de la définition des contextes locaux. Dans notre exemple, on voit qu'il existe un contexte local « *Version 2* » pour lequel une propriété « *contain* » est définie. Comme le contexte du modèle de données que nous étudions possède la référence à ce contexte local, cette propriété existe. Si la version du graphe étudié avait été la version 1, cette propriété n'aurait pas existé.

Intéressons-nous maintenant à la validité du modèle de données à proprement parler. Cette notion de validité à ce niveau est surtout intéressante en phase d'exploitation du modèle de données. La phase d'exploitation signifie une extraction de ce modèle de données pour l'afficher, soit de manière textuelle (arborescence alphanumérique par exemple, comme c'est le cas dans Active3D pour exploiter la sémantique), ou sous forme de scènes 3D lorsque de telles représentations existent. La validité du modèle peut alors prendre deux formes : soit on va filtrer l'affichage pour n'afficher que les modèles valides, soit l'on va ajouter à l'affichage les conditions de validité des objets et modèles affichés (dans la plateforme Active3D, cela peut être fait sous la forme de texte 3D par exemple, textes qui sont visibles en permanence indépendamment de l'angle et de la position de la caméra). Par exemple, si on s'intéresse à un modèle de données représentant une maison, puis à un autre modèle de données représentant un garage décrit sous forme d'annexe à la maison, on peut définir deux relations de synchronisation entre ces deux modèles : l'annexe n'existe que si le modèle de la maison existe, et l'annexe est tangente à la maison. Ainsi, si l'on affiche une scène 3D antérieure à l'existence de la maison, on sait que le modèle de l'annexe ne peut pas être affiché, car non valide, même si aucune date de validité de ce modèle de données n'a été définie au niveau du contexte de ce modèle. De plus, on sait que si la maison est affichée, et que pour une raison quelconque elle est repositionnée (par un changement de Système de référence géodésique par exemple), le modèle de l'annexe devra être spatialement repositionné pour garder la relation topologique de tangence décrite au niveau du modèle. Les relations spatiales sont vérifiées lorsqu'un modèle lié à un autre, ou l'une de ces caractéristiques est modifiée. Les relations temporelles permettent également de définir des états en fonction d'instantanés ou d'intervalles de temps définis plus tôt, par exemple l'ouverture d'un bâtiment ou d'un parc ouvert au public. Cela permet, dans le domaine archéologique par exemple, de superposer des éléments à des périodes différentes. On renforce ainsi par ce biais les notions et la visualisation de cycle de vie et de cohérence par rapport à l'espace au temps.

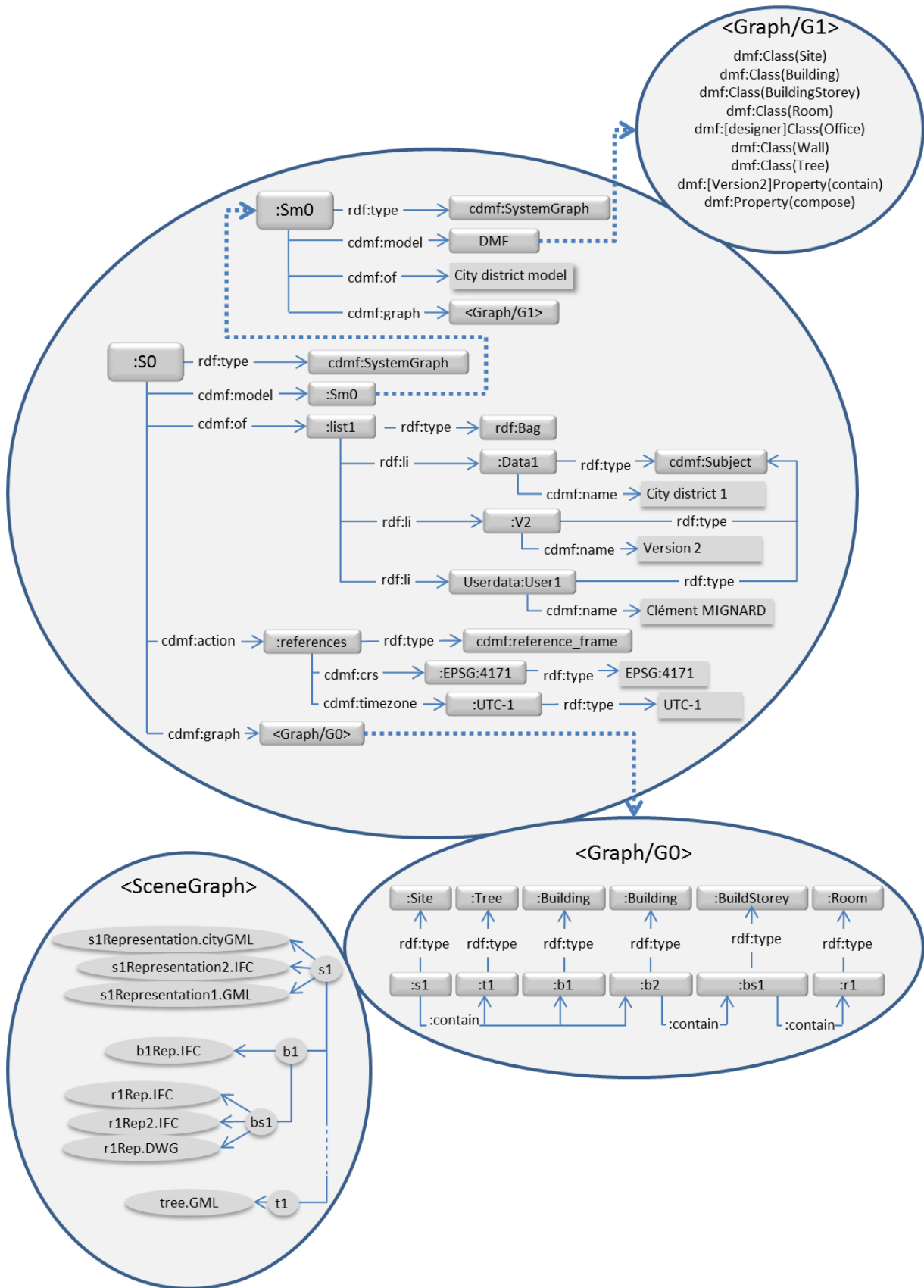


Figure 50. Exemple de définition d'un modèle de données (city district 1)

4 Conclusion

Nous avons étudié dans ce chapitre l'architecture SIGA3D. Celle-ci est composée dans sa globalité de six parties. Nous nous sommes attachés à formaliser de manière détaillée les couches de définition des données avec la mise en place du mécanisme de multireprésentation sémantique et la couche de contextualisation des données avec la description du mécanisme de synchronisation spatio-temporel des modèles de données. La combinaison de ces deux couches nous permet notamment, en phase d'exploitation, de combiner la connaissance sémantique et la connaissance géographique que nous avons d'un modèle de données pour créer des niveaux de détails géométrico-sémantique que nous avons appelés C-LoD (Contextual Levels of Details).

La couche des données que nous avons décrite est composée de trois blocs : les opérateurs de base de C-DMF (1^{er} bloc) qui est l'architecture de modélisation des BIM sur laquelle nous nous sommes basés, la définition de contextes locaux (2^e bloc) qui permettent, à l'aide d'opérateurs locaux (3^e bloc) de définir des concepts, simples ou complexes (basés sur des prédicats), possédant plusieurs représentations.

La couche des contextes contient, elle, quatre blocs. On y retrouve la définition et gestion du contexte d'un modèle de données, la gestion du graphe de données à l'aide d'opérateurs de graphes permettant de faciliter l'écriture, la maintenance et le suivi du cycle de vie des modèles de données, la possibilité de définir des liens entre les modèles (liens spatio-temporels), et la gestion des référentiels d'espace et de temps nécessaire à la normalisation des traitements liés à ces types de données.

Chapitre 5

Architecture SIGA3D

Résumé

Si le projet SIGA3D est avant tout un projet de recherche à forte dimension industrielle, la société Active3D a souhaité porter ce projet à l'échelle européenne. En partenariat avec la société allemande Lat-Lon, elle a donc obtenu un financement dans le cadre d'un projet Euréka. L'objectif de ce projet européen était d'obtenir, après 3 années, un ensemble de fonctionnalités permettant de mettre en place une extension d'Active3D à la gestion technique de patrimoine Urbain. Pour atteindre cet objectif, une architecture de processus plus complète a été développée autour du noyau d'architecture sémantique réalisé dans le cadre de ce travail de thèse. Cette architecture de processus est articulée en 6 Niveaux. Ces processus permettent la gestion des sources de données hétérogènes, la gestion des imports et exports des données, la gestion des connexions entre la base de données, les éléments web et le moteur 3D et enfin la gestion de l'affichage dans une interface Web. Les deux niveaux restants correspondant à la couche de modélisation sémantique des données de SIGA3D et à la couche de modélisation des contextes qui font l'objet de ce mémoire.

Plan

1	Niveau 1 – Sources de données	139
2	Niveau 2 – Import / Export SIGA3D	140
3	Niveau 3 – Data Model Framework de SIGA3D	142
4	La gestion spatiale des objets urbains.....	143
5	Niveau 4 – La couche de gestion du contexte	146
5.1	Le contexte	146
5.2	La gestion des référentiels et la synchronisation spatio-temporelle	148
6	Niveau 5 – La couche de connexion	149
7	Conclusion	149

Depuis 2008, dans le cadre d'un projet européen, nous développons une plateforme internet collaborative dédiée à la gestion d'éléments urbains (UFM pour Urban Facility Management). Cette approche est basée sur la définition d'un modèle d'information urbain (UIM) combinant des technologies du BIM (Building Information Model) et des SIG (modélisation sémantique, géométrique et géographique). Le but est d'accéder à la connaissance globale d'un environnement urbain complet, incluant des ensembles de bâtiments et les objets urbains qui composent cet environnement. L'architecture complète de la plateforme d'Urban Facility Management SIGA3D est représentée dans la Figure 51. Elle est divisée en six couches, depuis la récupération des données jusqu'à leur exploitation. Le but est de représenter un environnement composé d'objets urbains, incluant les bâtiments, réseaux, éléments urbains de proximité (éclairage public, espaces collectifs, mobilier urbain...), et plus généralement, tous les objets géographiques.

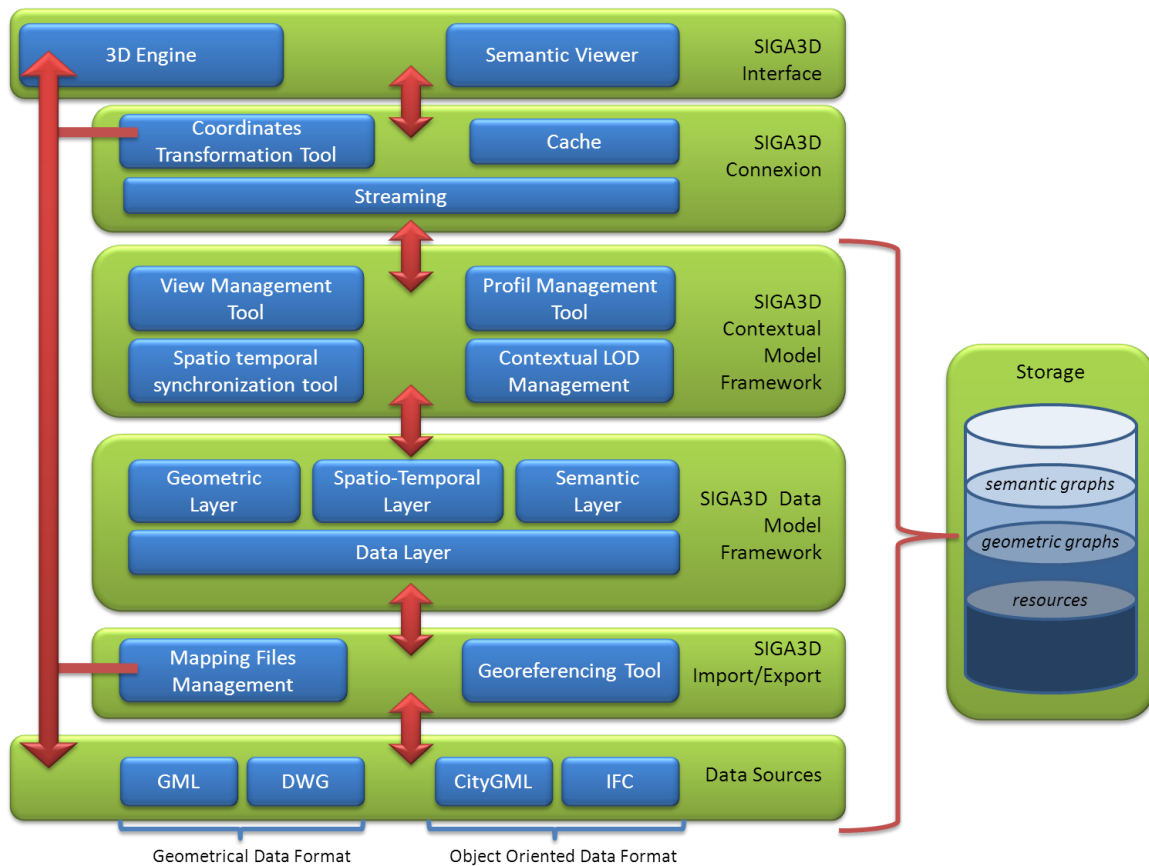


Figure 51. Architecture de SIGA3D

Dans ce chapitre, nous présentons succinctement ces 6 couches. L'objectif est de montrer qu'au-delà de la définition d'une architecture sémantique pour la résolution des contraintes de modélisation d'un modèle d'information urbain, mon travail de thèse a consisté à développer un système complet, depuis la lecture jusqu'à la visualisation des données. Cela demande notamment d'adapter la recherche fondamentale et théorique à une architecture d'entreprise existante.

1 Niveau 1 – Sources de données

Ce niveau correspond à l'acquisition des données depuis plusieurs sources souvent hétérogènes schématiquement, syntaxiquement et sémantiquement pouvant être traitées par SIGA3D. L'énumération des sources n'est ici pas exhaustive, car l'architecture a été conçue pour être évolutive et autoriser l'ajout de traitements de nouveaux formats de fichier ou web services. Actuellement, Active3D supporte les fichiers IFC

et CityGML (partiellement) dans la catégorie des formats de fichiers orientés objet, et les fichiers DWG, DXF et DGN pour la partie non-objet. Des études ont également été faites pour supporter les web services de l'OGC tel que WFS (Web Feature Service) et WMS (Web Map service) avec l'aide de notre partenaire européen, lat/lon. Leur solution, deegree, est un projet de l'OSGeo. Ensemble, nous travaillons à la gestion des objets géographiques dans Active3D. La coopération est basée essentiellement sur la problématique du géo-référencement, l'optimisation des larges scènes en 3D et la modélisation du terrain.

À ce niveau de l'architecture, les données peuvent être soit directement stockées en tant que ressources pour être intégrées par référencement dans les niveaux supérieurs, soit utilisées directement par le viewer 3D, avec éventuellement quelques traitements spécifiques des couches supérieures (gestion des coordonnées, détection des objets...), soit importées et transformées par la couche d'import pour ensuite être intégrée au niveau factuel de l'ontologie définie au sein de l'architecture SIGA3D.

2 Niveau 2 – Import / Export SIGA3D

Cette partie est l'implémentation technique du processus d'import. Le but est ici d'extraire tous les objets décrits dans les sources de données, qu'ils soient décrits explicitement (comme dans les IFC) ou non (comme dans les DWG par exemple), et d'établir des relations entre ceux-ci. On cherche à récupérer ou créer un identifiant pour chaque objet identifié de manière à traiter et importer l'information. Cette couche d'import est fortement liée aux niveaux supérieurs de l'architecture SIGA3D de modélisation et stockage de l'information. En effet, les types d'objets et relations dont nous avons besoin pour la reconnaissance des objets sont décrits dans ces niveaux. Pour connaître l'information que nous devons importer, et faire la correspondance avec les objets types définis, nous utilisons des fichiers de mappage. Ces fichiers sont également utilisés pour résoudre l'hétérogénéité syntaxique que nous pouvons rencontrer avec les nombreuses sources de données utilisables. Un exemple de fichier de mappage IFC est donné dans le Script 41. Nous y décrivons les correspondances entre les objets de type `BuildingStorey` dans Active3D et les objets `IfcBuildingStorey` des fichiers IFC. Le principe est de décrire dans ce fichier de mappage comment les objets d'un format particulier peuvent être intégrés et utilisés dans l'architecture Active3D. En l'occurrence, nous avons construit notre ontologie avec, entre autres, le concept d'étage (`building storey`), des relations entre ces étages (`hasName`), et le concept de nom d'étage. Il nous suffit alors d'instancier ces deux concepts avec les objets que nous trouvons dans le fichier IFC. Chaque nœud du graphe de l'ontologie est identifié. En faisant de même pour tous les concepts qui nous intéressent, nous pouvons importer automatiquement le niveau factuel de notre ontologie depuis des fichiers IFC.

```
<Objects>
  <Object id="869" name="BuildingStorey">
    <IfcBuildingStorey>
      <StoreyName id="2821" catId="610">
        .[Label]
      </StoreyName>
    </IfcBuildingStorey>
  </Object>
  <Object id="870" name="Rooms">
    <IfcSpace>
      <Function id="2826" catId="611">.[Name]</Fonction>
      <Affectation id="2872" catId="611">
        .[Label]
      </Affectation>
      <Area id="2830" catId="611">
        \RelProperties.conteneur-IfcPropertySet.1-
        (IfcPropertySingleValue{Name=ComputedArea}.[Value]
```

```

</Surface>
</IfcSpace>
</Object>
</Objects>

```

Script 41. Exemple de fichier de mappage IFC

Si les formats normalisés comme IFC ou CityGML contiennent des mécanismes d'identifiant uniques pour chaque entité, des formats comme DWG ne le font pas et il est alors nécessaire de créer les conditions permettant d'identifier l'information. Pour ces formats de fichier non-objet (ou la part de sémantique est minime), la méthodologie de mapping est donc quelque peu différente. Une première étape est nécessaire pour déterminer les entités sémantiques intéressantes depuis l'information géométrique. La méthodologie pour associer un nom et un identifiant à une entité (les éléments requis pour créer une instance d'objet dans SIGA3D) doit être détaillée dans le fichier de mappage. Ainsi, pour les géométries DWG par exemple, nous pouvons utiliser la décomposition des plans en calques pour identifier les types d'objets que l'on souhaite importer dans la plateforme : le Script 42, extrait d'un fichier de mappage, spécifie que les polygones fermés du calque appelé SURFACE forment des pièces, identifiées par le texte présent dans le calque ETIQ-PIECE et associées aux pièces correspondantes par un algorithme de reconnaissance d'inclusion des attributs dans un contour géométrique 2D (comme on traite des fichiers de type plan, la méthode sera souvent géométrique : détection de contour, inclusion...).

```

<OBJECT objId="XXX" src="PIECE" fatherRelative="true" calque="SURFACE"
perfectLoad="true" perfectLoadAction="archiver">
<ATTRIBUTE isId="true" src="NUMERO" catId="XXX" attId="XXX" calque="ETIQ-
PIECE"/>
<ATTRIBUTE isId="true" src="NUMERO" catId="XXX" attId="XXX"
matchMethod="GEOMCONTOUR2D"/>

```

Script 42. Exemple d'attributs utilisés pour le mappage de l'information dans des formats de fichier non-objet



Figure 52. Exemple de pièces à importer dans l'ontologie SIGA3D

La Figure 52 présente une illustration de plusieurs cas de figure ou des pièces d'un plan 2D pourront être détectées et importées dans l'ontologie SIGA3D à l'aide du fichier de mappage décrit précédemment. Cette méthode, décrite à l'aide d'un fichier DWG, peut très bien être utilisée pour des formats de fichiers vectoriels tels GML ou ShapeFile, en spécifiant par exemple les différents couches et/ou fichiers dans lesquels il est possible de trouver l'information requise. Pour GML, comme les primitives géométriques peuvent contenir un ou plusieurs noms (éléments `gml:identifiant` et `gml:name`) et une description (éléments `gml:description`), il n'est pas difficile d'appliquer ce principe à ce format de fichier et créer ainsi des objets non explicitement définis.

L'autre partie importante du processus d'import est le géoréférencement. En effet, dans la plateforme de gestion technique de patrimoine immobilier, une des principales contraintes est la gestion des systèmes de coordonnées géodésiques. Cette partie du monde des SIG est importante pour plusieurs raisons : gestion des

cartes et des modèles numériques de terrain, positionnement cohérent des objets dans les vues 3D et synchronisation spatiale des objets.

Dans SIGA3D, le géoréférencement est présent dans chaque couche de l'architecture, depuis l'acquisition jusqu'à l'exploitation et la visualisation des données par les utilisateurs finaux. Mais avant la phase d'import dans la plateforme, toutes les données ne sont pas géoréférencées. Il faut, dans cette partie, et avant de stocker les objets en base de données, modifier les coordonnées afin de les lier à un emplacement terrestre dans le but d'avoir une vision cohérente du patrimoine géré. Nous avons développé pour cette partie, dans le cadre du projet européen et avec notre partenaire lat/lon, un outil pour géocoder des objets ou pour modifier des coordonnées existantes. Il consiste à charger une carte de la zone où se situe l'objet dont on connaît le système de coordonnées, puis à positionner des points de référence de la représentation de notre objet sur la carte. Les coordonnées sont ensuite calculées, avec ou sans déformation de l'objet, selon plusieurs algorithmes implémentés. Ces informations sont ensuite stockées sous forme de matrice de transformation associée à l'objet (les données originales sont ainsi conservées, utiles pour un export ultérieur éventuel).

Enfin, on peut voir sur le schéma d'architecture qu'il existe une passerelle entre cette couche d'import et le processus de visualisation (flèche entre la couche des sources de données et le moteur de visualisation 3D) ; c'est le mappage dynamique pour une visualisation des fichiers en temps réel qui est représenté : les géométries et le graphe de scène sont construits en temps réel directement depuis les fichiers stockés en base de données ou dans un système de fichiers et référencés dans l'ontologie. Cela permet, entre autres, d'assurer la maintenabilité et l'exactitude des données générées, notamment en cas de mises à jour fréquentes des fichiers (lors des phases de contrôle des relevés du bâti par exemple). Le tout en pouvant bénéficier de la connaissance du modèle global et des mécanismes de cache, celui-ci étant également structuré sur le modèle de l'ontologie, et mis en place (préchargement des représentations, y compris les niveaux de détail, structuration des données suivant un index spatial...).

3 Niveau 3 – Data Model Framework de SIGA3D

Ce niveau de l'architecture contient les outils pour modéliser et construire une ontologie dynamique. Elle est une extension des travaux précédents d'Active3D pour la construction d'un BIM avec C-DMF. Cette couche est détaillée dans la partie de formalisation de l'approche dont l'aperçu ici est rapide. L'information de ce niveau est basée sur des opérateurs qui servent à construire un modèle de données générique (éléments sémantiques, relations et ressources). L'objectif de cette partie est de pouvoir définir une structure de graphe dans laquelle il est possible d'importer différents modèles provenant de formats de données divers telles que les IFC ou CityGML. Ceux-ci peuvent alors être fusionnés (interopérabilité sémantique pour faire correspondre, par exemple, un IFCWall avec un élément CityGML WallSurface). La consistance des modèles définis à ce niveau est vérifiée dans le niveau supérieur, CMF.

Les données sont traitées en fonction de trois grands axes : informations sémantiques, informations géométriques et les informations spatio-temporelles. Cette dernière catégorie est issue des SIG et des besoins spécifiques que nous avons de gérer ce type d'information. Il est notamment possible de les utiliser pour définir des contextes locaux qui autorisent la définition d'un concept de plusieurs manières, dépendamment du contexte local associé : c'est la base de nos travaux sur la multireprésentation et les niveaux de détail contextuels. Les concepts définis par le bloc spatio-temporel de l'architecture vont être réutilisés dans la couche supérieure de l'architecture afin de synchroniser les modèles de données. La synchronisation spatiale et temporelle consiste à définir des propriétés de modèles qui ne varient pas en fonction des systèmes de référence utilisés pour la description des données. On peut ainsi définir des relations entre deux modèles telles que l'adjacence, l'inclusion, l'ordre (avant, après)... ce mécanisme est détaillé plus en détail dans la section de formalisation de la couche contextuelle de l'architecture SIGA3D.

Enfin, la couche géométrique est utilisée à deux desseins : organiser l'information spatiale sous forme de graphe de scène, et définir la structure des index spatiaux. Ce dernier point étant primordial pour les

performances de la plateforme SIGA3D, nous allons faire un rapide rappel sur les index spatiaux et voir comment nous pouvons les gérer dans cette partie.

4 La gestion spatiale des objets urbains

L'objectif de cette section est de comprendre la manière dont sont organisés, au sein de SIGA3D, les objets géographiques. La majorité de ceux-ci sont, à l'origine, indépendants spatialement les uns des autres (c'est-à-dire que lorsque l'on importe un objet depuis une source quelconque, un IFC par exemple, celui-ci ne possède pas forcément de relations spatiales avec les autres objets déjà importés). Le but étant de pouvoir afficher en temps réel des environnements urbains, composés par ces objets géographiques, il faut optimiser les requêtes géographiques afin de répondre aux besoins de requêtes des objets de manière dynamique depuis la base de données et d'affichage des scènes.

La scène de l'environnement 3D que l'on souhaite afficher est gérée de manière hiérarchique afin de pouvoir utiliser les mécanismes de simplification de scènes que nous décrivons dans le paragraphe sur les niveaux de détail hiérarchiques. Cette hiérarchie lie les objets que l'on souhaite afficher par des relations de contenance. Ces relations, qui n'existent pas lorsque l'on acquiert l'objet (créé manuellement ou importé automatiquement comme décrit dans le paragraphe précédent) peuvent être définies de plusieurs manières, par le biais des index spatiaux.

Les index spatiaux permettent, au sein d'une base de données spatiale, d'optimiser les temps de réponse des différentes requêtes en autorisant une sélection rapide des données sur lesquelles elles portent. La plupart des index utilisés aujourd'hui ont été définis pour l'étude de l'espace à deux dimensions, mais peuvent s'adapter à la troisième dimension.

Selon (Ramos, 2003), l'optimisation se fait au travers de deux objectifs :

- La définition d'une structure de données pour avoir un accès plus rapide à l'information,
- Le regroupement d'information afin que les données qui sont proches dans la réalité le soient également en mémoire.

Nous allons donc présenter quelques index spatiaux qui sont utilisés pour stocker et faire des requêtes sur des données géographiques puis nous présenterons la solution retenue au sein de la plateforme Active3D.

- **L'Octree** est l'extension 3D du quadtree. Ce dernier permet de créer une hiérarchie en décomposant un espace à deux dimensions en quatre parties (quadrants) de manière récursive. L'octree est une simple généralisation de la hiérarchisation à l'espace 3D, avec cette fois-ci huit sous-espaces à chaque nœud. La Figure 53 illustre ce principe sur deux niveaux. Le principe de subdivision de l'espace comme montré ici est un principe commun à la majorité des mécanismes d'indexation de l'espace.
- Une autre méthode pour l'indexation des objets est la transformation de l'espace 2D et 3D vers un espace à une dimension par une **courbe mathématique** (on cherche une courbe qui passe par tous les points d'un espace). On peut citer en exemple les courbes de Peano et d'Hilbert (Housseno, 2009). Cette approche définit les quadtree linéaires dans lequel chaque feuille du quadtree est codée de manière à pouvoir retrouver sa position (on utilise une clé pour le codage, par exemple la clé de Peano⁷⁵ qui entrelace les bits de la représentation binaire des coordonnées géographiques).
- Les **arbres BSP** (Binary Space partitionning) constituent un index spatial adapté spécifiquement à la 3D. Cet algorithme a été beaucoup utilisé dans les jeux vidéo pour calculer les espaces « pleins » et « vides ». L'arbre BSP permet de subdiviser l'espace récursivement par rapport à un plan : chaque nœud de l'arbre correspond à un plan quelconque, souvent choisi en fonction de la scène, et stocke l'ensemble des polygones contenus dans ce plan.

⁷⁵ La clé de Peano est utilisée pour fusionner les coordonnées x et y d'un point, de telle sorte qu'il est possible de stocker la valeur dans un tableau à une dimension.

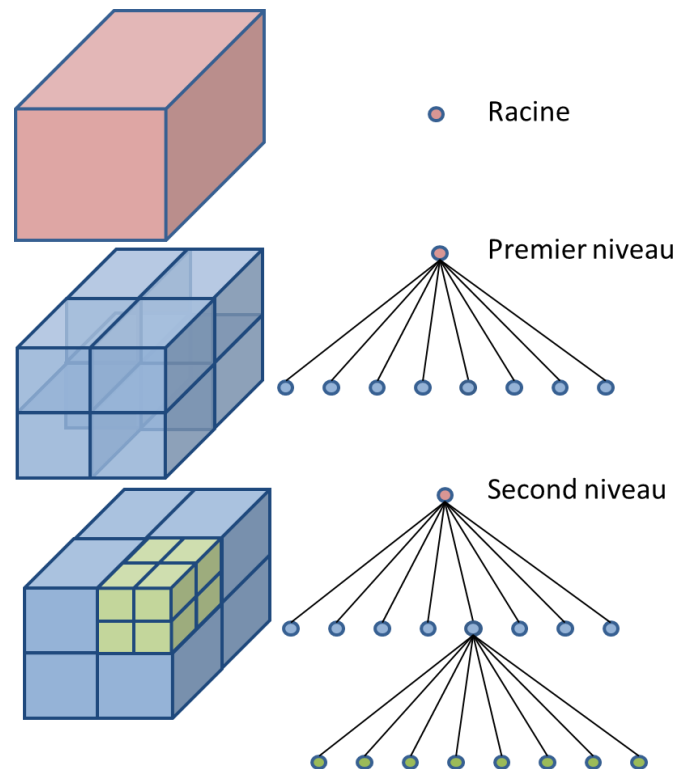


Figure 53. Illustration d'octree sur deux niveaux

- Le principe des index **R-Tree** (Guttman, 1984) est similaire au B-Tree (Balanced Tree), adapté au domaine spatial. Ils permettent d'indexer de l'information multidimensionnelle (comme les coordonnées d'une donnée géographique par exemple) et sont toujours équilibrés. Les rectangles englobants minimaux (Minimum Bounding Rectangles - MBR) des objets spatiaux, que l'on peut faire évoluer vers des boîtes englobantes minimales (Minimum Bounding Box - MBB - composée des coordonnées minimales et maximales - dans les trois dimensions - de chaque objet) dans le cas de données tridimensionnelles, sont regroupés dans les feuilles selon un critère de proximité. Dans le R-Tree de base, un objet spatial est inséré dans une seule feuille, mais les nœuds internes, représentant des portions de l'espace, peuvent se recouvrir. Les caractéristiques principales d'un R-Tree peuvent être résumées comme suit : le R-Tree est une collection de tuples, chacun possédant un identifiant unique ; les feuilles contiennent un tuple de la forme (MBB, O_i) , où MBB est la boîte englobante minimum d'un objet O_i . Les nœuds non-terminaux sont représentés par d'autres tuples de la forme (MBB, R_i) , où MBB est la boîte englobant l'ensemble des niveaux inférieurs. Le R-Tree est un arbre équilibré avec une hauteur maximale de $(\log_m N)-1$ où N est le nombre d'objets à inclure et m le nombre minimal de feuilles pour chaque nœud. D'après (Zlatanova, 2000), les R-Tree peuvent être utilisés à deux fins : une gestion simplifiée des LoD et l'accélération des traitements en base de données. Le premier objectif est réalisé grâce à la structure même de l'arbre : les feuilles contiennent la représentation détaillée de l'objet correspondant, le niveau supérieur qui correspond à la MBB de l'ensemble de ces fils peut également posséder une représentation globale de ceux-ci... Cette définition se rapproche de celle des H-LoD (LoD Hiérarchiques) que nous avons abordés précédemment.

Notre approche au sein d'Active3D se base sur ce dernier index spatial. Cette technique d'indexation est en effet la plus proche de la gestion que nous avons des objets avec les relations de contenance spatiale.

Cependant, notre approche, notamment avec les IFC, nous permet d’avoir des relations spatiales naturelles et souvent beaucoup plus logiques que ce qui peut être calculé par un algorithme de R-Tree. Le fait que les relations que nous définissons soient plus naturelles implique que lorsqu’un utilisateur va utiliser la maquette numérique d’Active3D pour naviguer dans son environnement, les requêtes effectuées seront à peu de choses près basées sur les relations définies (par exemple l’utilisateur va naviguer depuis un site vers un bâtiment, puis du bâtiment se diriger vers un étage).

Pour pallier à ce manque de logique humaine lors de l’indexation des objets, nous nous basons sur une vue figée du modèle de données utilisé (tout comme l’import, un paragraphe donnera une définition des vues du modèle de données) qui définit des relations de contenances par type d’objets. Ainsi, lors de la création d’objets dans le modèle de données, ceux-ci seront automatiquement rattachés spatialement à l’objet le plus proche géométriquement et dont le type est en accord avec les relations définies dans cette vue figée.

La Figure 54 montre un exemple de correspondance entre la vue figée et le graphe de scène. On remarque que tous les objets du graphe de scène ne font pas partie de la vue figée, certains objets étant considérés comme décor de la scène (et ne sont pas définis dans le modèle de données de l’utilisateur).

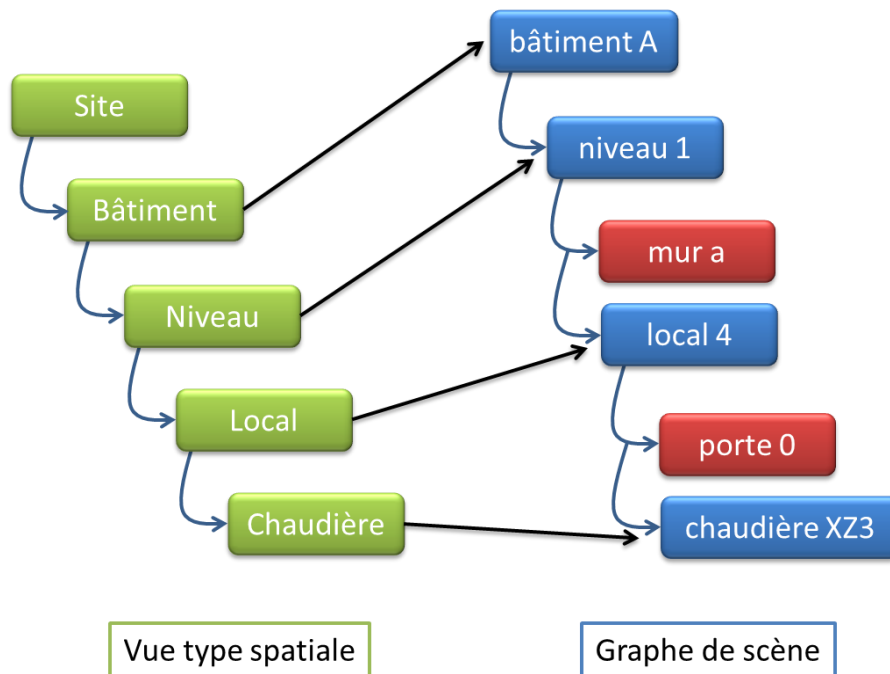


Figure 54. Exemple de correspondance entre la vue figée et le graphe de scène

Lors de l’import de données, les relations spatiales devront systématiquement être respectées. Ainsi, si l’on importe un local qui, dans son modèle original (par exemple IFC) est rattaché directement au bâtiment, il faudra que l’on construise un niveau puisque cette relation est définie dans la vue figée. Chaque objet importé possède une boîte englobante calculée lors de l’import des données. De plus, la position de l’objet correspond à ses coordonnées monde. Cette vue spatiale figée sert également à la création d’objets manuels. En effet, lorsque l’on va créer manuellement un objet depuis le moteur graphique, en fonction de sa position plusieurs choix de relations spatiales peuvent être proposés (il peut être par exemple à la fois dans un site et dans un bâtiment, un étage, une pièce...). On choisira alors automatiquement le premier objet dont le type est rattaché spatialement au type d’objet que l’on vient de créer dans la vue spatiale.

5 Niveau 4 – La couche de gestion du contexte

Cette couche a pour rôle de gérer le contexte des différents modèles de données définis dans les niveaux inférieurs de l'architecture. Le contexte d'un modèle correspond à toutes les informations liées au modèle comme nous allons le rappeler dans le prochain paragraphe. Pour SIGA3D, la notion de contexte a été élargie par rapport à la définition de C-DMF et inclut deux nouveaux blocs correspondant à la gestion des référentiels spatiaux et temporels utilisés pour la définition des données, et à la synchronisation spatio-temporelle des données. Si la gestion des référentiels est assez naturelle, le bloc de synchronisation peut paraître surprenant. Nous allons détailler ces deux points plus en détail dans les paragraphes suivants.

5.1 Le contexte

Afin de fixer les idées sur la gestion du contexte, nous allons faire un rapide rappel sur ce qu'est le contexte et comment il est possible de le gérer. Pour cela, nous nous basons sur la définition de Guha : «La notion de contexte est fondamentale en sémantique et dans la définition des ontologies. Si la syntaxe est entièrement dans le texte, la sémantique est dans le contexte. La définition d'un contexte est chose difficile et multidimensionnelle : il y a le contexte immédiat et littéral du mot - le texte autour du mot - le contexte culturel, économique, géographique d'un document... La compréhension d'une information par un lecteur humain dépend dans une très large mesure de sa connaissance du contexte. L'une des clefs de la réussite du Semantic Web sera sa capacité à faire passer dans les machines cette notion très subtile. » (Guha, 2004).

De fait, il existe de nombreuses solutions pour gérer les contextes dans les graphes RDF. L'une d'elles est le mécanisme de réification qui consiste à transformer un triplet (*subject, predicate, object*) en quatre triplets (*x, type, statement*), (*x, subject, s*), (*x, predicate, p*), (*x, object, o*). Ainsi la ressource *x* peut être utilisée dans d'autres triplets. Cependant, même si la réification permet la représentation contextuelle des triplets, ce mécanisme s'avère assez lourd dû à la multiplication des triplets. Ce n'est donc pas la solution retenue pour C-DMF. Une autre proposition plus pertinente pour les objectifs fixés est celle des Named Graphs qui consiste à ajouter une valeur supplémentaire aux triplets pour les transformer en quadruplets.

Voici quelques exemples dont la capacité de définition de méta information (c'est-à-dire du contexte) associée aux graphes RDF est requise:

- Distribution de données (Data syndication) : le système a besoin de garder une trace de la provenance des informations.
- Information de restriction d'usage (Restricting information usage) : les producteurs d'informations peuvent vouloir attacher des droits de propriétés intellectuelles ou leurs préférences privées aux graphes afin de réduire l'utilisation des informations publiées.
- Contrôle d'accès (Access Control): Un entrepôt de triplets peut requérir un contrôle d'accès fin, qui apparaît comme un ensemble de métadonnées associées aux graphes dans l'entrepôt.
- Version et évolution d'ontologie (Ontology versioning and evolution)

Un Named Graph est un graphe RDF auquel on assigne un nom sous la forme d'une URI, soit au sein du graphe, soit dans un autre graphe. Les Named Graphs apportent une nouvelle dimension aux graphes RDF et les informations sont décrites non plus avec des triplets, mais avec des quadruplets. Cette dimension supplémentaire procure aux graphes RDF une identification permettant ainsi d'être utilisée comme une ressource RDF dans un graphe. Des triplets permettent déjà de modéliser quantité de problèmes, des quadruplets apportent une autre dimension qui encapsule chaque triplet. C'est ainsi que l'on peut contextualiser les informations.

La définition formelle des Named graph peut être trouvée dans la Définition 1.

U = l'ensemble des URIs
 B = l'ensemble des nœuds anonymes (blank node)
 L = l'ensemble des littéraux RDF
 $V = U \cup B \cup L$ = l'ensemble des nœuds
 $T = U \times V \times U$ = l'ensemble des triplets RDF
 G = un ensemble de T
 Un Named Graph est un couple $ng = (n, g)$ avec n dans U et g dans G.

Remarques :

- un littéral peut être un sujet.
- U, B et L sont mutuellement disjoints.
- Les nœuds anonymes ne peuvent pas être partagés entre différents Named Graphs

Définition 1. Définition formelle du named graph.

Le Script 43 est un exemple de Named Graph dans la syntaxe TriG⁷⁶ proposé par (Carroll, 2005). Les préfixes de définition d'espaces de noms sont omis.

```
:G1 { :b1 rdf:type mb:Batiment .
      :b1 mb:name "Batiment A" .
      :G1 ma:autorisation :Architect }

:G2 { :G1 ex:date "2003-09-03" }
```

Script 43. Exemple de Named Graph

Cette approche contextuelle constitue une démarche globale ; plutôt que d'avoir une réflexion pour chaque besoin, Renaud Vanlande a eu une réflexion globale pour la gestion collaborative d'un système d'informations du bâtiment durant tout son cycle de vie pour répondre à diverses questions : Comment adapter mon réseau d'informations selon l'utilisateur ? Comment gérer les données dans le temps ? Comment fusionner des données provenant de modèles différents ? Comment gérer l'évolution d'un modèle de données ?

Par exemple, que ce soit un utilisateur, un domaine d'application, ou d'autres types de point de vue, le principe reste toujours identique, nous devons gérer des informations par rapport à un contexte. En effet, pour une utilisation dans un environnement collaboratif, le point de vue des utilisateurs est différent. Adapter la présentation des données à l'utilisateur, délimiter son champ d'action sur les informations, sont des mécanismes qui permettent de gérer, manipuler efficacement le réseau d'informations. Au-delà d'un contexte multi-utilisateur, nous avons dit que suivant un contexte, une définition de concept était différente. La définition d'un concept dépend du domaine d'application dans lequel il est défini. Une coccinelle est un insecte pour le domaine des sciences de la nature et une voiture dans le domaine de l'histoire de l'automobile. La notion de contexte est importante au niveau du modèle, mais également au niveau des données. « François Fillon est le Premier ministre de la France », cette affirmation ne sera pas toujours vraie, un contexte temporel permettrait d'apporter plus d'informations sur cette affirmation.

Si l'approche globale est intéressante pour les problématiques liées au bâtiment, les limites s'en font ressentir pour la gestion d'espaces touchant à l'environnement urbain. En effet, nous n'avons plus un seul modèle de bâtiment à gérer à la fois, mais plusieurs modèles hétérogènes comprenant modèles de bâtiments, modèles d'environnements urbains et gestion des réseaux et des éléments urbains. Cette multiplicité des modèles et

⁷⁶ TriG est une spécification de l'université de Berlin pour décrire en texte brut les Named Graph et RDF de manière plus concise que XML. <http://www4.wiwi.fu-berlin.de/bizer/TriG/>

des objets dans ces modèles qu'il nous faut traiter, leur nature géographique avec ce que cela implique dans l'évolution des objets dans l'espace et dans le temps (ces paramètres étant susceptibles d'influencer plus souvent leur définition en fonction de leur positionnement et de l'époque qu'un bâtiment dont les changements qui interviennent sont plus lents et modérés) demande de gérer les objets et leur contexte à un niveau plus local. C'est le rôle des contextes locaux que nous avons présentés dans les niveaux inférieurs de notre architecture. Par exemple, si l'on souhaite modéliser un domaine skiable (exemple repris de (Cullot et al., 2003)), de nombreux paramètres sont dépendants de la saison (neige en hiver), du temps (pas de neige en hiver), des tronçons de piste existants, praticables ou interdits... De plus, le fait de traiter, dans notre architecture, des ensembles de modèles conduit à traiter les relations qu'il peut y avoir entre ces modèles, et notamment celles qui sont spécifiques à certains modèles, de sorte que si l'on fait des modifications dans ceux-ci ou si l'on change des informations sur les systèmes de référence des modèles, ces relations décrites ne soient pas modifiées.

L'approche de gestion du contexte dans SIGA3D reprend donc les concepts de gestion globale de C-DMF, complétés par la gestion des contextes au niveau local, puis étendus avec des éléments spécialisés issus des systèmes d'information géographique. C'est ce que nous allons voir dans le paragraphe suivant traitant de la gestion des systèmes de référence, aussi bien pour l'espace que pour le temps, et la synchronisation des modèles de données dans ces dimensions.

5.2 La gestion des référentiels et la synchronisation spatio-temporelle

La couche de gestion du contexte a été spécialisée dans l'architecture SIGA3D et ne sert plus uniquement à la description du contexte global comme nous l'avons décrit dans le paragraphe précédent, à propos de C-DMF. En effet, l'un des aspects nouveaux de l'architecture SIGA3D par rapport au BIM est la gestion des objets géographiques. La gestion des coordonnées de ces objets nécessite la définition d'un système de coordonnées terrestre de référence. Comme nous l'avons présenté dans l'état de l'art, les sources de données sont hétérogènes et les systèmes de référence géodésique nombreux.

Nous avons fait le choix, pour notre architecture, de ne pas imposer l'usage d'un seul et unique système de coordonnées pour les sources de données, mais d'offrir la possibilité de transformer les coordonnées des objets en fonction d'un système de référence définis dans le contexte d'utilisation de ceux-ci. Le système de coordonnées d'utilisation est spécifié à ce niveau de l'architecture. Les transformations interviennent à deux niveaux :

- Comme nous l'avons vu précédemment, lors du stockage des objets en base de données. Cela permet de charger directement les géométries des objets avec les coordonnées correspondant au système géodésique du modèle de données, et évite ainsi les transformations lors de la phase de visualisation, ce qui serait possible, mais ralentirait de manière significative les performances lors de la visualisation des données directement depuis la base de données géométrique.
- Cependant, il est possible de transformer également les coordonnées lors de la phase de chargement des données, notamment dans le cas où les géométries et les objets sont chargés depuis des fichiers (IFC, CityGML...) ou des web services et où le système de référence défini dans le contexte courant n'est pas celui du fichier ou de la source de données. Le fonctionnement du système de référence est sensiblement identique pour le temps avec la définition du fuseau horaire.

Les relations spatio-temporelles définies entre les modèles pour leur synchronisation sont indépendantes de ces systèmes de référence. Il s'agit, dans ce bloc, de définir des relations entre les modèles afin de gérer au mieux les déformations induites lors de la modification et des systèmes de coordonnées et des transformations qui en découlent. Ces propriétés sont également utilisées pour vérifier la cohérence lors des modifications manuelles et mises à jour des modèles. Par exemple, si on s'intéresse à un modèle de données représentant une maison, puis à un autre modèle de données représentant un garage décrit sous forme d'annexe à la

maison, on peut définir deux relations de synchronisation entre ces deux modèles : l'annexe n'existe que si la maison existe (relation temporelle « pendant »), et l'annexe est tangente à la maison (relation topologique « tangent »). Ainsi, si l'on affiche une scène 3D antérieure à l'existence de la maison, on sait que le modèle de l'annexe ne peut pas être affiché, car non valide, même si aucune date de validité de ce modèle de données n'a été définie au niveau du contexte de ce modèle. De plus, on sait que si la maison est affichée, et que pour une raison quelconque elle est repositionnée (par un changement de système de référence géodésique par exemple), le modèle de l'annexe devra être spatialement repositionné pour garder la relation topologique de tangence décrite au niveau du modèle.

Dans notre architecture, les relations possibles ont été simplifiées (nous n'en définissons que certaines, les plus courantes comme décrites dans la formalisation). De plus, le niveau topologique porte sur les boîtes englobantes des objets afin de ne pas surcharger le rendu avec le calcul de relations complexes.

6 Niveau 5 – La couche de connexion

Cette couche est la couche de préparation des données avant leur affichage, aussi bien graphique (2D/3D), que alphanumérique (vue sémantique). On y retrouve le bloc de transformation des coordonnées dont nous avons déjà parlé. Il est essentiellement utilisé lorsque la source de données qui doit être affichée n'est pas dans le bon système de coordonnées et qu'aucune matrice de transformation ne lui est associée. Lorsque les géométries proviennent directement de la base de données, les transformations ont souvent été préalablement calculées et stockées sous forme d'une matrice de transformation associée à l'objet. Mais ce n'est pas forcément le cas lorsque la géométrie ou les informations attachées à l'objet requêté font référence à un fichier. Dans ce cas, la transformation des coordonnées est réalisée grâce à un outil que nous avons développé en partenariat avec lat/lon. Le bloc qui gère le streaming fait le lien entre les requêtes de l'interface utilisateur et la base de données qui contient à la fois les graphes sémantiques et géographiques. Son but est de stocker à la volée les informations nécessaires dans le cache de ce niveau. Le cache permet de stocker des extraits de la base de données afin d'accélérer les traitements tels que les transformations des coordonnées et surtout l'affichage. Il permet également d'accélérer le rendu des scènes ou des vues qui ont déjà été visualisées peu de temps auparavant. Il stocke les données de contexte courant, de modèles, des contextes de modèles, des extraits de graphes sémantiques et du graphe de scène correspondant au contexte local.

7 Conclusion

L'ensemble des couches présentées dans ce chapitre permet de construire une plateforme collaborative de gestion technique de patrimoine Urbain. Pour compléter ces couches, nous avons développé des interfaces de visualisation des informations. Cette visualisation est réalisée sous la forme d'une maquette numérique 3D et d'une arborescence des connaissances affichées selon le contexte et le profil de l'utilisateur au moment de la visualisation. La Figure 55 montre l'interface résultante de nos travaux. Cette interface est intégrée à l'ensemble des outils développés par Active3D. Elle permet la visualisation de plusieurs bâtiments IFC dans une représentation 3D. Ces bâtiments sont positionnés dans la scène selon leur géoréférencement et ils coexistent avec des objets urbains (ici des arbres et des bancs) qui ne sont pas modélisés dans la norme IFC. Le mode de fonctionnement d'Active3D a été respecté. Il est possible d'accéder directement à la fiche descriptive d'un objet, soit directement à partir de l'arborescence de scène soit à partir de la maquette 3D.

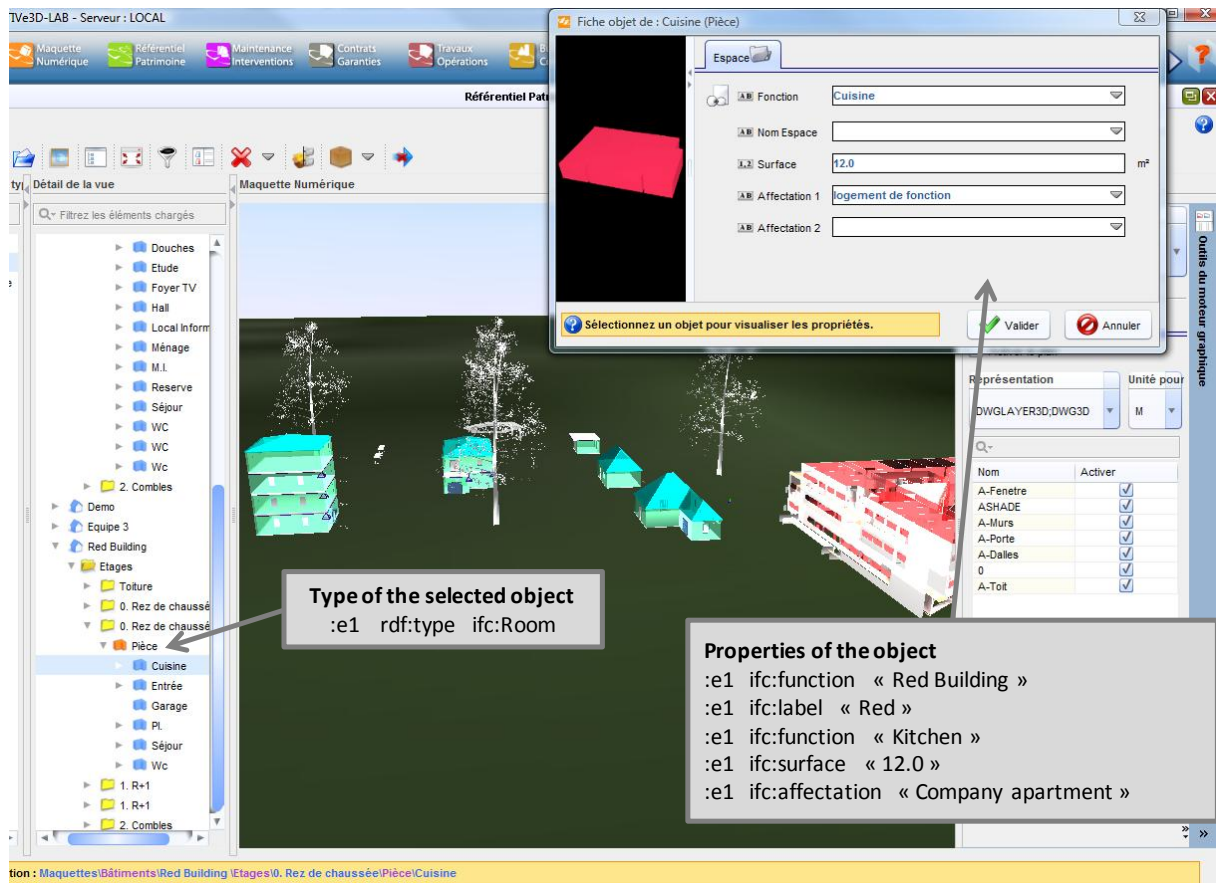


Figure 55. Exemple de visualisation de patrimoine urbain dans SIGA3D

Chapitre 6

Développement industriel

Résumé

Ce dernier chapitre présente les développements industriels que j'ai réalisés dans le cadre du projet Euréka SIGA3D. Lors de ces développements, j'ai coordonné une équipe de développeur au sein de la société Active3D et une équipe de développeurs dans la société Lat-Lon. Outre ces aspects de coordination, j'ai développé une grande partie des différents niveaux présentés dans l'architecture. L'ensemble des développements réalisés qui sont présentés dans ce chapitre concerne l'implémentation de l'ontologie en base de données, le développement d'un nouveau moteur 3D et le développement d'un streaming pour la gestion des niveaux de détail sémantiques.

Plan

1	Les processus autour de l'Urban Facility Management	153
1.1	Processus de modélisation	155
1.2	Processus d'acquisition des données	155
1.3	Processus d'import	156
1.4	Processus de géoréférencement	158
1.5	Stockage des données	159
1.6	Les processus de streaming et de visualisation	160
2	Le projet européen SIGA3D	162
2.1	Préparation technique d'Active3D	163
2.2	Gestion des coordonnées	164
2.3	Modélisation du terrain et gestion des objets urbains.....	165
3	Implémentation SIGA3D.....	168
3.1	Le développement d'un nouveau moteur graphique.....	169
3.2	Stockage du graphe de scène	173
3.3	Définition de l'ontologie	179
4	Conclusion	180

Ce dernier chapitre présente l'environnement applicatif mis en place pour l'exploitation de l'architecture décrite dans les chapitres précédents. La première partie présente le cadre de développement du projet. En effet, le projet SIGA3D est à la fois un projet de recherche, celui de la thèse, mais également un projet européen. SIGA3D, plus qu'une architecture, est un véritable ensemble de processus permettant le travail collaboratif autour de la gestion technique de patrimoine immobilier et urbain. Nous pouvons distinguer deux types de processus : les processus métier autour de l'UFM (Figure 56), et les processus propres à l'UIM.

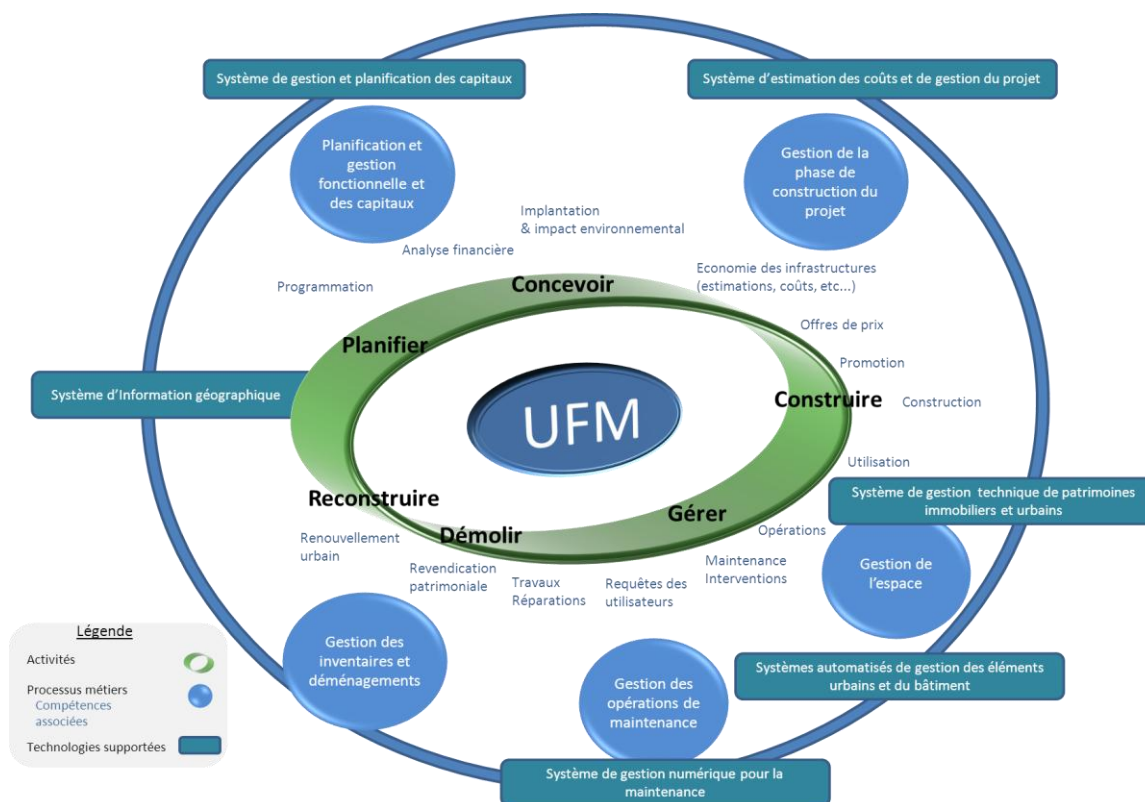


Figure 56. Urban Facility Management, processus métier et gestion

Ce chapitre est articulé en 3 parties. La première partie présente une architecture de processus autour de laquelle le modèle d'information urbain a été conçu. Les processus sont présentés dans l'ordre chronologique d'utilisation, c'est-à-dire depuis la modélisation des modèles et des données jusqu'à la visualisation des données importées puis stockées. La seconde partie présente le projet européen, ses objectifs et ses phases d'évolution, dont le but est une réflexion et une implémentation de la partie géographique du projet. Nous aborderons ensuite les développements effectués autour du grand projet final SIGA3D (travaux de recherche + projet européen). Celui-ci peut se découper en trois parties : la préparation technique de la plateforme Active3D pour supporter les évolutions du projet (choix d'un moteur de rendu, base de données graphique, mise en place des LoD), la gestion des coordonnées (outils de géoréférencement et de transformation de coordonnées), et la gestion particulière des objets externes au bâtiment.

1 Les processus autour de l'Urban Facility Management

Cette première partie est l'occasion de présenter plus en détail l'idée qui est faite par les professionnels d'un système de gestion technique de patrimoine urbain, ce que nous avons appelé Urban Facility Management en référence à la gestion technique de patrimoine immobilier, le Facility Management (FM). L'ensemble (non-exhaustif) des processus est représenté dans la Figure 58. Afin d'avoir une idée de l'évolution qu'il y a eu

autour de la plateforme Active3D, la Figure 57 montre la même architecture de processus de la plateforme Active3D, mais avant le projet SIGA3D.

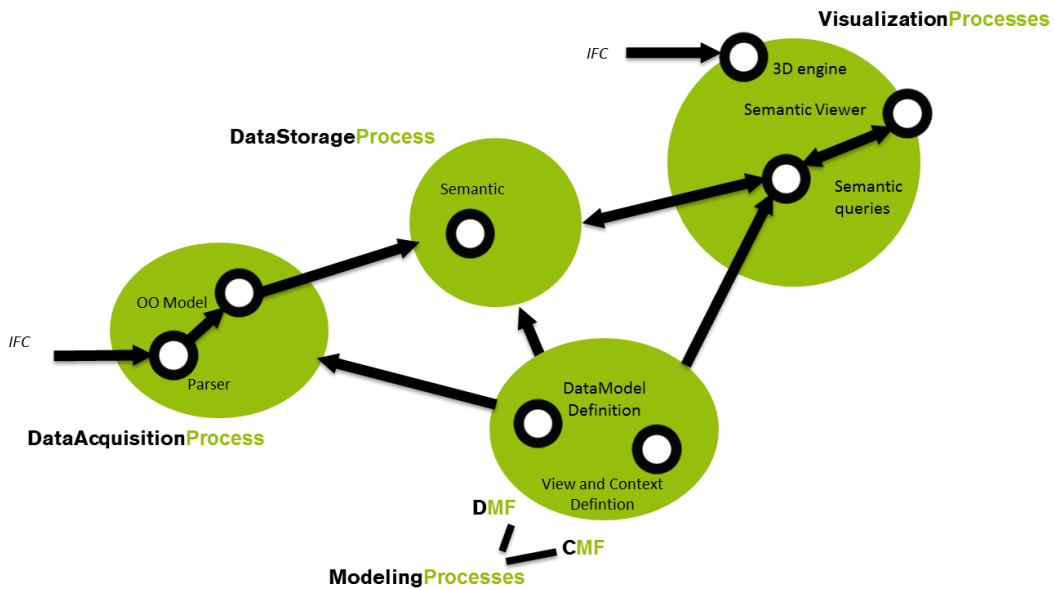


Figure 57. Architecture des processus de l'application Active3D avant le projet SIGA3D

L'idée est d'avoir une plateforme centrale pour gérer le travail collaboratif tout au long d'un projet de construction urbain. Les acteurs qui interviennent et les processus métiers qui y sont définis sont très proches de ce que l'on peut trouver autour du BIM et des plateformes de FM puisque ceux-ci font partie intégrante de la gestion technique urbaine. Les évolutions se font au niveau des dimensions du projet, du nombre et de la diversité des acteurs, mais aussi des relations avec l'environnement urbain existant, passé ou à venir. Afin de réaliser dans les meilleures conditions l'ensemble de ces processus, un modèle d'information urbain (Urban Information Model, UIM) est construit. Le prochain paragraphe détaille les différents processus nécessaires à la création, l'acquisition des connaissances et leur exploitation au sein de l'UIM.

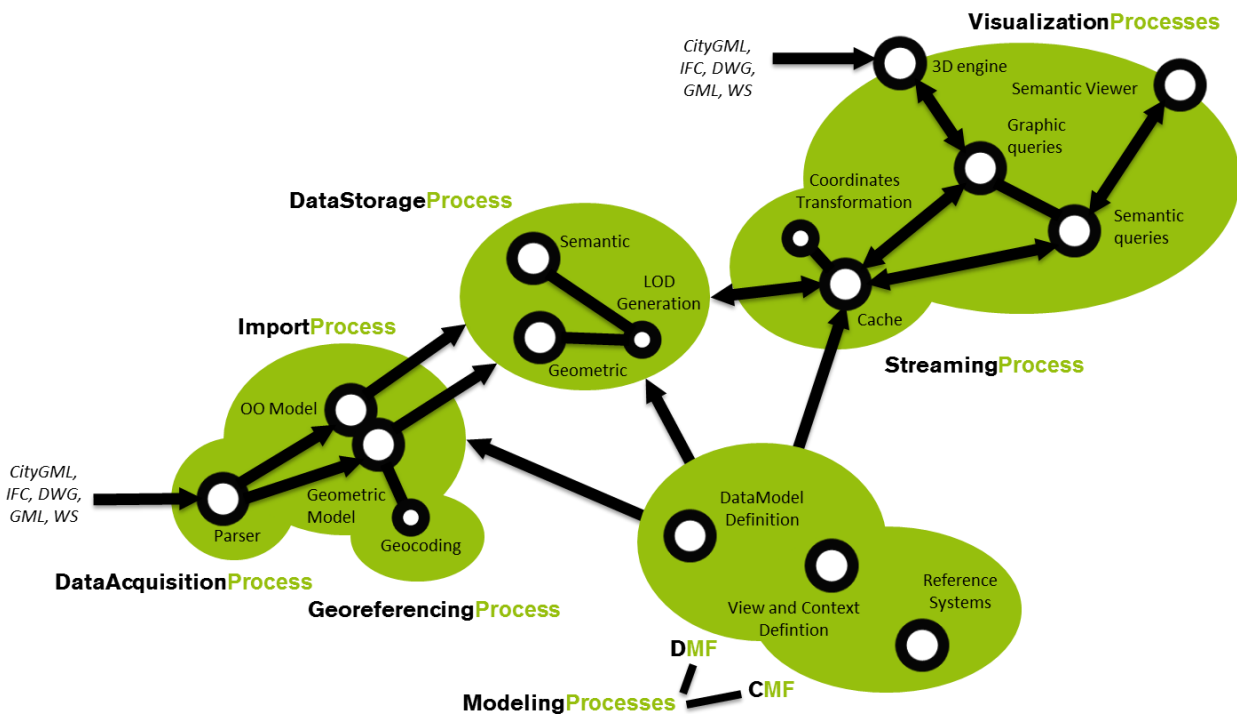


Figure 58. Architecture de processus

1.1 Processus de modélisation

La première étape d'utilisation de l'UIM est la création de modèles de données qui vont structurer les données importées en base de données. Cette partie consiste à construire une ontologie dynamique qui sera peuplée ultérieurement, soit par un import automatique des données, soit par leur création manuelle. Pour cela, on utilise les outils décrits dans l'architecture SIGA3D dans les couches DMF et CMF (gestion du modèle de données et du modèle de contexte).

La couche de modélisation des données permet de définir, à l'aide des opérateurs de construction du modèle de données, et du mécanisme de multireprésentation géométrico-sémantique, la partie structurelle du graphe sémantique. Les éléments peuvent être structurés autour des concepts d'espace et de temps. Il y a deux niveaux d'intégration de tels éléments dans l'ensemble de processus de modélisation. La première partie consiste à définir les éléments spatiaux et temporels tels que les points, ligne, instant, intervalle... La modélisation du temps est basée en partie sur OWL-Time (Hobbs and Pan, 2006). La partie spatiale, quant à elle, est basée sur la norme GML. Le second niveau d'utilisation des éléments spatio-temporels consiste à définir des relations entre objets, telles que adjacent, dedans, pendant, avant... le premier niveau est géré dans la définition du modèle de données alors que le second est utilisé dans la définition du contexte.

La définition du contexte est le processus qui consiste à définir un contexte à la fois sur les modèles de données et sur les données, en utilisant des opérateurs de graphe afin de simplifier l'écriture des graphes contextualisés. C'est également à ce niveau du processus qu'on définit, pour chaque modèle, les systèmes de référence utilisés. Un des résultats de ce processus est un ensemble de vues personnalisées qui vont être utilisées dans les processus de visualisation pour afficher les données en fonction du contexte utilisateur, aussi bien dans la partie sémantique (arbre alphanumérique dont nous parlons plus loin dans le document), que dans la partie graphique (notamment de par l'utilisation des C-LoD).

Le contexte décrit également des relations spatio-temporelles sur les graphes de la couche de modélisation des données.

1.2 Processus d'acquisition des données

Une fois que les différentes informations sur les modèles données et les contextes sur ceux-ci ont été définis, il faut peupler l'ontologie ainsi créée. Pour cela, deux possibilités : soit les données sont créées manuellement depuis l'interface de visualisation des données, soit les données sont importées automatiquement depuis des sources de données formalisées telles que des fichiers IFC, CityGML, DWG ou des web services. Cependant, ces différentes sources étant structurellement hétérogènes, il est nécessaire de les comprendre avant de les importer et les organiser de la manière dont l'ontologie pourra les exploiter. C'est le rôle du processus de parsing qui vise à analyser lexicalement et syntaxiquement les sources de données pour en récupérer les objets. Le processus est plus ou moins simple et rapide en fonction du type de source traité. Ainsi, les sources de données spécifiques au domaine du BIM, plutôt orienté sur l'échange de fichier comme les IFC, se fait simplement puisqu'il suffit de retrouver tous les objets contenus dans le fichier. Par contre, pour les données issues plus spécifiquement de la CAO (Conception Assistée par Ordinateur), où l'on trouve des fichiers contenant en grande majorité de la géométrie, sans objet, le processus est un peu plus complexe puisqu'il faut utiliser des algorithmes de reconnaissance des géométries afin d'en extraire les objets. Enfin, dans le domaine du SIG, ce sont principalement des web services qui vont être utilisés, pour lesquels l'information diffusée est souvent déjà formatée de manière compréhensible. Le résultat de ce processus est un ensemble d'objets, soit dans le sens du paradigme orienté-objet (issus des fichiers IFC ou CityGML par exemple), soit des objets géométriques (points, lignes, polygones...) sans autre connaissance particulière (issus des fichiers DWG ou GML), qu'il faut analyser pour peupler correctement l'ontologie.

1.3 Processus d'import

Une fois que les données ont été parsées et les différentes composantes reconnues (objets, géométries, relations...) pour chaque source de fichier, l'étape suivante consiste à les organiser selon le modèle de données qui a été défini à l'étape précédente. Pour cela, on construit un modèle mémoire qui reprend la structure de graphe du modèle de données dans lequel on souhaite stocker les objets et données nécessaires au peuplement du graphe de l'ontologie. Durant cette étape, tous les objets et relations sont analysés pour construire un graphe acyclique que l'on appelle arbre contextuel. Celui-ci est construit par l'utilisation de règles métiers telles qu'« une porte est un espace ouvert dans un mur » que l'on a défini lors du processus précédent. Une seconde phase du processus traite plus spécifiquement les géométries. Les objets géographiques sont organisés selon une vue spatiale prédéfinie qui contient des relations, souvent topologiques, entre les objets. Celle-ci constitue la base de notre graphe de scène. Ainsi, on peut avoir par exemple la définition d'un arbre contextuel où un site est composé de bâtiments et un bâtiment est composé d'étages. Lors de la détection et de l'identification des objets géométriques, la vue spatiale sera peuplée de manière cohérente. La création de géométries fonctionne aussi bien pour les données issues de formats de fichiers non objet, notamment lorsque l'on souhaite peupler l'ontologie à partir de fichiers anciens qui n'ont pas été décrits à l'aide d'objets, ce qui est actuellement le cas d'une majorité d'utilisateurs. Les projets objets tels que IFC sont utilisés pour des constructions récentes ou dans le cadre d'un relevé complet du patrimoine bâti, urbain ou immobilier. À ce niveau du processus, toutes les géométries définies dans l'arbre contextuel sont converties en un modèle surfacique triangulé qui sera directement affiché par le moteur 3D. Durant cette phase, tous les objets 3D sont associés à un identifiant unique (GID). Dans les formats de fichiers objets tels que IFC et CityGML, cet identifiant existe déjà et il nous suffit de le récupérer, ce qui permet un lien fort entre les fichiers sources et les géométries traitées dans SIGA3D, et facilite notamment les mises à jour de modèles grâce à cette identification (on parle de GUID, Identifiant Global Unique, qui signifie que chaque ressource est identifiée de manière unique dans le monde, et en cas de modification d'un fichier, les identifiants des objets ne changent pas afin de pouvoir les retrouver ultérieurement). Cet identifiant est utilisé pour identifier chaque objet métier et lier sa représentation géométrique de notre graphe de scène à sa représentation sémantique dans le graphe de l'ontologie.

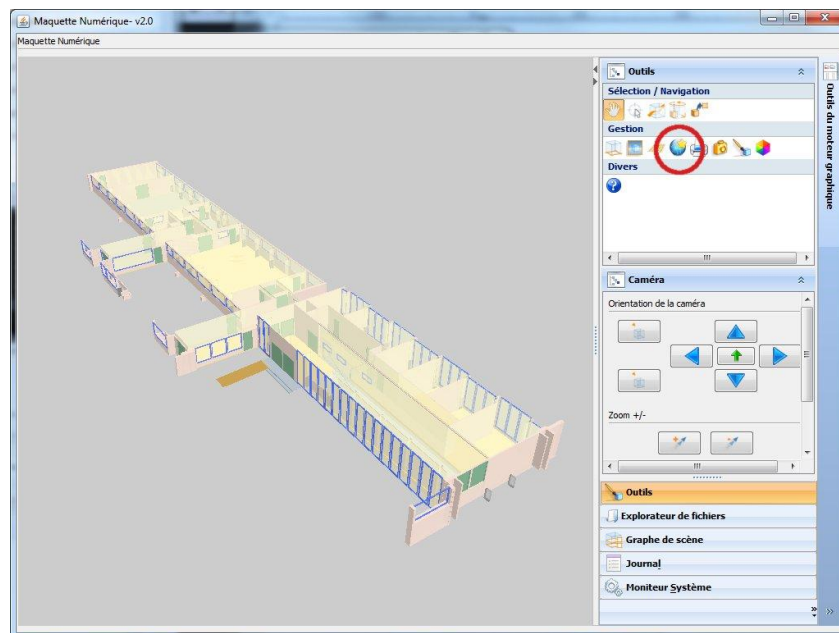


Figure 59. Outil de géoréférencement

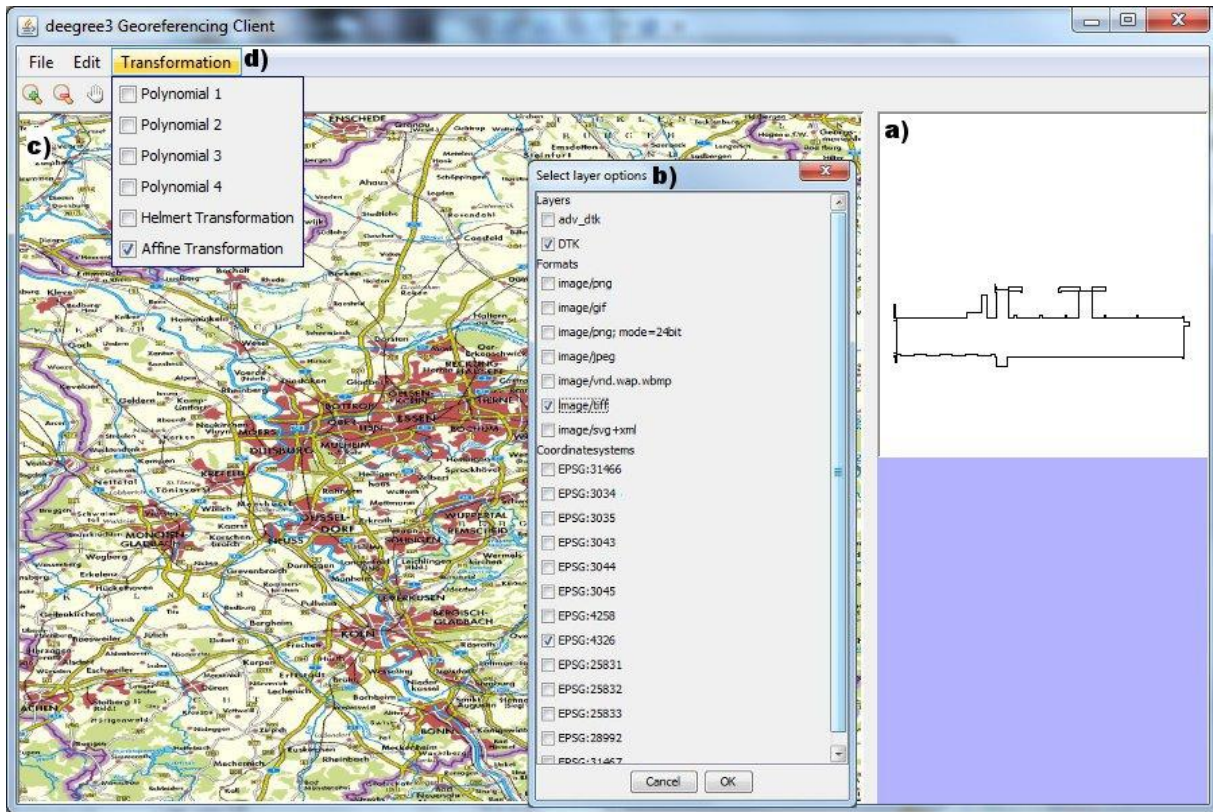


Figure 60. Processus de géoréférencement.

- a) Calcul de l'empreinte au format GML.
- b) Sélection du format de l'image et du système de coordonnées.
- c) Affichage du plan support. d) Sélection de la méthode de transformation des coordonnées.

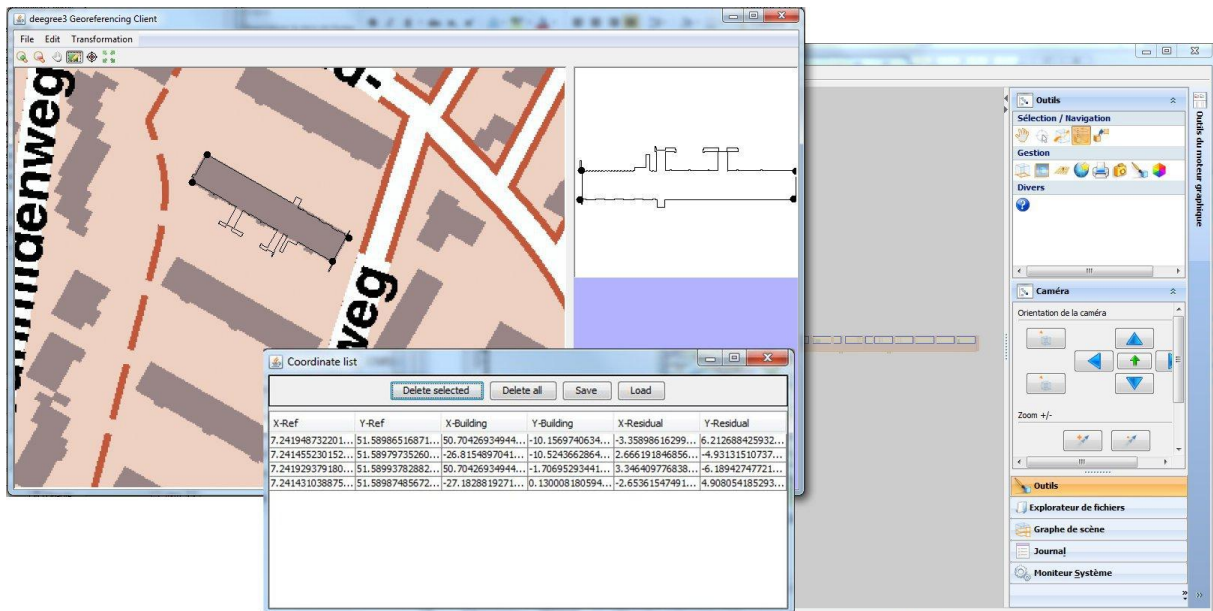


Figure 61. Processus de géoréférencement : positionnement des points de référence et calcul des coordonnées géographiques.

1.4 Processus de géoréférencement

L'un des objectifs du projet SIGA3D est la gestion des objets géographiques. Or, pour avoir des objets géographiques, il faut lier les objets que l'on souhaite traiter, géométriques ou sémantiques, à des coordonnées terrestres. Parfois, les concepteurs de projets urbains ou immobiliers associent les objets à un emplacement sur la Terre, pour pouvoir travailler sur un logiciel de SIG par exemple. Mais la plupart du temps, les plans que nous allons récupérer d'architectes sont créés dans des systèmes de coordonnées locaux, chaque système étant propre au logiciel utilisé pour la conception. C'est le rôle du géoréférencement de lier les représentations des objets à des coordonnées géodésiques.

Pour le projet SIGA3D, il a été décidé après une phase d'étude dans le cadre du projet européen SIGA3D que nous détaillons dans le chapitre suivant, de traiter la problématique du géoréférencement à deux niveaux de l'ensemble des processus que nous décrivons : une phase de géoréférencement à l'import, et une phase de transformation des coordonnées à l'affichage des données lorsque cela est nécessaire. En effet, les plans que nous sommes amenés à traiter dans l'application peuvent être associés à différents CRS (systèmes de coordonnées de référence). De plus, comme il a été détaillé dans la présentation de l'architecture SIGA3D, la partie de définition du contexte contient l'information du CRS utilisé par chaque utilisateur ; ceux-ci peuvent, en fonction de leur profil et de leurs critères de sélection, être amenés à choisir un CRS différent pour la visualisation des données (par exemple utiliser le système de référence géodésique WGS84 pour une utilisation GPS, et le système de projection Lambert 93 pour une utilisation architecturale). De fait, nous stockons les données issues de la phase précédente avec le système de coordonnées auquel il était attaché à l'origine. Cela permet également une réutilisation plus aisée des objets par leurs concepteurs lorsque cela est nécessaire. Nous décrivons plus loin le processus de transformation des coordonnés.

Le processus mis en place pour la transformation des coordonnées est le suivant :

- Lorsqu'un objet est importé dans Active3D, si celui-ci ne possède pas de coordonnées terrestres, nous proposons à l'utilisateur de positionner son bâtiment sur une carte afin de déterminer ses coordonnées (Figure 59).
- S'il accepte, une empreinte au sol en deux dimensions du bâtiment est calculée (algorithme de détection de contour sur une image projetée du bâtiment). Cette empreinte est ensuite convertie en GML afin d'échanger de manière normalisée entre les différents outils développés pour cette phase du projet. À ce niveau, avec les résultats du calcul de l'empreinte, une représentation simplifiée du bâtiment est alors générée par l'extrusion de la forme 2D et ajoutée à l'ensemble des représentations du bâtiment (Figure 60a).
- L'utilisateur doit ensuite choisir le système de référence à utiliser pour calculer les coordonnées (Figure 60b), l'algorithme qu'il convient d'utiliser pour déterminer les coordonnées associées à ce CRS (Figure 60d) et les cartes qu'il va utiliser pour positionner l'empreinte (Figure 60c). Le système de référence géodésique choisi doit être compatible avec la carte de support du processus de géoréférencement. Quant à l'algorithme sélectionné, celui-ci va influencer sur la précision du géoréférencement. À ce niveau du processus, les cartes sont souvent issues de web services (WMS). Cependant, il est possible d'utiliser n'importe quelle source de données (Shapefile, geoTIFF, CityGML...).
- Une fois la carte et l'empreinte de l'objet affichées côte à côte, l'utilisateur va positionner chaque point de l'empreinte sur la carte (Figure 61). Les coordonnées géodésiques sont alors automatiquement calculées et placées dans une matrice de transformation que l'on va ensuite stocker avec l'objet dans la base de données. Cette matrice de transformation (matrice 4x4) permettra, lors du processus de visualisation, de positionner facilement et rapidement les objets dans la scène, sans autre traitement si le système de coordonnées choisi pour l'affichage est le même que celui utilisé dans cette étape.

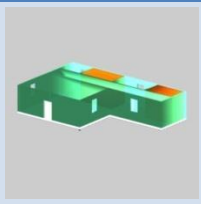
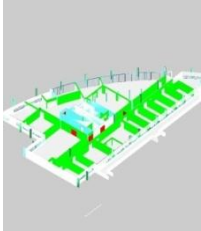
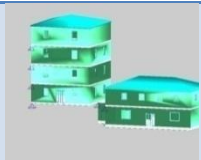
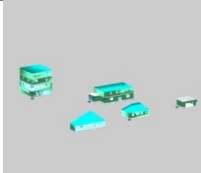
Building(s)	Size (Kb)	Number of buildings	Number of classes	Time (sec) Image 2500x2500px	Time (sec) Image 3500x3500px	Time (sec) Image 4000x4000px
	624	1	177	1.21	3.6	5.06
	1586	1	1312	1.7	4.01	5.3
	675	2	322	1.5	3.7	5.17
	675	5	725	1.82	4.2	5.45

Tableau 5. Performance du calcul d’empreinte 2D en fonction de la résolution de l’image sur laquelle est effectué le calcul

L’avantage de passer par une empreinte en deux dimensions de l’objet à géoréférencer est la rapidité et la précision des calculs des coordonnées en plus de la possibilité d’obtenir un niveau de détail intermédiaire entre la représentation détaillée de l’objet et sa boîte englobante. Pour information le Tableau 5 montre les performances du calcul de l’empreinte au sol des objets en fonction du nombre d’objets dans une scène et de la résolution d’image choisie (plus la résolution de l’image sur laquelle est effectué l’algorithme de détection des contours est grande, plus l’empreinte sera précise).

1.5 Stockage des données

Cette étape dans l’architecture des processus vise à stocker les objets tels qu’ils ont été identifiés et structurés dans les étapes précédentes. Ceux-ci sont stockés sous forme de graphes contextuels XML dans une base de données relationnelle. Le stockage est réalisé selon deux schémas de base de données correspondant à la description sémantique et à la définition géométrique des objets comme préconisé dans la norme ISO 19109 pour la modélisation des entités géographiques.

Les données géométriques sont stockées sous forme de graphe de scène, tel que nous l’avons vu précédemment. Ce type de stockage permet notamment d’afficher les géométries en effectuant peu de transformations. De plus, le graphe de scène étant construit de manière hiérarchique, il est aisé de choisir un nœud dans le graphe pour afficher un extrait de scène, mais aussi de sélectionner uniquement les représentations nécessaires (une représentation est systématiquement définie comme représentation principale par défaut, celle-ci pouvant être associée à plusieurs niveaux de détail classiques, chargés eux aussi uniquement sur demande). Cela permet, entre autres, d’optimiser la bande passante du réseau ainsi que la mémoire de la machine exécutant le client, en particulier dans le cas des configurations mobiles.

Pendant la phase de stockage, deux types de représentations géométriques sont stockées : les niveaux de détail géométriques générés automatiquement, s'il n'en existe pas déjà d'explicitement définis (provenant de CityGML par exemple), et des représentations personnalisées qui peuvent être ajoutées au sein de ce processus. Les représentations associées à un objet géométrique peuvent être soit des niveaux de détail, soit des représentations différentes d'un même objet.

Ces représentations sont sélectionnées lors de la phase de streaming selon les critères définis dans l'ontologie (notamment la partie qui définit le contexte comme nous l'avons vu plus haut).

Les données géométriques peuvent être exportées indépendamment de la structure complexe de l'ontologie et des vues personnalisées. En effet, leur stockage sous forme de graphe de scène permet à des applications clientes compatibles de les charger et les afficher en streaming (nous détaillons cette notion dans la section suivante). Cela permet notamment à Active3D de proposer une application indépendante dédiée à l'affichage de plans et maquettes numériques 3D provenant soit directement de la base de données, en chargeant des extraits du graphe de scène, soit directement depuis des fichiers IFC, DWG, CityGML. Cependant, toute la base de données géométrique n'est pas accessible pour un tel client, puisque la notion de droit, définie dans l'ontologie, est prise en compte, et seuls les fichiers « libre de droits » au sens de l'application sont accessibles. De plus, dans le cas de l'exploitation des données géométriques par un client qui ne gère pas l'information sémantique de notre ontologie, les niveaux de détail classiques sont gérés, mais il est possible de sélectionner les représentations à afficher de façon manuelle. Aujourd'hui, ces possibilités sont possibles uniquement dans le client Active3D par l'intermédiaire d'un mode dit « nomade ». Ce processus métier permet à un client de charger une partie de la base de données, sémantique, géométrique ou les deux, afin de pouvoir se rendre sur le terrain sans connexion à l'ontologie. Pour se faire, l'utilisateur sélectionne les informations auxquelles il souhaite avoir accès et celles-ci sont téléchargées dans une base de données locale sur son poste.

Le processus de stockage de l'information permet un stockage persistant des données géométriques, mais aussi de l'information sémantique. Comme évoqué en introduction de ce paragraphe, l'information est stockée sous forme de graphe contextuel XML dans une base de données relationnelle Oracle.

Le choix de la technologie utilisée pour le stockage découle directement des contraintes industrielles liées au projet. Il a fallu en particulier s'adapter aux solutions existantes comme la base de données relationnelle utilisée par Active3D. Si l'on regarde l'aspect géographique du projet, le premier réflexe est de regarder les modules géographiques spécifiques existants sur le marché tel que le module spatial d'Oracle ou postGIS. Mais ces solutions imposant un surcoût important et du temps d'adaptation et de développement supplémentaire, elles n'ont pas été retenues pour l'élaboration de la solution SIGA3D dans un premier temps. Nous avons donc fait le choix, pour l'implémentation et la mise en production, d'utiliser et d'adapter les éléments de SIGA3D décrits dans ce mémoire aux technologies dont nous disposons (en particulier la base de données Oracle pour l'exemple cité ci-dessus).

1.6 Les processus de streaming et de visualisation

Les deux dernières étapes de notre architecture de processus sont regroupées dans cette dernière partie. Le but est d'afficher de manière dynamique l'information stockée et transformée dans les étapes précédentes.

La visualisation est possible de deux manières : en 3D ou de manière textuelle, sous forme d'arbres, pour l'information sémantique. L'information affichée et sa forme dépendent de chaque utilisateur, et plus particulièrement de leur profil. De plus, la façon dont l'information est stockée nous permet de la charger de manière dynamique, à la demande.

Ce choix du streaming est guidé par les processus étudiés autour des domaines du BIM et des SIG. En particulier, la centralisation occupe une place centrale dans ceux-ci. Le BIM vise à centraliser l'information sémantique afin de faciliter le partage de la base de connaissance entre tous les acteurs. Les SIG, pour des raisons plus économiques cette fois, notamment en terme d'espace des données, utilisent depuis bien longtemps une approche centralisée et communautaire avec des grappes de serveurs contenant l'information

géographique. Notre système d'information se veut centralisé, autour d'un serveur commun à tous les acteurs d'un projet, et souple de manière à pouvoir toucher le plus grand nombre. Cette souplesse passe notamment par la prise en compte des limitations de chacun, et notamment sur l'aspect des structures physique. À titre d'exemple, certaines administrations parmi nos clients utilisent des ordinateurs portables avec un processeur simple cœur à 2Ghz, 512 Go de RAM et un chipset graphique (pas de carte graphique dédiée), le tout relié à un réseau bas débit (128Kb/s) ou en utilisation 3G. Le mécanisme de streaming se veut donc un moyen d'alléger les structures de réseau (on télécharge les données à la demande au lieu de télécharger une base de connaissance complète contenant des informations qui ne sont pas forcément pertinentes), l'utilisation des processeurs graphiques (les données étant stockées sous forme de graphe de scène prêt à être affiché, sans calcul lourd) et permet de construire un environnement à la demande, en fonction du profil et du contexte de l'utilisateur. Ce mécanisme est associé à un mode dit « nomade » dans lequel il est possible de précharger en local dans un cache un extrait de la base de connaissance afin de pouvoir l'utiliser ultérieurement en mode déconnecté. Il existe cependant des limitations à ce mode comme l'impossibilité de modifier certaines données de la base patrimoniale (procédures de fusion des modifications, de gestion de la concurrence et de vérification de la cohérence des données complexes) ou d'accéder à des données qui n'auraient pas été stockées.

Le processus de mise en cache des données est également d'une grande utilité en mode connecté. Il permet d'accélérer le transfert des données déjà visualisées et de précharger des données susceptibles d'être visualisées (niveau de détail plus précis par exemple). C'est également lors de la mise en cache des données, avant leur visualisation, que s'effectue le processus de transformation des données. Le cache, qui contient l'ensemble des données du contexte courant afin de charger les données, géométrique ou sémantique de façon à les afficher sans traitement supplémentaire (réorganisation de la vue par exemple), contient par conséquent les informations sur le système de référence géodésique utilisé pour la visualisation des données. En pratique, ces opérations de transformations de coordonnées sont peu utilisées puisqu'un gestionnaire technique de patrimoine possède souvent ses plans décrits dans un même système de coordonnées.

Revenons maintenant à la présentation des données. L'utilisateur peut choisir de visualiser son patrimoine en trois dimensions lorsque cela lui est permis. Cela requiert d'effectuer des requêtes graphiques dans la base de données pour récupérer les géométries. Quand de telles requêtes sont effectuées, le processus de streaming consiste à récupérer la partie du graphe de scène à laquelle l'utilisateur souhaite accéder, par exemple un site complet avec tous les bâtiments et objets urbains, ou alors juste un étage de bâtiment ou un ensemble d'objets de proximité comme l'ensemble des bouches à incendie d'une zone donnée. Pendant ce processus, deux opérations supplémentaires sont effectuées qui dépendent du contexte d'utilisation. Il faut en effet, pour chaque objet chargé, choisir sa représentation. Cette opération est applicable à la fois pour les niveaux de détail géométriques classiques, mais aussi pour les niveaux de détail contextuels. La seconde opération est la transformation des coordonnées du système de référence géodésique d'origine dans celui utilisé par l'utilisateur dans son contexte d'utilisation.

La visualisation de l'ontologie peut également être faite sous forme d'arbres alphanumériques contextuels. Pour ce faire, le processus consiste à charger depuis la base de données des extraits de l'information contenue dans le graphe sémantique créé durant le processus de stockage des données. Les données sont ensuite affichées dans une vue contextuelle créée durant le processus de modélisation, et choisies dépendamment du contexte courant de l'utilisateur. De la même façon que pour la partie 3D, seules les données visualisées sont chargées en mémoire ; ce mécanisme de streaming est possible grâce à l'utilisation d'arbres alphanumériques pour l'affichage où seuls les nœuds affichés sont chargés en mémoire, sans leur sous-arborescence).

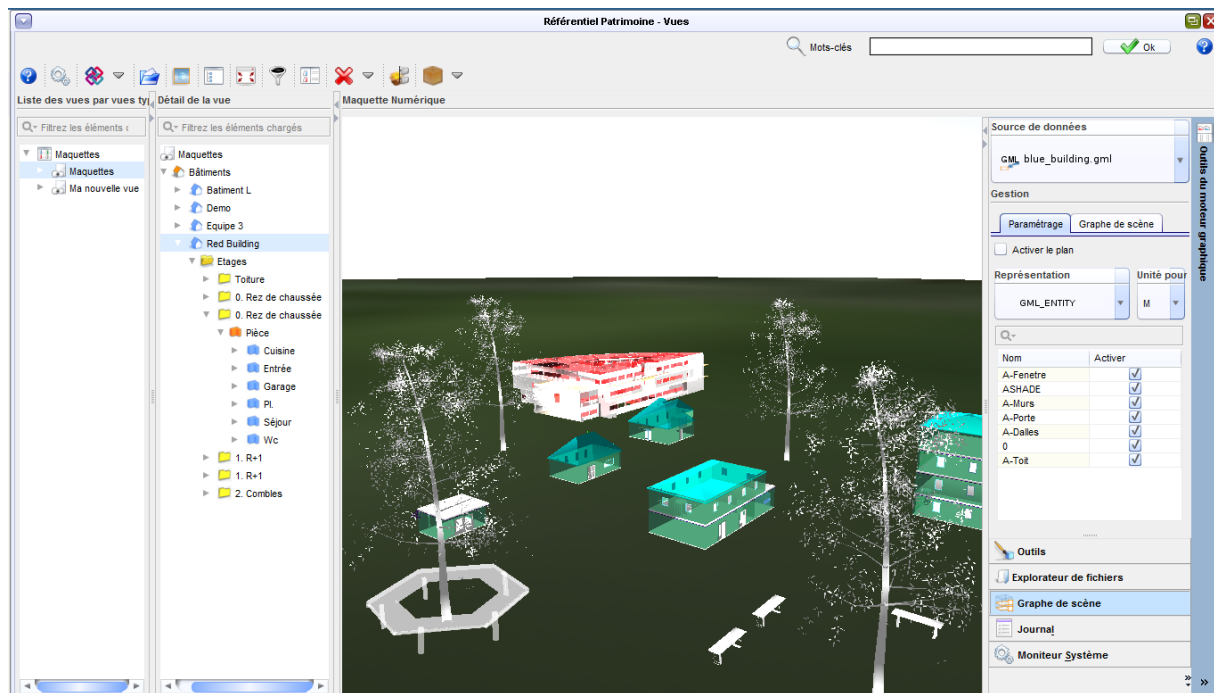


Figure 62. Snapshot of the SIGA3D platform

La Figure 62 est une capture d'écran de la plateforme Active3D dans laquelle on peut voir un site affiché à la fois dans le moteur 3D et dans la vue alphanumérique. Ces deux vues sont d'ailleurs liées et les objets sélectionnés et modifiés dans une vue le sont également dans l'autre grâce au mécanisme d'identifiant unique que nous avons expliqué dans la phase d'import des données. Il est alors possible d'effectuer une requête géométrique depuis la vue sémantique pour avoir une vision 3D et personnalisée de l'objet sélectionné, ou bien au contraire de faire une requête sémantique dans la vue 3D afin d'obtenir des informations supplémentaires sur l'objet sélectionné.

2 Le projet européen SIGA3D

Les travaux de recherche que nous avons menés s'inscrivent dans le cadre d'un projet européen qui vise à intégrer les techniques et technologies du domaine des systèmes d'information géographique dans la plateforme de gestion technique de patrimoine (GTP) Active3D. Durant mes années de thèse, j'ai eu la charge de diriger et coordonner le projet et ses intervenants. Ce projet vise essentiellement l'industrialisation des travaux de recherche menés pendant la thèse, en se focalisant plus spécifiquement sur les aspects géographiques.

Les deux projets constituant SIGA3D, le projet de recherche et le projet européen, possèdent la même finalité, mais l'approche diffère quelque peu : dans le premier cas nous cherchons à développer la solution qui nous semble la meilleure pour construire une plateforme de gestion technique de patrimoine urbain, dans l'autre cas nous prenons pour base la plateforme de gestion technique de patrimoine immobilier existante Active3D et nous la faisons évoluer afin d'intégrer les résultats de la recherche.

Le projet européen SIGA3D a été fait en collaboration avec un spécialiste du SIG en Allemagne, lat/lon⁷⁷. Cette société s'occupe de faire du développement de logiciels, mais aussi des formations et des ateliers dans le domaine de la géographie. Pour cela, une équipe interdisciplinaire composée de spécialistes de la Terre, de géographes, d'informaticiens, d'ingénieurs et de managers lie le savoir-faire technique à l'expérience de la

⁷⁷ <http://www.lat-lon.de/>

conception. Depuis le début, lat/on est membre de l'Open Geospatial Consortium (OGC). Leur objectif est d'établir une infrastructure cohérente et interopérable pour la gestion et le partage des données et de l'information spatiale. Ils développent des logiciels open source. En particulier, leur projet deegree est l'un des plus actifs concernant l'implémentation des standards de l'OGC. C'est un projet open source, initié en partenariat avec l'Université de Bonn, et porté par l'Open Source Geospatial Foundation. Ce projet se fait également en coopération avec deux laboratoires de recherche : le LE2I de l'Université de Bourgogne et le LSIS de l'Université de Provence. Cette collaboration se fait dans le cadre du projet de recherche transversal CheckSem⁷⁸. Le LE2I est orienté sur la partie gestion des connaissances du modèle d'information et le LSIS est orienté sur la partie gestion des standards de l'information géographique.

Toutes ces compétences regroupées sont mises en œuvre dans ce projet afin d'aboutir à la construction d'un modèle sémantique pour la représentation d'objets dans un SIG 3D. La collaboration porte sur divers aspects particuliers du projet. Parmi ceux-ci nous pouvons trouver la modélisation et la représentation géographique des données. Cela passe en particulier par la gestion de tous les aspects liés au géoréférencement de l'information ainsi qu'à la problématique de la montée en charge de l'application vis-à-vis du nombre d'objets pouvant être traités et affichés simultanément au sein d'une même scène graphique.

Les différentes tâches de ce projet sont regroupées en plusieurs objectifs de recherche : la préparation technique de la plateforme Active3D pour supporter les évolutions du projet (gestion des LoD, du streaming...), la gestion des coordonnées (avec l'expertise de lat/on), la modélisation du terrain (à la fois gestion des cartes et des MNT), et la gestion des objets externes au bâtiment.

2.1 Préparation technique d'Active3D

La première phase du projet a été de consolider les bases existantes de la plateforme Active3D afin de pouvoir accueillir les évolutions liées au traitement particulier de l'information géographique. Si l'on se base sur notre architecture de processus, cette partie touche les processus de visualisation, de streaming et de stockage des données. En particulier, il a fallu développer une nouvelle version du moteur Active3D. Pour ce faire, les résultats de l'état de l'art menés et présentés en début de mémoire ont été utilisés, et l'architecture présentée a été adaptée afin d'en intégrer les éléments clés dans l'environnement de développement industriel. Cette étape a vu la mise en place d'un moteur graphique gérant les mécanismes d'optimisation de scènes vus précédemment afin de supporter à la fois la montée en charge de l'application, mais aussi d'anticiper sur le mécanisme de multireprésentation contextuelle (C-LoD) qui a été détaillé dans l'approche de la solution. Des outils de manipulation des scènes ont également été développés afin que l'utilisateur puisse interagir avec celle-ci et effectuer à la fois des requêtes graphiques mais aussi manipuler la scène et consulter les informations sémantiques liées aux objets sélectionnés. C'est ainsi qu'il est possible de visionner en temps réel l'ensemble du graphe de scène chargé. Le deuxième gros travail du développement fut la mise en place du streaming et de la base de données permettant le chargement dynamique des objets. Il a fallu notamment choisir le type de base de données à utiliser (base de données géographique ou non) et déterminer le meilleur moyen d'intégrer, dans une architecture existante et complexe déjà utilisée par de nombreux utilisateurs, de nouveaux mécanismes de modélisation de l'information. La mise en place du mécanisme de chargement dynamique des objets, géométriques et sémantiques a également nécessité la mise en place d'une procédure de mise en cache de l'information. Ce cache est, comme nous l'avons vu dans l'architecture des processus, formaté différemment selon les utilisateurs et la structure de données qu'ils utilisent. C'est ce même cache qui stocke l'information du mode nomade que nous avons évoqué.

Une autre étape dans la préparation de la plateforme Active3D est la notion de bibliothèque graphique qui donne la possibilité d'ajouter des objets dans la scène directement depuis la base de données. Pour garder une cohérence dans les objets ajoutés et la structure du graphe de scène et du graphe sémantique, il a fallu

⁷⁸ <http://checksem.u-bourgogne.fr/www/>

développer des algorithmes pour déterminer la position des objets insérés et leur représentation dans le graphe de scène de la base de données géométrique. Enfin, pour finaliser l'étape préparatoire, nous avons intégré l'import des formats de fichiers non géographiques (du moins pas spécialisés dans ce domaine) et non objets (DWG, DXF, DGN et PDF) selon les critères que nous avons détaillés dans l'architecture SIGA3D.

2.2 Gestion des coordonnées

La seconde phase de ce projet européen porte principalement sur l'expertise de lat/lon en matière de SIG, et par conséquent le géoréférencement. Les objectifs, pour cette partie, ont été de définir les stratégies de gestion des coordonnées au sein du projet SIGA3D. Dans l'architecture SIGA3D que nous avons décrite, la gestion des coordonnées s'effectue à plusieurs niveaux.

Pour résumer la problématique de la gestion des coordonnées, nous allons prendre l'exemple du territoire français. Le système traditionnel en France jusqu'à la fin du XX^e siècle était la Nouvelle Triangulation de la France (NTF) associée à la projection de Lambert (Lambert I, II, III, IV et Lambert II carto en fonction de l'emplacement des objets à géoréférencer). Aujourd'hui, en métropole, le RGF93, qui a succédé à la NTF, est le système géodésique officiel et doit être utilisé pour toutes les applications précises (pour les applications courantes, le RGF93 peut être confondu avec le WGS 84). Les projections associées à ce système sont les Lambert 93 et Coniques Conformées 9 zones. En pratique, aujourd'hui tous ces systèmes cohabitent : Lambert de I à IV pour les plans anciens, Lambert II étendu et Lambert 93 pour les plans plus récents. Par conséquent, il faut choisir un système de coordonnées de référence pour chaque objet en fonction de sa région et de l'époque à laquelle il a été créé. Cela implique que les objets ne possédant pas les mêmes caractéristiques de géoréférencement ne peuvent être stockés dans une même table de base de données si celle-ci est associée à un unique système de coordonnées de référence. Une des solutions que nous avons évoquées plus tôt serait de transformer les coordonnées des objets que l'on insère dans la base de données afin d'homogénéiser le système de référence utilisé. Les systèmes géodésiques de référence globaux pourraient être une bonne idée, mais ceux-ci comme le système mondial WGS84 ou le système européen comme ETRS89 ne sont pas recommandés pour le stockage des coordonnées pour deux raisons :

- Les unités de mesure de ces systèmes ne sont pas métriques (il faut utiliser un système de projection)
- Ils ne sont pas assez précis au regard des objectifs fixés par le projet, à savoir la gestion technique d'éléments et la visualisation sur de larges scènes.

De même, si l'on se limite pour l'instant aux objets situés en France métropolitaine, la projection Lambert 93 peut sembler être une bonne solution, mais c'est justement l'importance de l'altération linéaire induite par la grande largeur de la zone d'application du Lambert-93 qui a conduit à développer les coniques conformes 9 zones. En pratique cependant, Les CC 9 zones ne sont que très peu utilisées. Une autre des raisons qui nous incite à ne pas modifier les coordonnées des plans est la réutilisabilité des plans : en effet, la majorité des clients de la solution Active3D n'utilisent le moteur graphique que pour visualiser des plans ou pour effectuer des modifications mineures. Modifier les plans et les systèmes de coordonnées qui leur sont associés compliquerait la réutilisation de ceux-ci dans les outils de production originaux tels que logiciels de CAO ou SIG. En raison de toutes ces considérations pour trouver un système de référence commun, nous avons décidé, comme il a été présenté dans l'architecture SIGA3D, de stocker chaque plan et objet avec son propre système de coordonnées et d'effectuer les transformations de coordonnées lors de l'utilisation de ceux-ci. Pour cela nous avons développé avec lat/lon un outil permettant la transformation de coordonnées, traitant en particulier les systèmes usités en France (il s'agit en fait d'une évolution de l'outil présent dans deegree). La transformation des coordonnées s'effectue sur l'empreinte des objets calculée au format GML. Dans certains cas, et selon les algorithmes utilisés, la transformation des coordonnées modifie l'aspect des géométries. Dans les développements que nous avons effectués, seules la position, l'orientation et l'échelle sur les objets dans

leur ensemble sont modifiées ; les effets de distorsions ne sont pas répercutés. Les algorithmes utilisés sont ceux de la rotation du premier méridien, les transformations de Helmert à trois et sept paramètres, les transformations de coordonnées géographiques vers des coordonnées géodésiques (2D->3D) avec la hauteur ellipsoïdale, et les transformations polynomiales directes (moindres carrés).

L'autre point sur lequel la collaboration franco-allemande sur le projet SIGA3D a porté ses fruits est le mécanisme de géoréférencement des données. En effet, dans le paragraphe ci-dessus nous traitons les objets géographiques comme s'ils possédaient tous, dès leur acquisition, des coordonnées géographiques. Ce n'est cependant pas le cas et de nombreux objets proviennent de projets architecturaux réalisés à l'aide de système de coordonnées d'ingénierie. Cela pose un problème puisque si l'on souhaite afficher deux sites possédant bâtiments et autres objets géographiques tels des arbres au sein d'une même scène, quel que soit le système de coordonnées utilisé, la position relative de ces deux sites ne pourra être exacte (voir la Figure 63). Pour cela, un outil a été développé selon le processus spécifique au projet que nous avons détaillé dans la partie précédente.

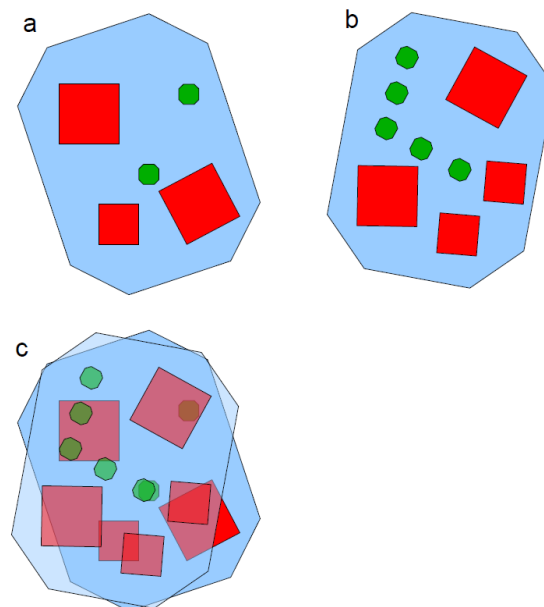


Figure 63. Rendu dans la même scène de deux sites dont les systèmes de coordonnées sont locaux et indépendants.

2.3 Modélisation du terrain et gestion des objets urbains

La modélisation du terrain est la troisième phase du projet. Celle-ci consiste à définir les techniques et formats de données utilisables pour représenter un modèle numérique de terrain dans le moteur et à intégrer le positionnement des objets par rapport au terrain, ce qui peut poser certains problèmes dans un environnement 3D (pour définir l'emprise du bâtiment par exemple). Sur les plans d'architecte en 2D, le plan d'élévation du bâtiment permet de positionner l'œuvre par rapport au sol. Cependant, dans les plans d'objets 3D le sol n'est pratiquement jamais représenté, ce qui constitue un problème pour leur positionnement vertical. La Figure 64 illustre les problèmes rencontrés. On peut voir sur cette figure différentes manières de positionner verticalement un objet. Par défaut, deegree est capable de positionner un objet 3D sur un modèle numérique de terrain, mais sans se soucier de son positionnement exact.



Figure 64. Illustration du problème de positionnement vertical des objets par rapport au terrain

La solution pour l'intégration de cette fonctionnalité dans Active3D a été le développement d'un outil permettant le positionnement vertical manuel des objets lorsque cela est nécessaire. Comme pour le mécanisme de géoréférencement, la matrice de transformation de l'objet repositionné est modifiée et stockée dans l'instance de l'objet en base de données. Une capture d'écran de cet outil et de son fonctionnement peut être trouvée dans la Figure 65. Sur celle-ci nous voyons à gauche un bâtiment et un plan (superposition d'une orthophoto et d'un plan cadastral provenant de l'IGN) qui ne sont pas alignés verticalement (le bâtiment flotte). La partie droite montre le bâtiment correctement repositionné ainsi que l'outil dans le menu qui permet de sélectionner l'objet à repositionner et d'effectuer l'opération. La position horizontale du bâtiment (géoréférencement) peut bien sûr être ajustée si le bâtiment n'est pas correctement positionné sur la carte après son repositionnement vertical.

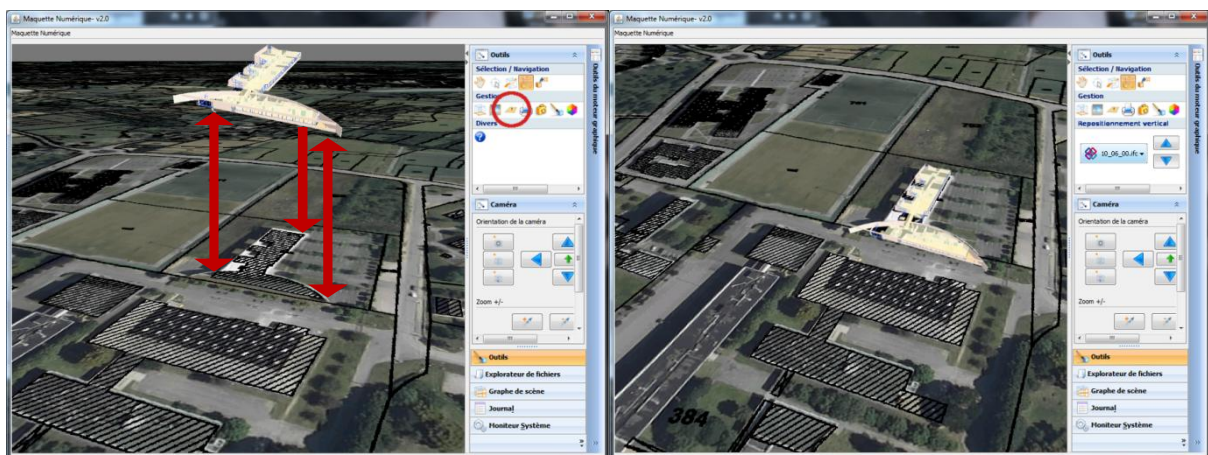


Figure 65. Exemple de repositionnement vertical d'un bâtiment (source du bâtiment : Active3D, source de la carte : IGN (orthophoto + cadastre))

Enfin, la dernière phase du projet est celle qui vise à la gestion des objets urbains. Le projet a abordé plusieurs problématiques qui ont trouvé des solutions dans la proposition d'architecture de SIGA3D présentée dans les chapitres précédents et qui ont permis de la compléter afin d'en faire un ensemble complet et cohérent pour la gestion d'objets urbains. Cette étape peut se décomposer en trois problématiques majeures :

- Le moyen de définir, représenter et stocker les objets qui ne sont pas rattachés au bâtiment, optimiser la gestion de la mémoire, du transfert réseau et de l'affichage pour afficher des centaines ou des milliers d'objets et trouver la manière de gérer la connaissance sur ces objets. On remarque que l'architecture proposée répond parfaitement à ces besoins. L'ontologie permet de stocker, de gérer et de structurer la connaissance à la fois sur les objets du bâtiment, mais aussi sur les objets géographiques. Les niveaux de détails, et en particulier les niveaux de détail sémantiques que nous avons introduits apportent une réponse sur la gestion d'un très grand nombre d'objets, à la fois dans leurs représentations 3D et dans leur représentation sémantique. Enfin, le mécanisme de streaming et

de gestion du cache que nous avons imaginé et implémenté, autorisé par la structure de stockage des données sous forme de graphes, permet d’optimiser les ressources réseau et les temps de calcul côté client. Nous revenons plus en détail sur l’implémentation de ces éléments dans la partie suivante du chapitre.

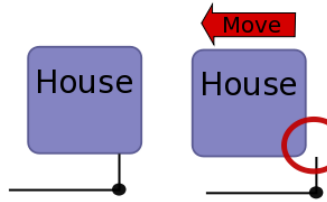


Figure 66. Illustration d'un problème dû au manque de topologie

- La gestion des objets qui apparaissent plusieurs fois dans un modèle de données urbain tel que les arbres, les feux de circulation... a également joué un rôle dans l’élaboration de l’architecture SIGA3D. C’est ainsi que la notion d’objet type permet d’associer une représentation par défaut, graphique ou sémantique, à un type d’objet. Cet objet type peut ensuite être instancié autant de fois que nécessaire pour décrire le modèle auquel il appartient, avec des paramètres qui lui sont propres (par exemple pour une représentation graphique, chaque instances d’objets types possède sa propre matrice de transformation ce qui permet d’avoir une seule représentation de base partagée entre tous les objets, mais chacun d’entre eux pouvant avoir une taille et une position différentes).
- La gestion des relations entre les objets, notamment lors de la modification d’objets au sein d’un modèle. Cette problématique apparaît notamment lors du repositionnement d’objet, tel que le géoréférencement. La Figure 66 illustre le cas typique de problème pouvant survenir suite au déplacement d’une maison par rapport au réseau électrique. Ce point a été complété dans notre architecture SIGA3D par la prise en compte des relations temporelles entre les objets.

Le projet SIGA3D est au final un projet de collaboration entre différents acteurs professionnels et du monde de la recherche : les atouts de chacun de ses domaines permettent de coupler les compétences afin de produire une solution innovante et fonctionnelle en exploitation professionnelle. Ainsi, la recherche fondamentale a su apporter des solutions à des problématiques et contraintes industrielles, et certains axes de réflexion dans le développement à des fins commerciales, ont permis d’enrichir la proposition formulée dans le cadre des travaux de cette thèse.

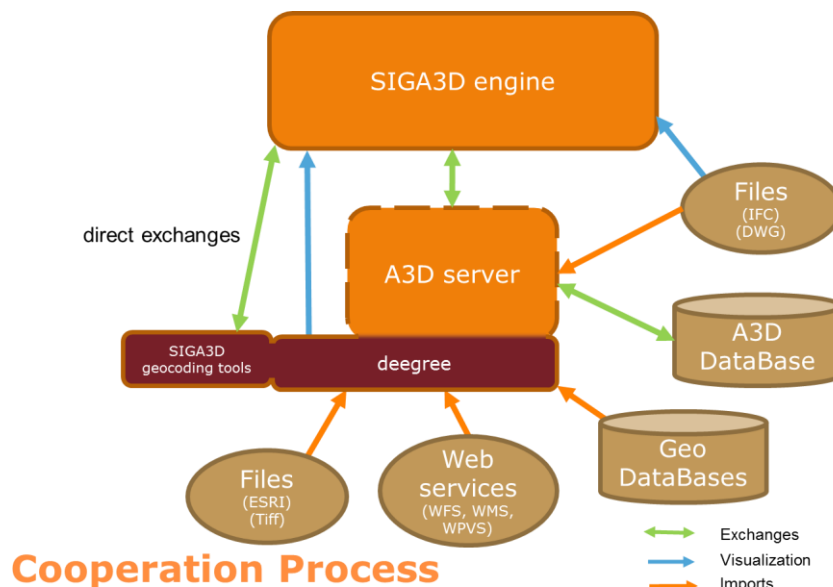


Figure 67. Architecture de collaboration SIGA3D

Pour conclure cette première partie, nous allons présenter l'architecture de collaboration entre les éléments apportés par lat/lon et la plateforme Active3D (Figure 67). Celle-ci présente les différents outils utilisés pour atteindre les objectifs fixés et constitue l'implémentation de l'architecture en 6 couches que nous avons décrite plus tôt. On peut y trouver notamment les différentes sources de données utilisées par SIGA3D, depuis les différents fichiers (IFC, CityGML, DWG...) jusqu'au web services en passant par les bases de données. C'est par ailleurs dans la base de données Active3D (A3D) que nous stockons les différents graphes contenant les données sémantiques et géométriques. La solution deegree qui apparaît ici sert à faire le lien entre les solutions d'acquisition de données géographiques telles que les web services, les bases de données géographiques et les formats de fichiers (ShapeFiles, GeoTiff...) et le serveur d'import de SIGA3D qui transforme et stocke les données de manière propre à chaque modèle d'information urbain. Le SIGA3D engine quant à lui regroupe l'ensemble des différents outils permettant de visualiser les données (moteur 3D et arbre alphanumérique), les mécanismes de streaming et de cache. Nous allons maintenant détailler de manière plus précise l'implémentation qui a été faite de ces outils.

3 Implémentation SIGA3D

Cette dernière partie du mémoire de thèse va présenter les développements réalisés dans le cadre du projet SIGA3D. Cette partie, comme nous l'avons déjà évoqué, est fortement liée aux contraintes industrielles et en particulier à l'existant. Il semble en effet difficile pour l'entreprise de remettre en question des installations chez les clients et utilisées par plusieurs milliers d'utilisateurs. D'autant plus lorsque nos clients possèdent de grosses infrastructures (comme le ministère de la Défense ou la mairie de Paris) et que chaque modification effectuée soit au niveau du serveur, soit du client Active3D, demande de nombreuses spécifications en termes de serveurs physiques et de dépendances logicielles, des protocoles de tests, de validation, d'étude de code, et ce à la fois par le client final, mais également par tous les intermédiaires possibles. Ces contraintes ont donc été décisives dans le choix technique de certains aspects de l'implémentation comme le choix de la base de données par exemple : la plateforme Active3D utilise par défaut une base de données Oracle. Pour le traitement des données géographiques, nous nous sommes naturellement intéressés aux bases de données géographiques. Celles-ci possèdent de nombreux avantages pour le traitement d'objets géoréférencés tels que les index spatiaux et l'optimisation des requêtes spatiales. S'il existe des solutions intéressantes et gratuites comme le module PostGIS du Système de Gestion de Bases de Données (SGBD) PostgreSQL, ces solutions n'ont pas été retenues. En effet, un premier problème est de savoir si l'on migre toutes les données de l'application sur les bases de données compatibles ou si l'on fait se côtoyer plusieurs SGBD pour traiter l'ensemble des données. Ensuite, pour la mise en place de l'architecture, cela demanderait :

- soit d'avoir nos propres serveurs mis à disposition des clients pour le traitement de l'information géographique, avec ce que cela entraîne en terme de confidentialité des données et problèmes de sécurité,
- ou alors à nos clients d'installer une telle base de données, ce qui leur demande des compétences et de la maintenance supplémentaires de leurs serveurs, avec une gestion des différents conflits que cela pourrait créer sur leur architecture d'entreprise existante.

On voit avec ces quelques questions d'approches que le choix n'est pas évident d'un point de vue d'administration des infrastructures dans l'entreprise. Vient alors la question des solutions payantes, mais compatibles avec l'architecture existante, à savoir l'installation du module spatial d'Oracle. Si cette solution a bel et bien été étudiée, elle n'a pas été retenue pour des questions de budget. En effet, ce module n'étant pas gratuit, il faut l'intégrer dans le prix de vente de l'application et des services que nous proposons lors d'appel d'offres. Le marché de la gestion technique de patrimoine étant actuellement très compétitif, et les entreprises n'ayant pas forcément encore cette culture, qu'elle soit appliquée au domaine immobilier ou urbain, les coûts

élevés supplémentaires engendrés par ce module peuvent être décisifs dans le choix de la solution qu’elles adopteront. C’est pour ces quelques considérations que nous avons décidé de développer notre propre solution basée sur les outils que nous utilisons déjà et optimisée pour l’architecture SIGA3D que nous avons décrite précédemment.

Dans cette partie, nous allons en particulier aborder les choix que nous avons faits pour le développement d’un nouveau moteur graphique pour l’application client Active3D, puis nous verrons la manière dont nous stockons les différents graphes de l’application (ontologie et graphe de scène).

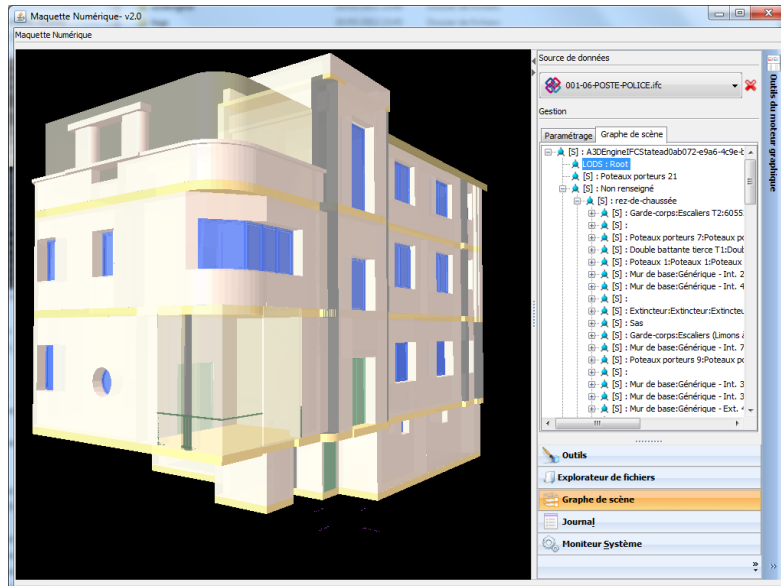


Figure 68. Exemple de bâtiment simple : affichage 3D et graphe de scène. Nombre d’objets dans la scène : 538.

3.1 Le développement d’un nouveau moteur graphique

L’architecture SIGA3D en six couches qui a été présentée dans les chapitres précédents a pour objectif l’élaboration de modèles de données dédiés à la gestion d’environnements urbains. Pour cela, un travail important a été réalisé sur l’optimisation des modèles, à la fois d’un point de vue sémantique, mais aussi d’un point de vue graphique. En effet, cette architecture de modèle d’information urbain et la solution proposée doivent permettre d’afficher de vastes scènes en 3D contenant plusieurs milliers d’objets simultanément. Si le moteur graphique d’Active3D est capable d’afficher des bâtiments au format IFC contenant déjà plusieurs centaines d’objets (la Figure 68 montre un bâtiment sur 4 niveaux décrits avec son architecture intérieure détaillée (murs, escaliers, sanitaires, extincteurs...) qui contient environ 550 objets découpés en 300 000 triangles), il n’est pas adapté pour un usage au sein de l’architecture SIGA3D :

- Le moteur éprouve des difficultés à afficher des bâtiments contenant plus de 4000 objets,
- il n’est possible de charger des plans que depuis des fichiers IFC (chargement complet du fichier en mémoire),
- il n’est pas possible de charger plusieurs bâtiments simultanément
- le géoréférencement des objets n’est pas géré (affichage en coordonnées locales).

Ceci est dû en particulier au manque de mécanismes d’optimisation des scènes 3D comme ceux présentés dans l’état de l’art de ce document. Il a été choisi, pour le développement du moteur de rendu, et afin de limiter les temps et coûts de développements, d’utiliser un moteur de rendu 3D existant et libre afin d’optimiser les temps et coûts de développements et de ne se focaliser que sur le développement des fonctionnalités propres à l’application SIGA3D. Il existe plusieurs moteurs de rendu qui sont développés depuis plusieurs années et qui

possèdent de nombreuses fonctionnalités éprouvées. Ce paragraphe déroule une étude comparative entre deux moteurs afin de comprendre le choix que nous avons fait.

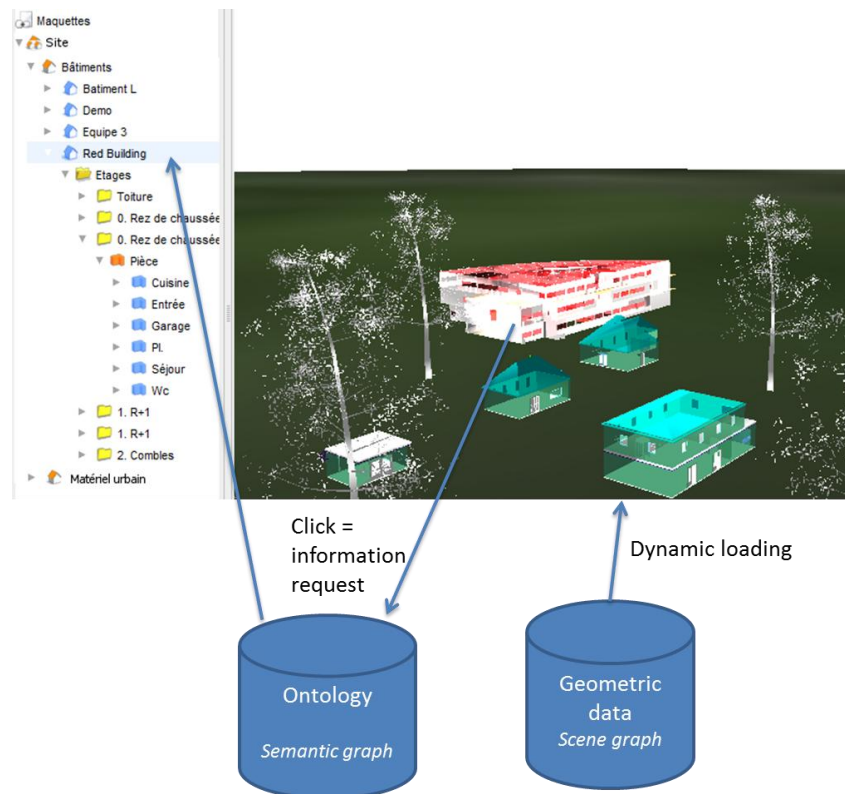


Figure 69. Fonctionnement du lien entre le moteur graphique et la visualisation des informations sémantiques

Pour permettre de gérer les scènes graphiques provenant de l'UIM, des objectifs ont été définis afin de déterminer le moteur de rendu le plus approprié, ainsi que la structure de stockage des informations. Cette dernière inclut notamment le graphe de scène, les représentations multiples et le lien entre l'information graphique et les modèles de données sémantiques définis grâce à l'architecture de l'UIM. Les critères retenus pour la sélection du moteur sont basés sur l'approche de (McCall, 1977) et adaptés aux objectifs fixés. Nous rappelons ici les facteurs de qualités et les critères associés :

- *Maintenabilité* : Le code réutilisé doit, dans le produit final, rester simple, concis et bien documenté.
- *Testabilité* : Le produit final doit pouvoir se prêter, en phase d'exploitation, à une vérification du respect des spécifications.
- *Flexibilité* : Le code du moteur doit être facilement adaptable aux besoins du projet et le résultat doit s'adapter aux besoins des utilisateurs finaux.
- *Portabilité* : Le produit doit pouvoir s'adapter facilement à des environnements variés (client lourd, léger, client mobile...).
- *Réutilisabilité* : Le produit proposé devra pouvoir être réutilisé en dehors de la plateforme Active3D, pour d'autres applications (moteur en standalone pour visualiser des IFC ou autres formats de fichiers, sans base de données par exemple)
- *Interopérabilité* : les développements devront être effectués autour de normes afin de faciliter l'interconnexion avec les autres systèmes existants et l'implémentation de nouvelles normes a posteriori.
- *Exactitude* : Les fonctionnalités du moteur doivent correspondre à ce qu'attend l'utilisateur en terme de visualisation de patrimoine urbain.

- *Fiabilité* : Les transferts de données doivent être fiables et les données résultantes doivent être utilisables pour le but visé.
- *Efficacité* : Le logiciel doit être apte à utiliser d'une façon optimale les ressources physiques (espace mémoire, carte graphique...) mises à disposition.
- *Intégrité* : Les données des utilisateurs doivent être protégées et sécurisées.
- *Ergonomie* : Le moteur doit pouvoir être utilisé de manière intuitive, sans laisser apparaître la complexité des transferts de données.

Ces critères ont logiquement orienté le choix de l'architecture de fonctionnement du moteur Active3D. L'application doit pouvoir être utilisée sur un grand nombre de postes utilisateur aux caractéristiques hétérogènes. Il a également fallu prendre en compte les problématiques liées au chargement des données directement depuis la base de données Active3D. En effet, la visualisation se fait de manière dynamique ce qui permet d'utiliser le moteur de rendu dans des environnements variés : client lourd (comme c'est le cas pour la plateforme Active3D), clients légers (via un navigateur web), ou bien encore d'autres machines ne possédant que très peu d'espace disque (environnements virtualisés, périphérique mobile...). Les données graphiques dans la base sont stockées directement sous forme de graphe de scène, ce qui permet de simplifier les calculs des géométries à afficher côté client et permet l'utilisation d'algorithmes spécifiques (comme les LoD hiérarchiques) afin d'optimiser les ressources (mémoire système, trafic réseau...) en ne chargeant que les parties du graphe de scène que l'on souhaite afficher. La Figure 69 montre un exemple de site affiché dans le moteur graphique, l'arborescence correspondante dans la partie alphanumérique (c'est-à-dire les objets de la scène et leurs liens sémantiques), et les opérations effectuées lors d'un clic pour obtenir le lien entre ces différentes informations.

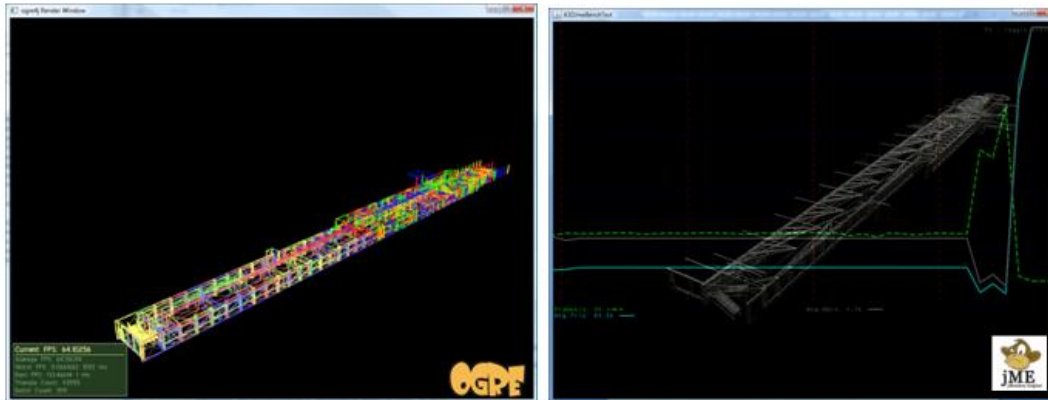
Dans un second temps une étude comparative a été menée afin de sélectionner le moteur de rendu adapté aux besoins de SIGA3D. Tout d'abord, pour une plus grande portabilité de l'application Active3D, celle-ci est développée entièrement en java, avec une mise à jour constante du code afin de satisfaire aux besoins des dernières versions de la machine virtuelle fournie par Oracle. Nous avons fait le choix d'utiliser un moteur de rendu pour les scènes 3D qui soit utilisable avec Java et la norme OpenGL, sur toutes les plateformes. De plus, afin que l'utilisation d'un moteur de rendu graphique développé en dehors d'Active3D soit bénéfique (en temps de développement notamment), nous avons choisi un moteur qui propose de nombreuses optimisations, plus ou moins complexes, telles que la gestion des niveaux de détail discrets ou la gestion des objets hors du champ de la caméra. Le moteur choisi est de fait facilement adaptable à notre architecture et permet notamment d'afficher de manière optimisée les géométries qui sont calculées depuis le parser de données SIGA3D. Ce parser permet de récupérer et transformer les géométries issues de données IFC, DWG, CityGML ou autre.

Le choix s'est opéré entre JMonkeyEngine (JME)⁷⁹ et Ogre3D pour java (appelé Ogre4J)⁸⁰. Ce sont deux moteurs complets au niveau des fonctionnalités qu'ils proposent, qui sont régulièrement mis à jour par leur communauté et qu'ils possèdent une bonne réputation dans le monde du jeu vidéo. La différence principale entre ces deux moteurs vient du fait que le code de JME est natif Java alors que celui d'Ogre4J est un portage de la version C++. Parmi les fonctionnalités communes de ces deux moteurs, nous pourrions noter qu'ils sont tous deux orientés objet, la gestion des scènes est basée sur les graphes de scènes, ils supportent des techniques d'optimisation tels que les LoD et le culling (frustum, back-face culling...), ils supportent l'accélération matérielle des cartes graphiques, ils sont compatibles avec OpenGL 2.0 et offrent un choix entre plusieurs bibliothèques de rendu OpenGL (point qui a son importance puisque l'import IFC dans le moteur original d'Active3D est basé en partie sur certaines fonctionnalités d'une de ces bibliothèques). Cependant, JMonkeyEngine

⁷⁹ <http://jmonkeyengine.com/>

⁸⁰ <http://ogre4j.sourceforge.net/>

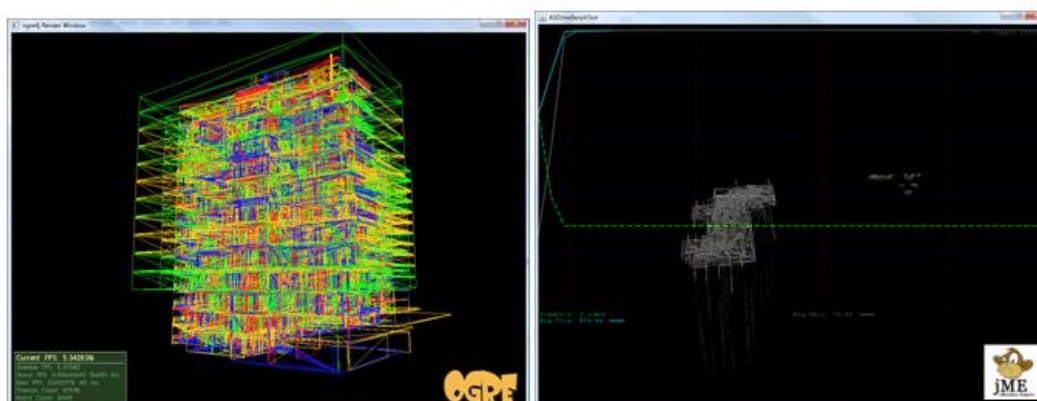
propose une palette de fonctionnalité plus large que ogre3D, notamment dû au fait que les fonctionnalités de JME sont intégrées directement dans le SDK, alors que le mappage des fonctionnalités de Ogre3D vers Ogre4j n'est pas immédiat. Ainsi, JME propose en outre une meilleure gestion des contrôles, des facilités pour la modélisation des terrains et une prise en charge étendue des périphériques.



(A1)

(B1)

- Scène complète => 94K triangles (sans optimisation)
- Ogre4J 64 FPS en moyenne (A1)
- JME 54 FPS en moyenne (B1)



(A2)

(B2)

- Scène complète => 875K triangles
- Ogre4J : 5 FPS (A2)
- JME : 3 FPS (B2)

Figure 70. Comparatif des moteurs de rendu JME et Ogre4J.

Nous avons ensuite testé les performances brutes et les limites de chaque moteur. Pour réduire toute influence sur les performances, seul un fichier IFC est affiché à la fois. La scène comprend une ou deux lumières et le mode WireFrame est utilisé. La mémoire de la machine virtuelle Java est fixée à 1024 Mo. Aucune autre technique d'optimisation n'est utilisée (niveaux de détail par exemple). Le jeu de test se compose de 3 fichiers IFC bien distincts, que l'on peut qualifier de simple (4Mo, 94K triangles), complexe (24Mo, 875K triangles) et très complexe (94Mo). Il s'avère qu'aucun des deux moteurs n'a pu charger le dernier fichier en moins d'une heure avec les paramètres par défaut (sans optimisation spécifique). Les tests pour ce fichier ont été

abandonnés. La Figure 70 montre l’affichage des deux premiers bâtiments dans les moteurs avec leurs performances. Il est apparu que les performances brutes de Ogre4J étaient en général supérieures à celle de JME sur les scènes de taille moyenne (de 10 à 15 images par secondes de plus sur des scènes d’environ 100 000 triangles) et les plus grosses scènes (moins de 5 images par secondes pour les deux moteurs sur des scènes approchant le million de triangles). Cependant, les fonctionnalités et l’intégration du moteur dans la plateforme Active3D sont bien meilleures pour JME (le fait que Ogre4j utilise un wrapper JNI⁸¹ nécessite un travail supplémentaire de gestion de la mémoire, par exemple le fait de devoir supprimer manuellement tous les objets natifs) et les différences de performance sont suffisamment faibles pour que notre choix se soit orienté sur JME.

Une des fonctionnalités intéressantes de JME pour SIGA3D est la prise en compte des ‘states’ qui permettent de préparer une scène pendant qu’une autre est en préparation. Cela nous permet entre autres de pouvoir précharger des représentations différentes pour les objets affichés (niveau de détail y compris) en fonction de la mémoire disponible sur le système et optimiser les temps de chargements entre chaque changement de représentations. Le mécanisme de coloration des objets dans la scène (sélections, requêtes graphiques) passe également par cette fonctionnalité.

Pour la gestion du cache, c’est une autre solution existante qui est utilisée : EhCache⁸². Elle offre une très grande flexibilité dans son utilisation et des performances adaptées à l’usage que nous en faisons. Le cache est utilisé dans le client lourd Active3D pour stocker sur le disque des extraits de la base de données géométriques qui ont déjà été consultés. Cela permet à la fois un accès plus rapide à ces données, mais autorise également de précharger des données lorsqu’une utilisation en mode nomade de l’application est prévue.

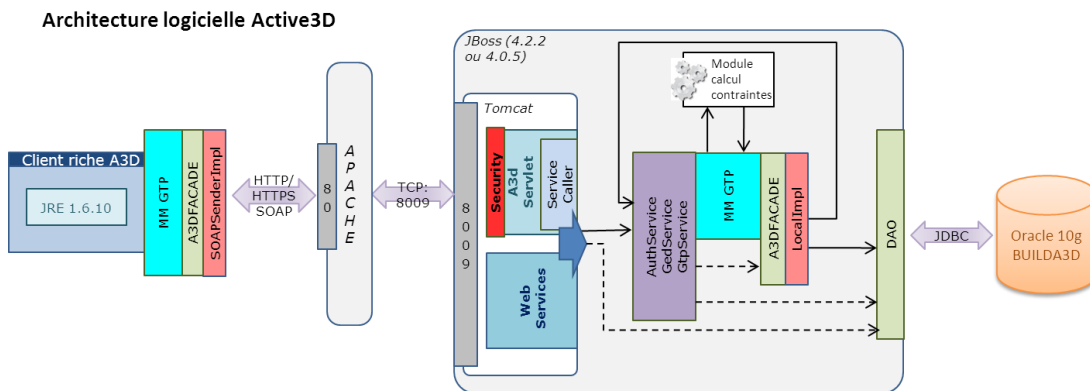


Figure 71. Architecture logicielle 3-tier Active3D

3.2 Stockage du graphe de scène

Le moteur graphique représente pour les utilisateurs une fenêtre sur le modèle d’information urbain construit et stocké en bases de données. Le paragraphe précédent pose les bases de l’architecture qui nous permet de visualiser un patrimoine urbain. Le choix du moteur étant arrêté, nous présentons maintenant l’architecture de stockage des géométries qui nous permet d’effectuer un chargement dynamique des scènes depuis une base de données.

Le mécanisme de streaming des données depuis une base de données distante est un véritable défi puisqu’il nécessite une optimisation des ressources sur plusieurs niveaux de l’architecture SIGA3D, depuis la base de données jusqu’au client de visualisation, en passant par la gestion du trafic sur le réseau. La partie précédente présente quelques mécanismes d’optimisation côté client (LoD, culling). Nous allons maintenant voir les choix

⁸¹ Java Native interface, permet d’interfacer du code natif comme le C++ en code Java

⁸² <http://ehcache.org/>

techniques qui ont été effectués pour la mise en place de la partie serveur, et plus particulièrement la partie stockage des données. L'implémentation décrite ici repose sur l'architecture existante d'Active3D que nous pouvons voir sur la Figure 71. C'est une architecture 3-tier classique avec un client lourd, un serveur d'application et une base de données. Cette architecture gère tous les modules Active3D répartis en 3 services : AuthService pour la gestion des utilisateurs et des autorisations, GedService pour la gestion des documents (échanges, partage, versionning... pour le travail collaboratif), et GtpService pour la gestion technique de patrimoine à proprement parlé. C'est à ce dernier niveau qu'interviennent les développements pour SIGA3D. Les échanges entre le client et le serveur se font à l'aide du protocole SOAP et les données sont encapsulées en http, ce qui permet une utilisation facile à travers internet (le port 80 est celui utilisé par les navigateurs internet pour la navigation standard).

Les données géométriques sont stockées en base de données sous forme de graphe de scène. La particularité du graphe de scène dans Active3D est que sa structure est construite de manière dynamique en fonction des informations décrites dans l'ontologie. Ce point est abordé plus loin. Le mécanisme des graphes de scènes est aujourd'hui largement utilisé dans les applications faisant de la modélisation 3D ou même 2D (jeux vidéo, visualisation scientifique, CAO...) du fait de sa compatibilité avec les principes de la programmation objet et une bonne structuration de la scène. On utilise naturellement le graphe de scène pour décrire des relations spatiales entre les objets. Le principe est d'utiliser un graphe acyclique orienté, afin d'organiser hiérarchiquement une scène 3D. Le stockage des objets et de leurs géométries sous forme de graphe de scène permet un chargement direct des données (sans transformations des données) au sein du moteur graphique de l'application (n'importe quel nœud du graphe de scène peut être la racine d'un sous-graphe de scène). Cela permet de ne télécharger depuis la base de données que les objets que l'on souhaite effectivement visualiser. Le graphe de scène construit et stocké dans Active3D, offre en particulier la possibilité de stocker plusieurs représentations pour chaque objet du graphe de scène. Si le mécanisme est courant afin de gérer les niveaux de détail géométriques, l'approche dans SIGA3D se montre plus originale et va servir à implémenter les niveaux de détail contextuels. Ceux-ci nécessitent un lien avec le modèle d'information sémantique. Ce paragraphe explique comment le lien est créé entre les objets du graphe de scène et ceux de l'ontologie, puis les mécanismes mis en place pour effectuer le streaming des géométries.

La Figure 72 présente un extrait de la base de données relationnelle qui permet de réaliser toutes ses fonctionnalités. Ce modèle conceptuel peut être décomposé en quatre parties fonctionnelles.

La première correspond à la mise en place de la structure du graphe de scène. Il s'agit notamment de décrire les relations père-fils entre ses nœuds. La Figure 73 ci-dessous nous montre l'extrait du modèle conceptuel de données qui réalise cette opération. Celui-ci est composé des classes ENG_OBJ qui représentent l'objet dont on décrit sa géométrie et de la relation REL_OBJ_CHLD qui permet de décrire le sous-graphe de scène.

La seconde partie est celle qui décrit le stockage des représentations pour chaque objet. Un des objectifs principaux est de pouvoir associer plusieurs représentations aux objets définis dans la première partie afin d'implémenter de manière graphique le principe de multireprésentation. Comme nous l'avons expliqué plus tôt, ce mécanisme réalise celui des niveaux de détail contextuels et permet à un objet d'adapter sa représentation au contexte de l'utilisateur. La structure mise en place pour cette partie permet, quelle que soit la représentation, de calculer et stocker des niveaux de détail géométriques classiques.

La Figure 74 nous montre les classes qui décrivent le mécanisme complet des représentations. Elle est composée de la relation REL_OBJ_REPRS et de la classe ENG_REPRS qui contient les représentations de chaque objet. Chaque objet est lié à un ou plusieurs représentations.

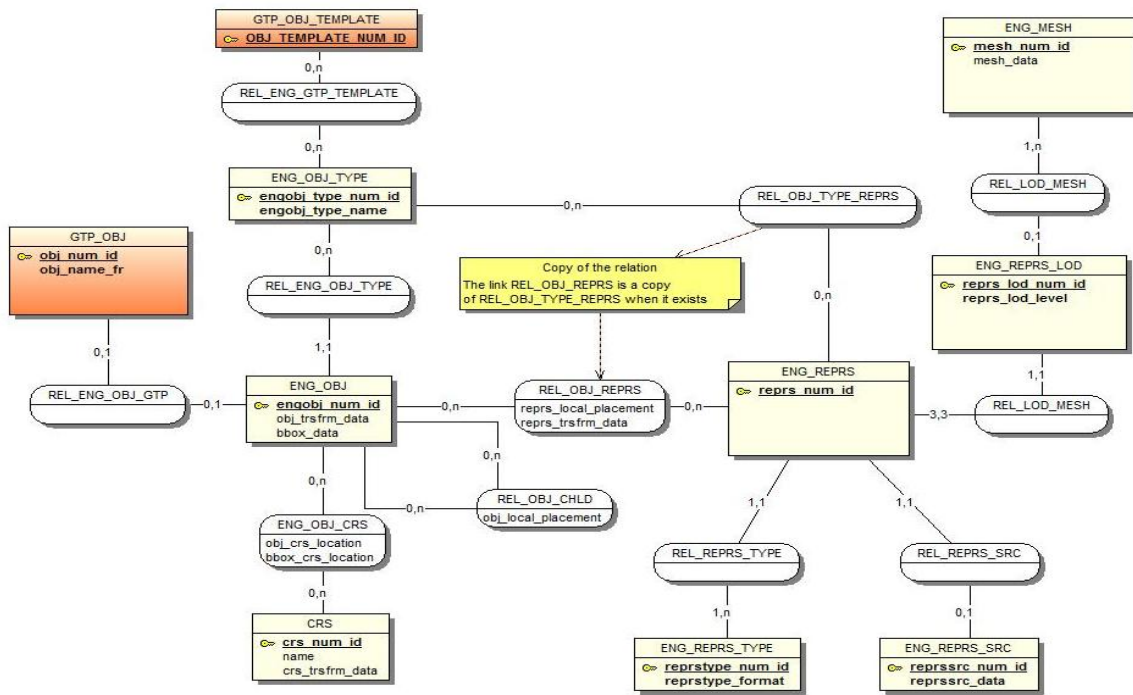


Figure 72. Extrait du modèle conceptuel de la base de données relationnelle stockant le graphe de scène

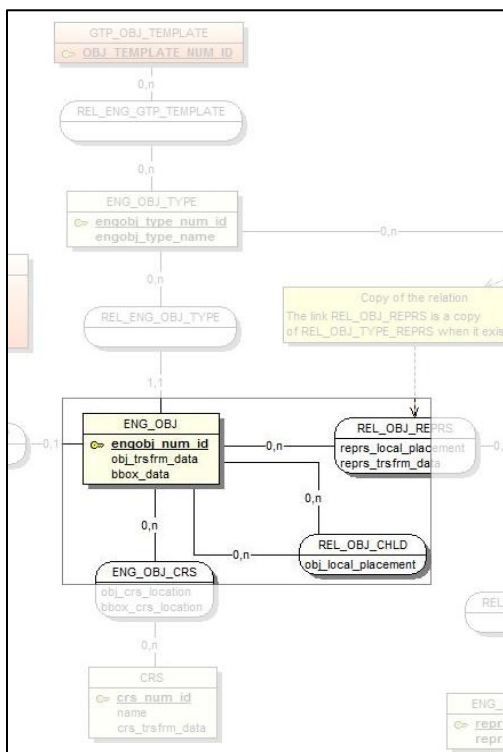


Figure 73. Définition du graphe de scène

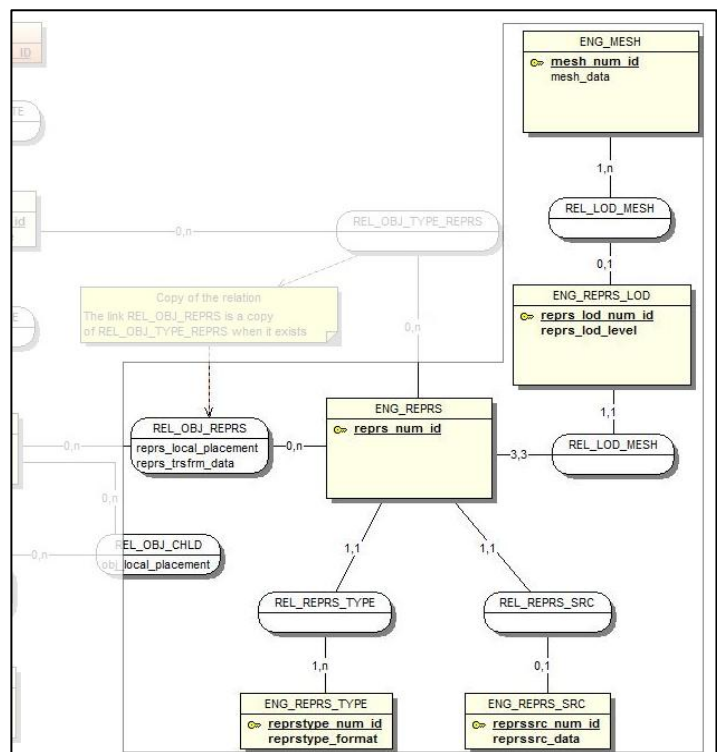
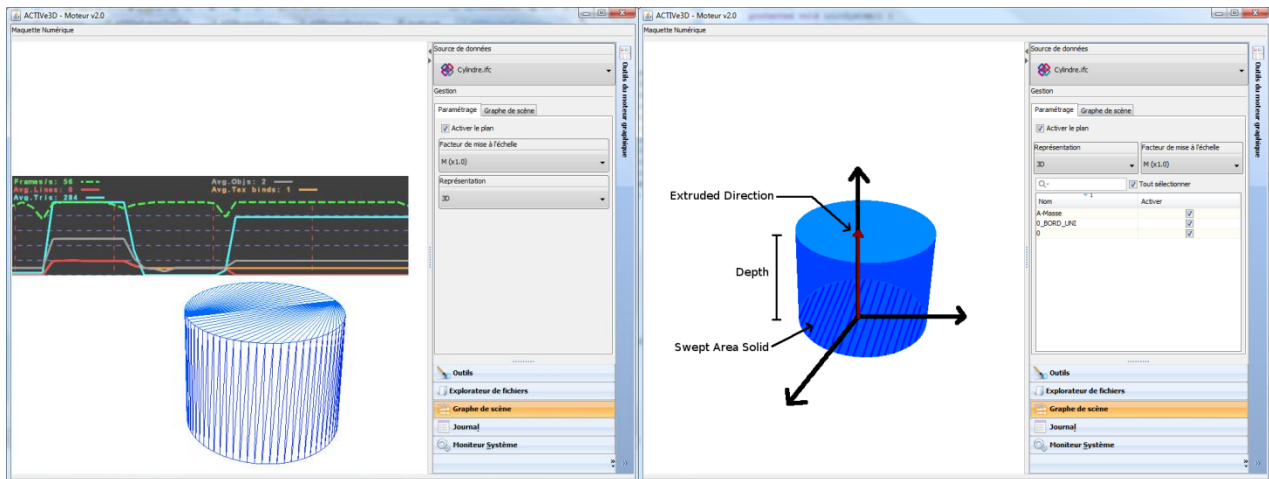


Figure 74. Représentations

Le stockage des géométries dans cette partie va influencer directement les performances du chargement dynamique. Les géométries peuvent être stockées sous deux formes : soit directement sous forme de maillages (description explicite des géométries), soit sous forme de primitives, telles qu'elles sont décrites dans les fichiers sources. Ce dernier cas permet en général d'avoir une bonne factorisation des géométries. Par

exemple, en IFC un mur peut être décrit par une extrusion verticale d’une empreinte polygonale du corps du mur (la Figure 75 illustre ce principe sur un cylindre). La taille des données transférées est alors bien moins importante que la description explicite des géométries, ce qui fait gagner du temps lors du transfert des données. Cependant, pour afficher les géométries, le moteur de rendu doit recalculer toutes les géométries, y compris chacun des LoD quand cela est possible, ce qui consomme de la mémoire et du temps GPU côté client.



Le maillage formant ce cylindre est composé de 284 triangles qui doivent être décrits explicitement sous forme de mesh.

Dans un fichier IFC, le cylindre est décrit avec une surface 2D et une extrusion guidée par une direction et une hauteur.

Figure 75. Comparatif d'une représentation explicite par Trimesh et d'une représentation par primitive

Plusieurs benchmarks ont été faits et sont présentés dans le Tableau 6 ci-dessous.

Type de fichier	Nom du fichier de référence	Poids du fichier (Ko)	Taille du cache (Ko)	Objets 3D	Nb objets	FPS	Tps d'ouverture (s)
DGN	080105002Q_0013_N_00_ATL	151		/		-	7
DGN	080105002Q_0006_N_00_ATL	43		/		-	6
DWG	IVR020-01-A-SURF	138		NON	268	-	11
DWG	CNB250-00-A-SURF	201		NON	170	-	9
DWG	plan pk	566		NON	2564	-	8
DWG	plan masse	1760		NON	32131	-	9
DWG	ba999-v2004	7693		NON	527	-	37
DWG	Maison_test	625		OUI	25	-	4
DWG	Site_Full	676	4403	OUI	131	-	7
DWG	10_11_01	1408	13170	OUI	2015	-	12
DWG	Nantes_long_10_10_00	1688	92211	OUI	569	-	26
DWG	1er_Etage_mairie_de_paris_2010_purge	3600		OUI	85	-	16
DWG	1er_Etage_mairie_de_paris_2010	3767		OUI	236	-	41
IFC	10_10_00.ifc	6100	8606	OUI	907	40	40
IFC	Universite_Nantes.ifc	3872	6484	OUI	1217	45	19
IFC	POSTE-POLICE.ifc (Figure 76)	7060	2872	OUI	538	60	41
IFC	toison_extension.ifc	23823	76757	OUI	5743	18	379

Tableau 6. Performances du moteur par rapport au poids des fichiers et taille des géométries

Les temps d'ouverture des fichiers qui apparaissent dans le tableau comprennent le parsing du fichier, sa mise en mémoire vive (selon un modèle mémoire générique utilisé pour les tests décrivant les relations de contenance naturelles) et en mémoire cache (écriture sur le disque dur), la création des différentes représentations (LoD classiques ou représentation 2D), l'affichage dans le moteur et le stockage dans la base de données géométrique. Si le fichier est déjà en cache, l'ouverture est très souvent presque immédiate et dans tous les cas ne dépasse pas quelques secondes.

On voit également apparaître la taille des fichiers sources, dans différents formats, et la taille des géométries dans le cache de l'application (stockées sous forme de trimesh). Les fichiers sources contiennent à la fois des géométries décrites par primitives et par mesh, ainsi que certaines informations sémantiques (nom d'objets ou autres relations pour les IFC par exemple). La différence entre la source et le cache peut varier d'un facteur 1 (lorsque les géométries sont décrites quasi exclusivement sous forme de mesh dans le fichier, ou que la quantité d'information sémantique dans le fichier est importante) à un facteur 50 dans le pire des cas (bâtiment au format DWG). Dans la majorité des cas, le facteur de différence entre les tailles originales d'un fichier et la description uniquement par mesh des géométries avec tous les niveaux de détail tels que traités dans le moteur varie entre 3 et 5. La Figure 76 est un exemple de fichier IFC pendant la phase de test.

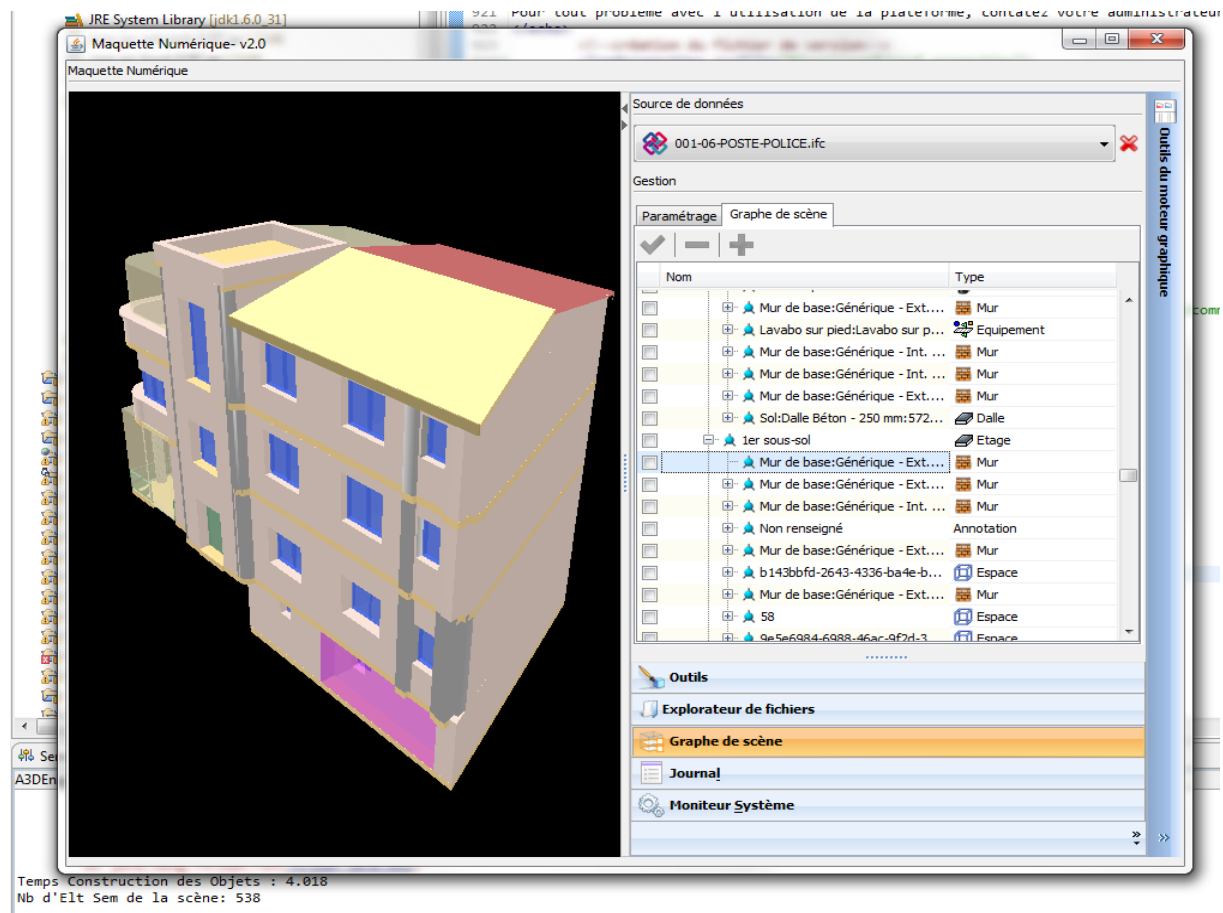


Figure 76. Exemple de fichier IFC et son graphe de scène

Afin de ne pas se limiter à une solution en particulier, nous avons décidé d'intégrer ces deux modes de stockage des géométries dans la base de données, le choix de la stratégie étant fait au niveau du client de visualisation. En pratique, pour limiter les opérations de calcul graphiques côté client, il a été décidé, dans la plateforme Active3D, d'envoyer directement les maillages au moteur graphique afin que celui-ci n'ait plus qu'à les afficher. Ce choix est dicté par le profil de nos clients qui, étant très souvent au sein d'administrations, possède des machines peu puissantes, mais dont les serveurs de données sont connectés aux réseaux locaux

(bonne bande passante). Enfin, chaque représentation est associée à trois niveaux de détail, ceux-ci étant sélectionnés par le moteur, à la demande pour limiter l'occupation mémoire on ne charge qu'un niveau de détail en même temps (le nombre de niveaux de détail est arbitraire, basé sur notre expérience professionnelle et le nombre de LoD qu'il est possible de créer facilement).

L'affichage des niveaux de détail contextuels (c'est-à-dire une représentation choisie selon des critères sémantiques et non plus géométriques) est alors rendu possible par ce schéma de base de données. Il est en effet possible d'associer n'importe quelle représentation à un objet, le choix de la représentation étant opéré par le client en fonction du contexte de l'utilisateur. Un extrait du MCD plus large et spécifique à la gestion technique de patrimoine montre plus loin dans le document (Figure 80) la centralisation des classes et fonctionnalités autour de l'utilisateur.

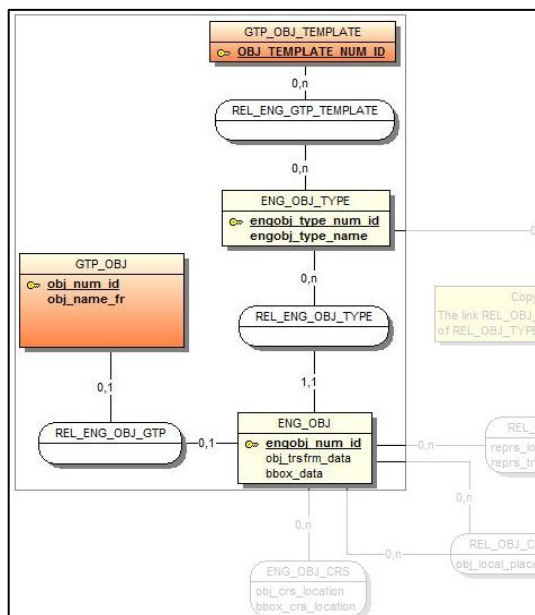


Figure 77. Lien entre le modèle graphique et le modèle sémantique

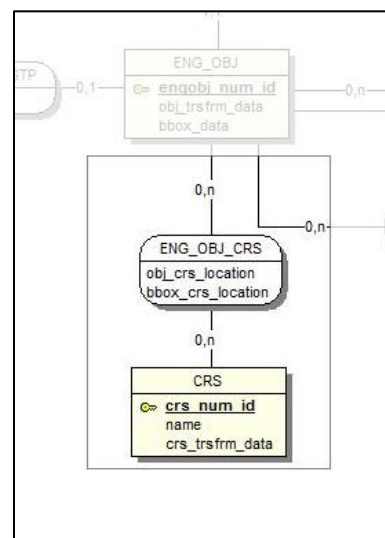


Figure 78. Le géoréférencement

Le lien entre les deux modèles est décrit dans la troisième partie (Figure 77). On y trouve des relations entre les modèles existants : celui de la base de données graphique et celui de l'ontologie SIGA3D. Les classes GTP_OBJ_TEMPLATE et GTP_OBJ font référence respectivement à des modèles d'objets et des objets du modèle de données dans la GTP (c'est-à-dire le niveau conceptuel et le niveau factuel de l'ontologie), et leur homologue dans la base graphique.

Le but de ce lien est de pouvoir retrouver, à partir de chaque objet de l'ontologie, l'objet géométrique qui lui est associé et à partir duquel on peut retrouver sa ou ses représentations graphiques (comme dans la Figure 69). L'inverse est également vrai puisqu'il nous est possible de visualiser une scène depuis la base de données géographique et d'interroger le modèle sémantique des données pour obtenir des informations sur les objets visualisés graphiquement. Le lien se fait à deux niveaux : au niveau du modèle de données et au niveau des instances de données. Dans la partie sémantique du modèle, les objets types servent à créer la structure du modèle de données en spécifiant notamment les relations entre types d'objets. Les objets décrits dans cette partie sont des instances d'objets types. Dans la partie graphique, nous définissons des templates d'objets auxquels on peut associer une représentation par défaut au travers de la classe ENG_OBJ_TYPE. Les objets de cette partie graphique représentent des nœuds du graphe de scène stockés et peuvent être des instances des templates définies plus tôt, auquel cas la représentation associée à l'objet type leur est automatiquement associée par défaut. Ce mécanisme est utilisé en particulier pour créer une bibliothèque d'objets graphique.

Celle-ci permet aux utilisateurs d'ajouter des objets de la bibliothèque à une scène, lorsqu'ils possèdent les autorisations nécessaires.

Enfin, la dernière partie de ce schéma de base de données est celle qui gère les coordonnées des objets. Pour cela nous stockons pour chaque objet graphique le système de référence dans lequel ses géométries sont définies. Cette information permet de recalculer les coordonnées géographiques dans un système global lorsque, dans un même modèle de données, ces informations ne sont pas homogènes (dans le cas où un modèle de données est construit à partir de plusieurs sources différentes par exemple). Sur la Figure 78, nous y voyons la classe CRS qui contient le système de référence géodésique dans lequel les géométries de l'objet associé sont définies. Cette classe contient une matrice de transformation qu'il nous suffit d'appliquer aux objets pour obtenir leur position dans le système de référence terrestre du modèle de données sémantique (associé au contexte de l'utilisateur).

L'ensemble de ces différentes classes permet de procéder au stockage des géométries pour la description des objets du modèle d'information urbain et peut être exploité dans le cadre de différentes fonctionnalités décrites tout au long de ce mémoire. L'architecture de streaming mise en place repose également sur ce schéma. Comme le montre la Figure 79, du côté serveur nous avons la base de données avec un schéma de modèle d'information défini dans l'ontologie, et côté client un modèle mémoire permet de stocker des extraits du graphe de scène avec les mêmes informations que dans la base de données. Ces informations sont dans un premier temps copiées en mémoire centrale du poste utilisateur, puis recopiées dans le cache lorsque cela est nécessaire. L'avantage de ce modèle mémoire est qu'il peut extraire des informations partielles de la base de données (par exemple un sous-graphe de scène ou bien une seule représentation d'un objet qui en possède plusieurs). C'est à partir de ce modèle mémoire que le moteur de rendu récupère les informations.

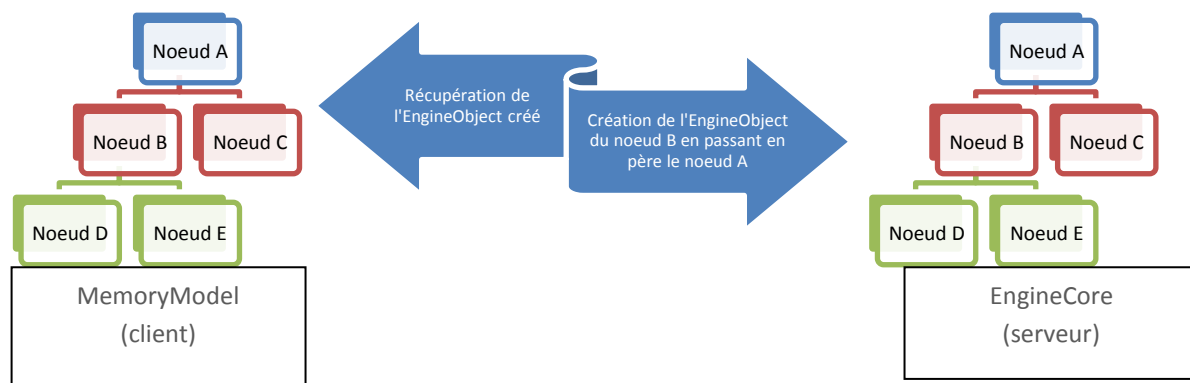


Figure 79. Mise en interface MemoryModel/EngineCore

3.3 Définition de l'ontologie

Dans les explications précédentes, il est souvent fait référence à l'ontologie SIGA3D. Cette ontologie, que nous avons décrite et formalisée dans notre approche pour décrire la conception d'un modèle d'information urbain avec des outils du web sémantique, n'a pas été implémentée telle quelle. Il a, en effet, fallu s'adapter aux contraintes industrielles du projet. Nous avons implémenté les différents concepts ontologiques à l'aide de schémas de bases de données relationnelles. La Figure 80 est un extrait de ce schéma. Celui-ci est centré autour de l'utilisateur afin de compléter l'architecture SIGA3D. En effet, cela permet de définir facilement un contexte pour chaque utilisateur avec les différents paramètres formalisés dans le chapitre de l'approche SIGA3D, et d'effectuer des requêtes facilement en fonction des propriétés de celui-ci.

De plus, deux types de classes apparaissent dans ce schéma :

- celles qui décrivent le niveau conceptuel de l'ontologie (les classes « templates » telles que GTP_VIEW_TEMPLATE pour la définition conceptuelle des vues, la classe GTP_OBJ_TEMPLATE que nous avons vue dans le paragraphe précédent pour la définition conceptuelle des objets, et ainsi de suite pour les différents types de concepts que peut définir l'ontologie comme les relations, les attributs...),
- et celles qui décrivent le niveau factuel de l'ontologie (pour chaque classe « template » il y a une classe qui sert à représenter les faits de l'ontologie, une instantiation des concepts en quelque sorte).

Cette métastructure de données permet aux utilisateurs de SIGA3D de créer leurs propres modèles d'information urbains. D'un point de vue industriel, en exploitation, pour chacun de nos clients, un groupe d'administrateurs est spécialement formé pour se charger de créer un modèle d'information urbain propre au client et qui sera ensuite utilisé (peuplé, visualisé, modifié...) par les différents utilisateurs.

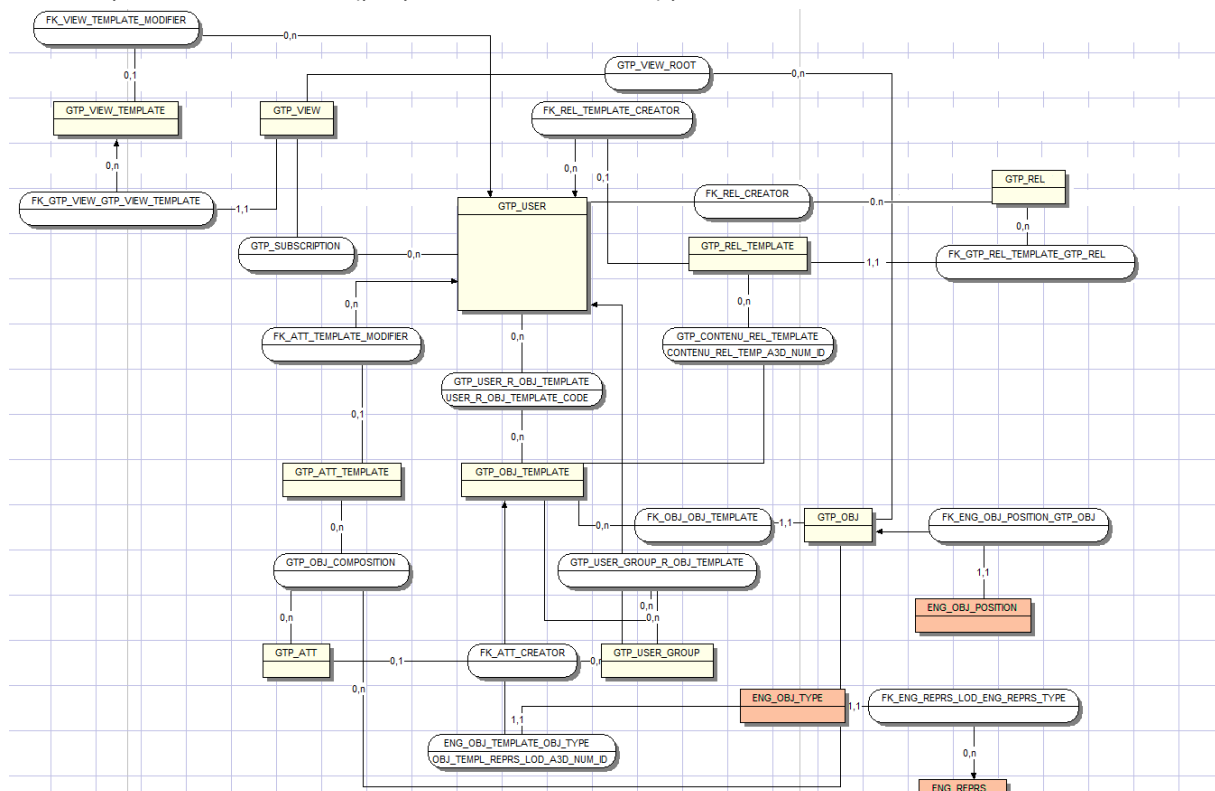
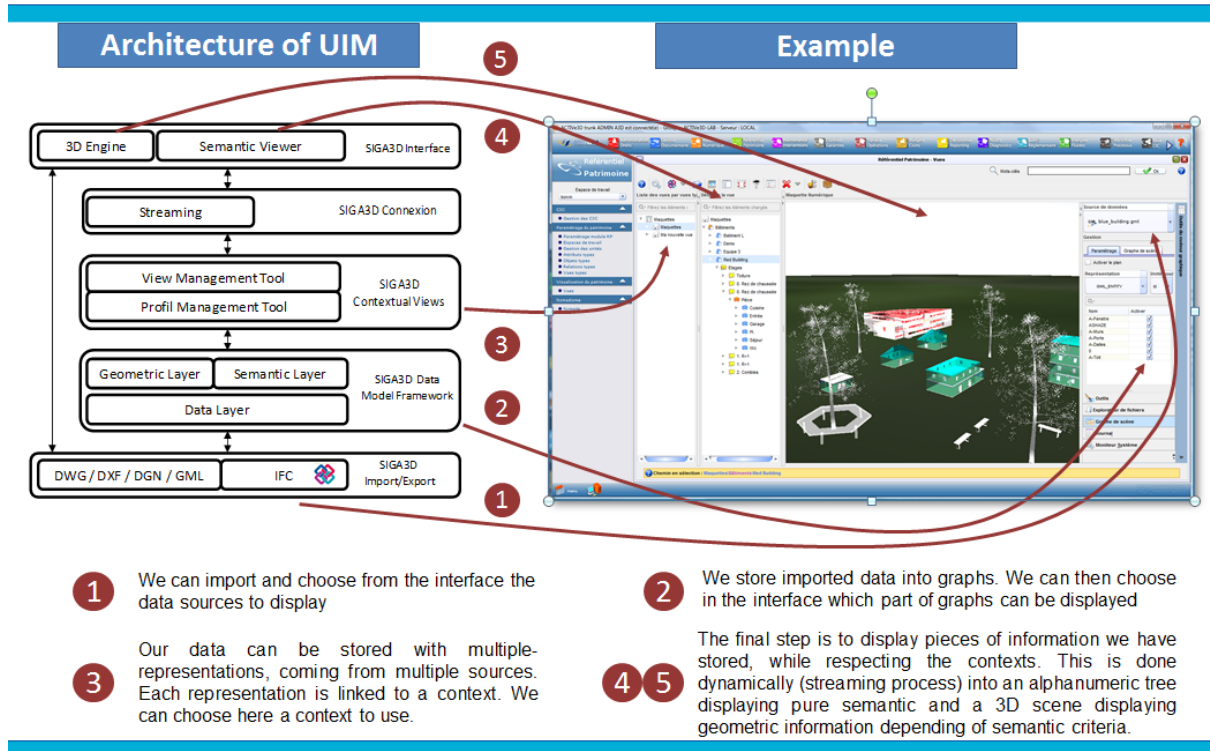


Figure 80. Extrait du schéma conceptuel de base de données de la GTP

4 Conclusion

Ce chapitre a présenté les différents points de développement des travaux de recherche menés dans le cadre de cette thèse. La Figure 81 issue de (Mignard et al., 2011) présente une capture d'écran de l'application dans laquelle est chargée un extrait de patrimoine urbain composé de plusieurs modèles d'information immobilier (chaque bâtiment affiché est basé sur son propre modèle – concepts et relations). Il est notamment fait le lien entre les éléments de l'interface utilisateur (graphe de scène à droite (2), listes des données chargées depuis la base de données ou des fichiers au-dessus (1), moteur 3D au centre (5), listes des vues contextuelles à gauche (3), et arborescence du patrimoine selon la vue contextuelle sélectionnée entre les deux précédents éléments (4)) et l'architecture SIGA3D simplifiée. Cela permet de voir à quel niveau intervient chaque élément de l'architecture, et comment en sont développés les différents éléments.



- 1 We can import and choose from the interface the data sources to display
- 2 We store imported data into graphs. We can then choose in the interface which part of graphs can be displayed
- 3 Our data can be stored with multiple-representations, coming from multiple sources. Each representation is linked to a context. We can choose here a context to use.
- 4 5 The final step is to display pieces of information we have stored, while respecting the contexts. This is done dynamically (streaming process) into an alphanumeric tree displaying pure semantic and a 3D scene displaying geometric information depending of semantic criteria.

Figure 81. Correspondance entre l'architecture SIGA3D et les éléments de l'interface utilisateur

Ce travail de recherche et de développement réalisé dans le cadre d'un projet Européen a permis à la société de disposer d'un environnement de gestion technique de patrimoine urbain sous la forme d'une preuve de concept. Il reste maintenant à l'équipe de développement d'Active3D de s'approprier les résultats de R&D pour développer cette plateforme de gestion du patrimoine urbain et la commercialiser.

Conclusion & Travaux Futurs

Résumé

La conclusion de cette thèse va maintenant faire le point sur ce qui a été fait et décrit dans ce document. En particulier, je mets en correspondance les objectifs du projet décrits en introduction avec la proposition formulée dans ma thèse afin de montrer comment ceux-ci ont été atteints. Je conclus en donnant un aperçu des travaux qu'il reste à réaliser dans ce vaste domaine des systèmes de gestion technique adaptée à la ville.

Le travail de recherche présenté dans ce mémoire définit une nouvelle approche de gestion technique de biens immobiliers et urbains. Pour cela, nous définissons un processus de production et de gestion de l'information du bâtiment, de son environnement et des objets extérieurs, tout au long de leur cycle de vie, au sein d'un concept que nous avons appelé Urban Facility Management, ou gestion technique de patrimoine urbain. Nous avons en particulier créé un Modèle d'Information Urbain (UIM pour Urban Information Model, par analogie au Building Information Model) qui nous permet de modéliser dans une ontologie toute l'information de la ville et du bâtiment.

Le projet qui porte ces travaux, SIGA3D, est à la fois un projet de recherche de l'entreprise Active3D en collaboration avec le laboratoire LE2I, mais également un projet européen de coopération entre Active3D en France et lat/lon en Allemagne. C'est grâce à toutes ces compétences réunies, réparties entre professionnels et chercheurs, que nous avons pu à la fois définir une architecture sémantique pour stocker et gérer l'ensemble des informations, sémantiques, géométriques et géographiques, d'un patrimoine immobilier et urbain, mais aussi en implémenter une grande partie au sein de la plateforme existante Active3D.

Nous avons commencé notre travail de recherche par identifier clairement les objectifs du système de gestion technique urbaine que nous voulions construire, puis les contraintes et difficultés principales liées au projet. L'approche que nous avons choisie pour réaliser nos objectifs, celle qui est la plus logique par rapport à l'environnement du projet (la plateforme Active3D de gestion technique de patrimoine immobilier), est l'extension et l'adaptation du framework de modélisation des données décrit par Renaud Vanlande pour les IFC à la gestion d'objets géographiques. Nous nous positionnons donc comme une architecture de médiation évolutive qui intègre les IFC/BIM et les SIG.

La proposition que nous avons faite consiste en une architecture à six couches permettant de résoudre les problématiques de manière systémique :

- La couche d'import/export des données permet de résoudre le problème d'hétérogénéité syntaxique et schématique créé par la diversité des sources de données.
- La couche de définition de modèle permet à la fois de regrouper différents formalismes du web sémantique nécessaires à la structuration de notre ontologie pour en simplifier la description, mais aussi de résoudre en partie l'hétérogénéité sémantique de nos données. De plus, c'est à ce niveau que sont gérées les représentations multiples pour nos objets, tant sémantiques que géométriques. Les niveaux de détail, qui sont un apport important de nos travaux sont défini à ce niveau, grâce aux contextes locaux.
- La couche de gestion des contextes permet de résoudre les contraintes de montée en charge des données sémantiques en permettant la définition de modèles à partir d'opérateurs de graphe. Cela permet d'alléger la structure en évitant de nombreuses redondances dans la description des modèles. Ce niveau permet également de rendre cohérent l'ensemble des données en définissant un contexte sur les modèles (informations sur l'utilisateur, les versions, les mises à jour...), sur les relations entre modèles et sur les systèmes de référence des modèles (contraintes provenant de la gestion de l'espace (y compris les problématiques de géoréférencement) et du temps dans l'ontologie).
- La couche de connexion permet essentiellement de pallier aux contraintes industrielles liées à la portabilité de l'architecture (possibilité de l'adapter à d'autres outils d'exploitation), aux performances (machine des utilisateurs peu puissante temps de chargement longs...). Elle contribue également à la résolution du problème des coordonnées en permettant de transformer, si nécessaire, les coordonnées d'objets du modèle à afficher en fonction du contexte.
- La dernière couche de l'architecture de l'UIM est celle qui affiche les données. La contrainte forte à ce niveau est de pouvoir afficher un nombre important d'informations. En effet, si dans l'approche

existante le BIM représente en général un bâtiment, l'UIM regroupe, entre autres, un ensemble de bâtiments, ce qui augmente rapidement le nombre d'objets géométriques à traiter. Le mécanisme évolué des niveaux de détail permet d'optimiser les scènes affichées, soit de manière traditionnelle en utilisant les niveaux de détails géométriques classiques ou même hiérarchiques, soit en utilisant les niveaux de détail contextuels, sélectionnés en fonction du profil et des préférences de l'utilisateur.

Nous avons proposé également une architecture de processus, afin non seulement de comprendre le fonctionnement de l'architecture intégrée à un workflow, ainsi que sa mise en place, mais aussi d'homogénéiser les outils pour chaque intervenant autour du projet, celui-ci étant dédié avant tout au travail collaboratif.

L'ensemble de ces architectures, les possibilités de modélisation, d'évolution, et de travail collaboratif qu'elles définissent et supportent permettent de répondre parfaitement aux objectifs du projet définis en introduction (dans la suite du paragraphe, nous mettons en gras les objectifs et indiquons entre parenthèses sa position dans la liste des objectifs de l'introduction). Tout d'abord, la **définition de modèles de données (1)** est rendue possible grâce à l'architecture basée sur C-DMF que nous avons détaillée. C'est le cœur même de notre proposition. Celui-ci étant basé sur des langages du web sémantique, notamment le formalisme RDF, des **inférences (2)** sont tout à fait possibles dans l'ontologie. La traduction de tous les opérateurs de définition du modèle en logique de description en atteste également. Les données de l'ontologie ainsi créées peuvent être **manipulées (3)** librement, ce qui est notamment décrit dans l'architecture des processus. Ensuite, les opérateurs de graphes du niveau contextuel que nous utilisons dans notre architecture logicielle permettent de simplifier l'ajout de nouveaux modèles ou la modification de modèles existants tout en évitant de les réécrire complètement. Outre les avantages cités ci-dessus, cela rend possible une **évolutivité (4)**, une maintenabilité et une **traçabilité (8)** facilitée des modèles de données. Ce mécanisme des opérateurs de graphes permet également de définir des modèles de graphes qui associent des graphes différents selon le **contexte d'utilisation (11)**, comme la **langue (9)** par exemple. Ces contextes d'utilisation ont en particulier été décrits en détail dans notre architecture de processus, au niveau de l'affichage des données : l'information peut être affichée textuellement ou en 3D, de manière contextuelle, soit par l'organisation des vues qui les affiche, soit par la représentation sélectionnée pour l'affichage. De plus, le mécanisme de chargement dynamique des données, associé à une structure de cache performante structurée selon les informations du modèle de données étudié, permet une utilisation de cette architecture en **environnement nomade (12)**, dans des endroits où la bande passante est limitée, ou sur des machines peu puissantes.

L'un des objectifs du projet, et apport principal par rapport à la plateforme BIM existante, était la gestion des données dans et hors du bâtiment, ce qui implique la **prise en compte des objets géographiques (5)**, mais aussi la définition et le traitement de l'information spatio-temporelle. Notre extension de C-DMF vise essentiellement à remplir cet objectif, et va même beaucoup plus loin en permettant de définir plusieurs représentations géométriques à un objet, plusieurs représentations sémantiques à ce même objet, et de coupler les représentations pour créer les niveaux de détail contextuels. Ces derniers peuvent très bien être utilisés pour reproduire le fonctionnement des niveaux de détail géométriques classiques. Ils permettent également à la **grande quantité d'information (6)** présente dans l'ontologie et dans la base de données géométrique associée d'être affichée au sein d'une scène 3D composée de ces objets. Les concepts spatio-temporels décrits ici, et en particulier les relations topologiques spatiales et temporelles, peuvent être utilisés dans la couche de gestion du contexte pour décrire des **relations entre les modèles de données (13)**.

Enfin, ce modèle d'information a été fait pour être compatible avec les solutions existantes et pour communiquer avec d'autres logiciels comme les SIG ou des logiciels de CAO. Pour cela, une étude poussée sur

les normes dans ces domaines a été réalisée, et l'architecture définie est suffisamment générique pour pouvoir accepter de **nombreux formats de données (10)**, notamment les modèles d'**information normalisés (7)** tels que les IFC ou CityGML. En particulier, la solution que nous avons choisie pour le traitement des sources de données géographiques, deegree, est une implémentation officielle des standards de l'OGC. Nous avons également conçu notre système dans le respect des normes étudiées.

Cette proposition d'architecture que nous avons faite a été en partie implémentée au sein de la plateforme existante Active3D. L'environnement industriel pour cette intégration, associé au contexte de crise sur les marchés financiers depuis 2008 et qui impacte grandement les entreprises, a contraint à des restrictions budgétaires qui impactent directement le projet ; ainsi, hors cadre du projet européen, le nombre d'ingénieurs ayant pu participer au développement des fonctionnalités de SIGA3D a été limité (nous étions deux permanents), et le choix de technologies payantes (comme le module Spatial d'Oracle) dont le coût a un impact à la fois sur le prix de vente directe de la solution, mais aussi des répercussions indirectes sur l'aménagement des infrastructures existantes et de l'administration réseau ont dû être mises de côté. Cela étant, nous avons pu développer une solution personnalisée et adaptée à nos besoins.

Ainsi, la plateforme Active3D intègre aujourd'hui de nombreux développements effectués dans le cadre du projet SIGA3D, à commencer par un nouveau moteur graphique, optimisé par rapport à celui existant qui n'affichait qu'un bâtiment au format IFC à la fois. Celui-ci gère maintenant, entre autres optimisations, les niveaux de détail ce qui permet de charger plusieurs bâtiments ou objets 2D et 3D, depuis plusieurs formats différents (IFC, DWG, DXF, DGN, shapefile, GML...) et de les positionner sur une carte issue de services web, le tout de façon dynamique depuis la base de données Active3D que nous avons détaillée dans le chapitre précédent. Il est également possible de définir un contexte utilisateur et des vues sémantiques afin d'afficher de l'information contextualisée, aussi bien dans le moteur graphique que dans l'arbre alphanumérique.

Toutes ces fonctionnalités développées ont été intégrées aux nombreuses fonctionnalités de la plateforme Active3D. Il est possible d'effectuer de nombreuses opérations sur la base de connaissance : enrichissement du Référentiel Patrimoine, bibliothèque d'objets normalisés CIC (Composants Installations Constituants) (qui utilise le mécanisme de stockage d'objets types associés à des propriétés et une représentation par défaut décrit dans cette thèse), traitement automatisé du suivi des actions de maintenance préventives, réglementaires, visites..., un environnement de gestion, d'optimisation, de simulation et de saisie manuelle des informations de consommation, capable de remonter, en temps réel, des informations de comptage d'énergie, d'alertes de surconsommation, la gestion documentaire pour le travail collaboratif, la gestion des Interventions (de maintenance ou remontée par les utilisateurs) et l'aide à l'arbitrage, une gestion des droits centralisée et personnalisée sur chaque fonctionnalité de l'application, ainsi que de nombreuses autres...

Comme nous pouvons le voir, les opérations qu'il est possible d'effectuer sur l'ensemble du patrimoine sont nombreuses. Il reste cependant de nombreux axes de recherche et développement à faire autour de ce projet. L'axe principal d'évolution de notre plateforme d'UIM est certainement le développement de fonctionnalités et d'outils dédiés à la gestion urbaine. En effet, si Active3D propose désormais une offre complète spécialisée autour de la gestion de bâtiment, dont de nombreuses peuvent être utilisées sur la partie urbaine de l'ontologie, les opérations spécifiques possibles sur le modèle d'information urbain restent limitées. Dans un premier temps, nous pensons qu'une mise en place de requêtes géographiques serait un vrai apport pour la plateforme, avec notamment la possibilité d'effectuer des statistiques sur l'information sémantique liée à la géographie et au positionnement des objets, voire aux relations topologiques puisque cette possibilité a été décrite dans l'architecture (un exemple de requête pourrait être : combien y a-t-il d'arbres le long de cette route ? En partant du principe que les relations d'adjacence entre la route en question et les arbres qui la

bordent ont été correctement définies). De manière plus générale, un des axes d'amélioration prévu est une intégration plus poussée et une meilleure exploitation de la spatialité et de la temporalité au sein de notre modèle.

Tout cela demande bien sûr de faire évoluer le modèle d'implémentation. Nous identifions, dans un premier temps, deux axes d'évolution possible : la prise en compte d'une modélisation topologique plus évoluée, le mécanisme décrit dans ce document étant relativement simple, et l'amélioration des algorithmes et processus de géoréférencement afin de prendre en compte la déformation des objets lors d'un changement de référentiel. Ces deux améliorations couplées permettraient d'obtenir un modèle plus juste et cohérent que l'approche, néanmoins nécessaire, que nous avons faite.

Enfin, les choix effectués pour le développement de l'architecture montrent la difficulté et la réticence des entreprises à adopter les outils du web sémantique en environnement industriel. Un second axe d'amélioration de notre infrastructure serait de retravailler la base de connaissance effectivement implémentée afin d'utiliser ces outils ontologiques, tel qu'il a été défini dans la formalisation de notre proposition. Cela permettrait d'améliorer le contrôle de l'information stockée et offrirait des fonctionnalités supplémentaires en termes de contextualisation et de recommandations. La prise en charge de la montée en charge et du dimensionnement de la base en serait grandement améliorée et les requêtes, qui pourraient être plus facilement contextualisées, seraient plus efficaces. Toutes ces suggestions pourront être réalisées rapidement grâce à l'engouement des institutions et des personnes suscité par de tels projets, mais aussi par l'activité grandissante, à la fois des chercheurs et des industriels, sur ce sujet.

Bibliographie

- (Abdul-Rahman, 2007) Alias Abdul-Rahman and Morakot Pilouk. Spatial data modelling for 3D GIS, chapter 2. Springer, 2007.
- (Allen, 1983) J. Allen, Maintaining knowledge about temporal intervals, *Commun. ACM* 26 (11) (1983) 832–843
- (Altamimi, 2006) Zuheir ALTAMIMI. Systèmes de référence terrestres : définition, réalisation, application à l'ITRF, état actuel et perspectives. Habilitation à diriger des recherches (hdr), Université Pierre et Marie Curie (Paris VI), 2006.
- (Baader et al., 2003) F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge University Press, Cambridge, UK, 2003. ISBN 0-521-78176-0
- (Barrett et al., 2003) Peter Barrett and David Baldry, *Facilities Management, Towards Best Practice*, Blackwell Publishing, 2003
- (Batty et Smith, 2009) Urban Modelling, Michael Batty, A. Smith, Spatial division of labour, In: R.Kitchin and N.Thrift (eds) *International encyclopedia of human geography*, Elsevier, London
- (Baxter, 2000) William Baxter. Occlusion culling for walkthroughs of large virtual environments. 2000. <http://billbaxter.com/projects/occlusion.html>
- (Bazjanac et al., 2011) Bazjanac, V., Rose, C. M., Maile, T., O'Donnell, J. T., Mrazović, N., Morrissey, E. and Welle, B.R. An assessment of the use of building energy performance simulation in early design. *Proc. Building Simulation 2011*. Sydney, Australia, 2011.
- (Benner et al., 2005) Benner, J., Geiger, A. and Leinemann, K., Flexible Generation of Semantic 3D Building Models. In: Gröger, G. et al. (Eds.): *Proceedings of the 1st International Workshop on Next Generation 3D City Models*, Bonn, Germany, 2005.
- (Benslimane et al., 2003) D. Benslimane, C. Vangenot, C. Roussey, A. Arara. Multi-representation in ontologies, *Proceedings of 7th East-European Conference on Advances in Databases and Information Systems*, 2003
- (Berlo, 2009) Van Berlo, L., *CityGML extension for Building Information Modelling (BIM) and IFC*. Free and Open Source Software for Geospatial (FOSS4G), Sydney, Australia, 2009
- (Berlo et Laat, 2010) Van Berlo L., de Laat R., "Integration of BIM and GIS: The development of the CityGML GeoBIM extension", In: *Proceedings of the 5th International 3D GeoInfo Conference*, Berlin, Germany, 2010
- (Berstel et al., 2000) Berstel J., Boasson L., *XML Grammars*. MFCS 2000: pp182-191, 2000.
- (Brodeur, 2010) Jean Brodeur, Road map semantics - ontology, 31st plenary meeting of ISO/TC 211, Australia, December 7, 2010
- (Carroll, 2005) J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. In *Proceedings of the 14th International World Wide Web Conference (WWW)*. ACM Press, May 2005.
- (Chena et al., 2004) Po-Han Chena, Lu Cuia, Caiyun Wana, Qizhen Yangb, Seng Kiong Tinga, and Robert L.K. Tiong. Implementation of ifc-based web server for collaborative building design between architects and structural engineers. *Automation in Construction* 14, 2004.
- (CityGML, 2010) CityGML – ADE (2010), CityGML application domain extensions. Retrieved April 20, 2010 from the World Wide Web: <http://www.citygmlwiki.org/index.php/CityGML-ADEs>
- (Clemen et Gründig, 2006) Christian Clemen and Lothar Gründig. The industry foundation classes – ready for indoor cadastre ? In *TS 48 - Engineering Surveys for Construction Works II*, 2006.
- (Clarke, 1997) *Getting started with geographic information systems*, Clarke, K.C, Prentice Hall Upper Saddle River, 1997.
- (Coors, 2003) Coors, V., *3D GIS in Networking environments*, Elsevier, Amsterdam, (Special Issue 3D Cadastre), pp 345-357, 2003.
- (Cox et al., 2004) Simon Cox, Paul Daisey, Ron Lake, Clemens Portele, and Arliss Whiteside. *OpenGIS Geography Markup Language (GML) Implementation Specification*. OpenGIS Recommendation Paper, OpenGIS, février 2004.
- (Cruz, 2004a) Christophe Cruz, *Intégration et Manipulation de Données Hétérogènes au Travers de Scènes 3D Dynamiques, Evolutives et Interactives. Application aux IFC pour la Gestion Collaborative de Projets de Génie Civil.* », Thèse de doctorat, Université de Bourgogne, 2004

- (Cruz, 2004b) Christophe Cruz, Renaud Vanlande, Christophe Nicolle, Modélisation sémantique pour la génération dynamique de scènes 3D Application aux projets du bâtiment, Sciences of Electronic, Technology of Information and Telecommunications SETIT'04, IEEE, Sousse, Tunisie, 2004.
- (Cruz et Nicolle, 2006) Cruz C., Nicolle C., "Active3D: Vector of Collaboration, Between Sharing and Data Exchange.", INFOCOMP, Journal of Computer Science, 5 (3), pp. 1-8, January 2006.
- (Cullot et al., 2002) Nadine Cullot, Christine Parent, Stefano Spaccapietra, and Christelle Vangenot, Ontologies : À contribution to the DL/DB debate, 2002
- (Cullot et al., 2003) Nadine Cullot, Christine Parent, Stefano Spaccapietra, Christelle Vangenot, Des SIG aux ontologies géographiques, Revue Internationale de Géomatique. Numéro spécial "Les SIG sur le Web", Vol. 13 (No. 3), pp. 285-306, Hermès Science Edition, 2003
- (Davis et Laender, 1999) Clodoveu A. Davis, Jr, Alberto H. F. Laender, Multiple representations in GIS: materialization through map generalization, geometric, and spatial analysis operations, In Proceeding GIS '99 Proceedings of the 7th ACM international symposium on Advances in geographic information systems, pp 60 – 65, 1999.
- (De La Losa, 2000) Arnaud De La Losa. Modélisation de la troisième dimension dans les bases de données géographiques. Thèse de doctorat, Université de Marne-La-Vallée, 2000.
- (Devogele, 1997) Thomas Devogele, Processus d'intégration et d'appariement de Bases de Données Géographiques, Application à une base de données routières multi-échelles, Thèse de doctorat, Université de Versailles, 1997
- (ECCE, 2002) European Council of Civil Engineers, Competitiveness Working Group, Phase II Information and Communication Technologies 2001 - 2002 Preliminary Report, March 2002
- (Erikson et al., 2001) Carl Erikson, Dinesh Manocha, and William V. Baxter III. Hlods for faster display of large static and dynamic environments. In Symposium on Interactive 3D Graphics, 2001.
- (Ferries et Légglise, 2006) Bernard Ferriès, Michel Légglise, Frictions between heterogeneous worlds, 3rd International Conference on Virtual City and Territory, Bilbao (Spain), 2006
- (François et al., 2010), A. François, R. Raffin, M. Daniel, GEOMETRIC DATA STRUCTURES AND ANALYSIS IN GIS: ISO 19107 CASE STUDY, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15, 2010
- (Germs et al., 1999) Germs, R., G. V. Maren, E. Verbree, and F. W. Jansen, A Multi-view VR Interface for 3D GIS, Computer & Graphics, 23:497-506, 1999.
- (Gesbert, 2005) Nils Gesbert, Étude de la formalisation des spécifications de bases de données géographiques en vue de leur intégration, Thèse de doctorat, Université de Marne-La-Vallée, 2005
- (Gruber, 1993) Gruber, T. R., Toward principles for the design of ontologies used for knowledge sharing, International Workshop on Formal Ontology, Padova, Italy, 1993
- (Guha, 2004) R. Guha, R. Fikes, R. McCool, *Contexts for the semantic web*, In Proceedings of ISWC'2004, 2004.
- (Guttman, 1984) Guttman, Antonin. R-Tree: A Dynamic Index structure for Spatial Searching. Proceedings of ACM SIGMOD Conference on Management of Data. 1984.
- (Hamilton et al., 2005) Hamilton A, Wang H, Tanyer A M, Arayici Y, Zhang X and Song Y (2005) Urban information model for city planning, ITcon Vol. 10, Special Issue From 3D to nD modelling , pg. 55-67, 2005
- (Hayat et al., 2007) Khizar Hayat, William Puech, Gilles Gesquière, An Efficient Data-hiding Method Based on Lossless JPEG2000 for a Scalable and Synchronized Visualization of 3D Terrains, EUSIPCO 07: 15th European Signal Processing Conference, Poznan, Poland, 2007
- (Herold et al., 2003) Herold M., Goldstein N.C., Clarke K.C., The spatiotemporal form of urban growth: Measurement, analysis and modeling (2003) Remote Sensing of Environment, 86 (3), pp. 286-302.
- (Hobbs et Pan, 2006) Hobbs, J.R., and Pan, F., Time Ontology in OWL. Ontology Engineering Patterns Task Force of the Semantic Web Best Practices and Deployment Working Group, World Wide Web Consortium (W3C) notes. 2006. (<http://www.w3.org/TR/owl-time/>)
- (Hogenboom et al., 2010) Hogenboom, F., Borgman, B., Frasinca, F. and Kaymak, U., Spatial Knowledge Representation on the Semantic Web, In : ICSC '10 Proceedings of the 2010 IEEE Fourth International Conference on Semantic Computing, IEEE Computer Society Washington, DC, USA, 2010.
- (Housseno, 2009) S. Housseno, A. Simonet, and M. Simonet, "Ub-tree indexing for semantic query optimization of range queries" International Journal of Computer, Information and Mechatronic Engineering, issue 35, vol. 59, pp. 177–184, 2009.

- (Huang, 2011), Huang M. and Tian Y., Semantic Web Technologies Based Geospatial Services Discovery and Data Retrieval, In Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011) November 19–20, 2011, Melbourne, Australia
- (Hu, 2008) Ming Yuan Hu, Semantic Based LOD Models Of 3D House Property, Proceedings of Commission II, ISPRS Congress Beijing, 2008.
- (Huot, 2005) Stéphane Huot, Une nouvelle approche pour la conception créative : De l'interprétation du dessin à main levée au prototypage d'interactions non-standard, thèse de doctorat, 2005
- (IFC, 2005) Norme ISO/PAS 16739:2005, Industry Foundation Classes, Release 2x, Platform Specification (IFC2x Platform), 2005, révisé en 2012 par le projet de norme internationale ISO (PAS) DIS 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries
- (IGN, 2008a) IGN. Systèmes géodésiques.
http://professionnels.ign.fr/DISPLAY/000/527/311/5273119/Geodesie_systemes.pdf, 2008.
- (IGN, 2008b) IGN. Rappel historique sur la notion de géodésie.
http://education.ign.fr/DISPLAY/000/527/315/5273150/geodesie_histoire.pdf, 2008.
- (IGN, 2008c) IGN. Systèmes de coordonnées.
http://education.ign.fr/DISPLAY/000/527/312/5273120/geodesie_coordonnees.pdf, 2008.
- (Isikdag et Zlatanova, 2009) Isikdag, S. and Zlatanova, S. : Towards defining a framework for automatic generation of buildings in CityGML using BIM, in Lee and Zlatanova (eds.), 3D geo-information sciences, LNG&C, Springer Verlag, pp. 79-96, 2009.
- (Isikdag et Zlatanova, 2009b) Isikdag, U. and Zlatanova, S.: A SWOT analysis on the implementation of BIM within geospatial environment, in Krek, Rumor, Zlatanova & Fendel (eds.) Urban and Regional data Management, UDMS Annuals 2009, CRC press, pp. 15-30, (2009).
- (ISO/TC 211, 2009a) ISO/TC 211 Geographic information/Geomatics, Standards Guide ISO/TC 211 Geographic information/Geomatics, 2009, <http://www.isotc211.org/opendoc/211n2726/>
- (ISO/TC 211, 2009b) ISO/TC 211 Geographic information/Geomatics , Report from stage 0 Project 19150 Geographic information-Ontology, 28th ISO/TC 211 plenary meeting, Norway, 2009, <http://www.isotc211.org/protdoc/211n2705/>
- (ISO/TC 211, 2010), ISO/TC 211 Geographic information/Geomatics, New work item proposal: ISO/TS 19150-1, Geographic information - Ontology - Part 1: Framework, 2010, <http://www.isotc211.org/protdoc/211n2983/>
- (Jeong et al., 2009), Y.-S. Jeong, C.M. Eastman, R. Sacks, I. Kaner, Benchmark tests for BIM data exchanges of precast concrete, Automation in Construction, 18 (2009), pp. 469–484
- (Khemlani, 2010) Lachmi Khemlani, The IFC Building Model, Civil Engineer Group, 2010.
<http://www.civilengineergroup.com/ifc-building-model.html>
- (Kim et al., 2009) Byung-Guk Kim, Hoon-Sung Choi et, Yeun J. Jung, Research and Development Program for the Third National Geographic Information System Project in Korea: Korean Land Spatialization Program, dans 3D Geo-Information Sciences, Lee, Jiyeong, Zlatanova, Sisi, pp33-45, 978-3-540-87395-2, 2009
- (Kolbe, 2009) Kolbe, Thomas H., Representing and Exchanging 3D City Models with CityGML, dans 3D Geo-Information Sciences, Lee, Jiyeong, Zlatanova, Sisi, pp15-31, 978-3-540-87395-2, 2009
- (Kolbe et al., 2009) Kolbe T., K. Nagel, A. Stadler. "CityGML – Standard in Photogrammetry?", 52nd Photogrammetric Week in Stuttgart, September 7-11, 2009.
- (Kolbe et Gröger, 2004) Kolbe, Th. and Gröger G., Unified Representation of 3D City Models. Geoinformation Science Journal, Vol.4, No. 1, 2004.
- (Königer et Bartel, 1998) A. Königer, S. Bartel, 3d-Gis for Urban Purposes, Geoinformatica, Springer Netherlands, pp. 79-103, Volume 2, 1998.
- (Koussa, 2011) Chokri Koussa, Implantation d'un système d'information géographique 3D sur Internet pour la gestion des modèles urbains, Thèse de doctorat, Université de Strasbourg, 2011.
- (Langlois, 1994) Langlois, P., Formalisation des concepts topologiques en géomatique, Revue internationale de géomatique, 4(2/1994), 181-205, 1994.
- (Laurini, 2001) Laurini, R., Real Time Spatio-Temporal Databases, Transactions in GIS, 5: 87–97, 2001.
- (CNIG, 2006) CNIG, Conseil National de l'Information Géographique, Profil français de la norme EN-ISO 19115, 2006, http://www.cnig.gouv.fr/Front/docs/cms/mise-en-oeuvre-norme-iso_124052510694329600.pdf

- (Li, 2011), Ke Li, Han Lin Zhang, Lin Du, Research on Technology of Semantic Fusion in Geospatial Information Based on Ontology, In *Advanced Materials Research*, 403-408, November 2011
- (Liebich , 2001) Thomas Liebich, IAI / IFC - Concept of the IFC Standard and the relation ISO and XML (Presentation), Presented at: Interoperability in AEC&FM Sydney, Australia 29.-30.10.2001
- (Liebich , 2010) Liebich, T., Unveiling IFC2x4 - The next generation of OPENBIM, Proceedings of the 2010 CIB W78 Conference, 2010.
- (Longley, 2005) Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W., *Geographic Information Systems and Science*, Chichester: Wiley, 2nd edition, 2005.
- (Lopez et al., 2011) López-Pellicer, FJ. and Vilches-Blázquez, LM. and Zarazaga-Soria, FJ. and Muro-Medrano, PR. and Corcho, Óscar (2011) The Delft Report: Linked Data and the challenges for geographic information standardization. In: *Jornadas Ibéricas de Infraestructuras de Datos Espaciales 2011 (JIIDE 2011)*, 9-11 noviembre 2011, Barcelona, España.
- (MacPherson et Raymond, 2010) Deborah L. MacPherson and Michelle A. Raymond, *Interoperability on a large scale*, Journal of Building Information Modeling, Spring 2010.
- (Mao, 2011) B. Mao, Y. Ban and L. Harrie, "A Multiple Representation Data Structure for Dynamic Visualisation of Generalised 3D City Models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66(2), 2011, pp. 198-208.
- (McCall, 1977) McCall, J.; Richards, P.; Walters, G., "Factors in Software Quality," three volumes, NTIS AD-A049-014, AD-A049-015, AD-A049-055, November 1977.
- (Mekawy et Östman, 2010) Mohamed El-Mekawy and Anders Östman. Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CITYGML. In Proceedings of the 3rd ISDE Digital Earth Summit 12-14 June, 2010, Nessebar, Bulgaria, 2010
- (Mekawy et Östman, 2012) El-Mekawy, Mohamed and Anders Östman. "Ontology Engineering Method for Integrating Building Models: The Case of IFC and CityGML." *Universal Ontology of Geographic Space: Semantic Enrichment for Spatial Data*. IGI Global, 2012. 151-185. Web. 5 Apr. 2012. doi:10.4018/978-1-4666-0327-1.ch007
- (Mignard et Nicolle, 2011) Clément MIGNARD, Christophe Nicolle, SIGA3D: Semantic Combination of IFC and GIS to support Urban Facilities Management, In *Handbook on Reserach of E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies*, Frank Dietter-Dorloff, Ivan Bedini and Ejub Kajan (Ed.), 2011.
- (Mignard et al., 2011) Clément Mignard, Gilles Gesquière, Christophe Nicolle Interoperability between GIS and BIM, International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, KMIS 2011, October 26-29, 2011, Paris, France
- (Miguel et al., 2002) Miguel, M., Jourdan, J. and Salicki, S. (2002), Practical Experiences in the Application of MDA, 5th International Conference on the Unified Modeling Language, Springer LNCS, Vol. 2460, pp: 128-139, Dresden, Germany
- (Moisuc et al., 2004) Moisuc, B., Gensel, J., Martin, H., Représentation de connaissances par objets pour les SIG à représentations multiples, Proc. Cassini-SIGMA 2004.
- (Molenaar, 1990) Martien Molenaar, A formal data structure for three dimensional vector maps, Proceedings of the 1st European Conference on Geographical Information Systems 1990 Amsterdam, The Netherlands, vol. 2, p. 770-781, 1990
- (Nagel et al., 2009) Nagel, C., Stadler, A. and Kolbe, T.: Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models, Academic Track of Geoweb 2009 Conference, Vancouver, 2009.
- (Nebiker, 2002), Nebiker, S., Multi-scale representations for scalable and dynamic 3D geoinformation services, Workshop on Data Generalisation and Data Mining, ISPRS Commission IV WG IV/3, Ottawa, Canada, 2002.
- (Nicolle, 2001), Christophe Nicolle, A Metamodel Specification Tool Based on Description Logic and XML-DATA », *Columbian Journal of Computation*, ISSN 1657 – 2831, pp. 1 – 19, Vol. 2, N° 1 June 2001.
- (Nicolle, 2003) Christophe Nicolle, Kokou Yétongnon, Jean-Claude Simon, « XML Integration and toolkit for B2B applications », *Journal of DataBase Management, Special Issue on Data Management for e-business*, Sudha Ram & Akhilesh Bajaj Editors, pp 33-58, Vol.14, No.4, October-December 2003.
- (Nicolle, 2005), C. Nicolle, Interopérabilité des Systèmes d'Information, habilitation à diriger des recherches, Université de Bourgogne, 2005
- (Parent et al., 1998) Christine Parent, Stefano Spaccapietra, Esteban Zimányi, Pier Donini, Corinne Plazanet, Christelle Vangenot, Nadia Rognon, Pierre-André Crausaz, MADS, modèle conceptuel spatio-temporel, 1998
- (OGC, 2006) OGC 05-102r1, OWS 3 GML Topology Investigation, Version 0.0.5, OpenGIS® Discussion Paper, Mars 2006

- (Oosterom, 1994) Peter Van Oosterom, Wilco Vertegaal, Marcel Van Hekken, Integrated 3D modelling within a GIS, in Advanced geographic data modelling, Martien Molenaar Sylvia de Hoop, p.80-95, 1994.
- (OpenGIS, 2008) OpenGIS, editor. OpenGIS City Geography Markup Language (CityGML) – Implementation Specification, chapter 8. Open Geospatial Consortium Inc., 2008.
- (Poupeau, 2008) Benoît Poupeau. Analyses et Requêtes de données géographiques 3D – Contribution de la cristallographie Géométrique -. Thèse de doctorat, Université Paris-Est - Marne-La-Vallée, 2008. chapter 7.
- (Quillian, 1968) Quillian, R., Semantic memory, *Semantic information processing*, 227-270, 1968
- (Ramos, 2003) Fabien Ramos. Modélisation et validation d'un Système d'Information Géographique 3D opérationnel. Thèse de doctorat, Université de Marne-La-Vallée, 2003.
- (Recanati, 93) Recanati F., *Direct Reference: From Language to Thought*, Blackwell Publishers, Oxford, UK, 1993.
- (Schevers et al., 2007) Schevers H., Mitchell J., Akhurst P., Marchant D., Bull S., McDonald K., Drogemuller R., Linning C. (2007), "Toward Digital Facility Modelling fo Sydney Opera House using IFC and Semantic Web Technology", *ITcon Vol. 12*, 2007.
- (SCHOLL et al., 1996) Michel Scholl, Agnès Voisart, Jean-Paul Peloux, Laurent Raynal, Philippe Rigaux, *SGBD Géographiques, Spécificités*, International Thomson Publishing 1996.
- (Shen et al., 2010) W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani, H. Xue, Systems integration and collaboration in architecture, engineering, construction, and facilities management: a review, *Advanced Engineering Informatics*, in press, doi:10.1016/j.aei.2009.09.001.
- (Sheth, 1998) Sheth, Amit P. Changing focus on interoperability in information systems : From system, syntax, structure to semantics. In : Goodchild, M. F., Egenhofer, M. J., Fegeas, R. et Kottman, C. A. (dir.) *Interoperating geographic information systems*. Dordrecht : Kluwer academic, 1998.
- (Sotnykova et al. 2005) Sotnykova, A., Vangenot, C., Cullot, N., Bennacer, N., Aufaure, M.A.: Semantic mappings in description logics for spatio-temporal database schema integration. *Journal on Data Semantics III*, Special Issue (2005) to appear
- (Spaccapietra et al., 2002) S. Spaccapietra, C. Parent, and E. Zimányi, The MurMur project: Modeling and querying multi-representation spatio-temporal databases, 2002 ScienceDirect
- (Spaccapietra et al., 2004) S. Spaccapietra, N. Cullot, C. Parent, C. Vangenot. On Spatial Ontologies. 6th Brazilian Symposium On Geoinformatics, Campos do Jordao, Brazil, November 22-24, 2004.
- (Spagnuolo et Falcidieno, 2009) Spagnuolo, Michela and Falcidieno, Bianca, "3D Media and the Semantic Web." *IEEE Intelligent Systems*, 24(2), pp. 90-96, 2009.
- (Stangeland, 2011) Bjørn K Stangeland , *BIM Collaboration Format (BCF)*, buildingSmart, 2011.
http://iug.buildingsmart.com/resources/abu-dhabi-iug-meeting/IDMC_017_1.pdf/at_download/file
- (Stoter et al., 2008) Stoter J., Kluijver H., Kurakula V.: 3D Noise Mapping in Urban Areas, *International Journal of Geographical Information Science*, Vol. 22, No. 8, s.907–924, 2008, Elsevier
- (Sui, 1998) D.Z. Sui, GIS-based urban modeling: Practices, problems, and prospects. *International Journal of Geographic Information Sciences*, 12 (1998), pp. 651 à 671.
- (Towne, 2009) Nancy Towne, GIS and BIM interoperabilty, ESRI Federal User Conference, 2009.
- (Vangenot, 2004) Christelle Vangenot, Multi-representation in spatial databases using the MADS conceptual model, ICA Workshop on Generalisation and Multiple representation, 2004
- (Vanlande, 2007) Renaud Vanlande. « C-DMF : Une Architecture de Modélisation Sémantique des Données. Application à la Gestion Technique de Patrimoine Immobilier. », Thèse de doctorat, Université de Bourgogne, 2007.
- (Vanlande et al., 2008) Vanlande R., Nicolle C., Cruz C., IFC and building lifecycle management, *Automation in Construction*, 18 (1), pp. 70-78, 2008.
- (Vosselman, 2003) Vosselman, G., 3D reconstruction of roads and trees for city modeling, *Int. Arch. Photogramm. Remote Sens.*, 34 (part 3/W13) (2003), pp. 231–236, 2003
- (Wache et al., 2001) Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G, Neumann H and Hübner S. 2001. *Ontology-based integration of information - A survey of existing approaches*
- (Zaninetti, 2005) *Statistique spatiale - Méthodes et applications géomatiques*, Jean-Marc Zaninetti, Hermès – Lavoisier, 2005.

(Zeitouni, 1995) Zeitouni, K., Topological Modelling for 3D GIS. 4th International Conference on Computers in Urban Planning and Urban Management, Melbourne, Juillet 1995.

(Zlatanova, 2000) Siyka Zlatanova. 3D GIS for Urban Development. Thèse de doctorat, Graz University of Technology, 2000.

Rapports Techniques

SIGA3D European Project

1. SIGA3D : Minutes of Meeting in Bonn, Clément Mignard, Andreas Poth, Benoît Vervandier, 15/16 March 2010
2. IFC Guide, Clément Mignard, Marc Rastoix and Olivier Clercq, RT, Mars 2010
3. SIGA3D Technical Report, Clément MIGARD, Andreas Poth, Mars 2010
4. First explanations about the database MCD of A3D, Clément Mignard, SIGA3D, Avril 2010
5. Synthèse des évolutions du schéma par LatLon, Clément Mignard, SIGA3D, Mai 2010
6. Evolution of the database model and explanation of the link with our facility management data base, Clément Mignard, SIGA3D, Juin 2010
7. Debriefing of the mid-project meeting, Eléonore VENIN, Clément MIGNARD, Juillet 2010, Dijon
8. Coordinate Reference Systems used in France, Clément MIGNARD, RT, Août 2010
9. SIGA3D – Conséquence d'un import dynamique sur l'avancement du projet européen, Clément MIGNARD, SIGA3D, octobre 2010
10. 3D Engine Objectives for Active3D, Olivier Clercq, Marc Rastoix, Octobre 2010
11. SIGA3D – Detailed technical analysis report on expected tasks from Lat/Lon, Clément MIGNARD, SIGA3D, Novembre 2010
12. Analyse sur l'intégration de SIGA3D dans Active3D, Clément Mignard, Julien Reverdiau, Novembre 2010
13. Summary of advancement on working packages defined in SIGA3D, Clément Mignard, Février 2011
14. Analysis on deegree capacity and connecting, Clément MIGNARD, Andreas Poth, Juin 2011
15. Geo questions for SIGA3D, Clément MIGNARD, Andreas Poth, Juillet 2011.
16. Technical discussion about GML and CityGML import/export in SIGA3D, Clément MIGNARD, Olivier CLERCQ, Andreas Poth, Andreas Schmitz, Juillet 2011
17. SIGA3D cooperation architecture and processes, Clément MIGNARD, Juillet 2011
18. Paramétrage de l'import des données deegree dans Active3D, Olivier Clercq, Clément Mignard, RT, Juillet 2011
19. Minutes of the fourth plenary meeting, Joseph Lorang, Clément Mignard, Jens Fitzke, 22 Juillet 2011, Luxembourg
20. Document de travail, suivi de l'avancement des travaux, Clément MIGNARD, Oséo – SIGA3D, Novembre 2011

SIGA3D Recherche

1. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, Organisation des données, Clément Mignard, RT, 5 août 2009
2. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, La géodésie - ou comment mesurer la Terre, Clément Mignard, RT, 25 août 2009
3. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, La troisième dimension, Clément Mignard, RT, septembre 2009
4. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, Organisation des données, Clément Mignard, RT, 5 août 2009
5. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, La géodésie - ou comment mesurer la Terre, Clément Mignard, RT, 25 août 2009
6. Introduction aux principes de fonctionnement des Systèmes d'Information Géographique, La troisième dimension, Clément Mignard, RT, septembre 2009
7. C-DMF, caractérisation du BIM et définition du modèle de données avec gestion du contexte, Clément Mignard, d'après Renaud Vanlande, Février 2010
8. Etude préliminaire : objectifs de l'architecture SIGA3D, Clément Mignard, RT, Mars 2010
9. Les IFC, présentation et limites, Clément Mignard, RT, Avril 2010.
10. Architecture provisoire de SIGA3D, Clément Mignard, RT, Mai 2010
11. Planification du travail de thèse, Clément Mignard, RT, Juillet 2010

12. Introduction aux C-LoDs, Clément Mignard, RT, Août 2010
13. Présentation des évolutions de C-DMF, Clément Mignard, RT, Août 2010
14. Ontologie dynamique : Approche de l'évolution spatio-temporelle au niveau ontologique, Clément Mignard, Octobre 2010
15. Intégration du nouveau moteur 3D dans Active3D et support JME, Clément Mignard, Olivier Clercq, RT, Octobre 2010
16. Spécifications de l'import ODA, Marc Rastoix, Clément Mignard, RT, Novembre 2010
17. Réflexion sur l'évolution du nouveau moteur et l'import des données, Clément Mignard, Olivier Clercq, Marc Rastoix, RT, Décembre 2010
18. Paramétrage de l'import dynamique des données objets et non-objets, Marc Rastoix, Clément MIGNARD, RT, Avril 2011
19. Coupler le BIM et les SIG, Clément MIGNARD, RT, Mai 2011
20. Formalisation : L'Architecture clem-A, Clément MIGNARD, RT, Juillet 2011
21. Urban Information Modeling, Clément MIGNARD, RT, Juillet 2011
22. Introduction aux IFC, Clément MIGNARD, RT, Août 2011
23. Positionnement des travaux de recherche sur la modélisation d'environnement urbain, Clément MIGNARD, RT, Septembre 2011
24. SIGA3D : Introduction de la couche de gestion du contexte, Clément MIGNARD, RT, Septembre 2011
25. Architecture formalisée SIGA3D, Clément MIGNARD, RT, Septembre 2011
26. SIGA3D : Etat de l'art, Clément MIGNARD, RT, janvier 2012
27. SIGA3D : Approche : Structure générale – mécanismes de multireprésentation sémantique, Clément MIGNARD, RT, février 2012
28. SIGA3D : Architecture, Clément MIGNARD, RT, Mars 2012

Active3D

1. Mignard C., Mise en place d'un système de déploiement de la plateforme Active3D, septembre 2008.
2. Gestion de la 3D par JNI dans Java, Olivier Clercq, Clément Mignard, RT, Juillet 2009
3. Test et performances de la plateforme Active3D pour la Mairie de Paris, Julien Reverdiau, Clément Mignard, Janvier 2010
4. Wise Image, un moteur de l'intégration des formats CAD dans Active3D ?, Clément Mignard, Olivier Clercq, RT, Mars 2010
5. Rapport sur la mise en place de l'encryptage des mots de passé de la base de données Active3D, Clément Mignard, RT, Juin 2010
6. Dossier de spécifications détaillées, plateforme Active3D, module Maquette Numérique, Clément Mignard, Olivier Clercq, Marc Rastoix, Documentation technique Active3D, Juillet 2010
7. Rapport sur la mise en place de l'encryptage des mots de passe de la base de données Active3D, Clément Mignard, Septembre 2010
8. Analyse sur la gestion des textes 3D, Grégory Barbier, Clément Mignard, RT, Juillet 2011
9. Procédure d'accès et de tests de la plateforme A3D sur la machine 64bits dédiée
10. Analyse liée à la sécurité de l'authentification et des communications sur la plateforme Active3D, Clément Mignard, RT, Avril 2012

Publications

Chapitre de livre

SIGA3D: Semantic Combination of IFC and GIS to support Urban Facilities Management, C. Mignard, C. Nicolle, Handbook on Research of E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies, Frank Dietter-Dorloff, Ivan Bedini and Ejub Kajan (Ed.) (2011) 23, 2011

Urban Information Modeling combining BIM and GIS, C. Mignard and C. Nicolle, In Mehdi Khosrow-Pour (ed) Encyclopedia of Information Science and Technology (3rd edition), PA: IGI Publishing, Août 2012

Conférences Internationales

SIGA3D: A Semantic BIM Extension to Represent Urban Environnement, Clément Mignard, Gilles Gesquière, Christophe Nicolle, Fifth International Conference on Advances in Semantic Processing, SEMAPRO 2011, November 20-25, 2011, Lisbon, Portugal

SIGA3D: Urban Information Modelling, Combining IFC and GIS. Clément Mignard, Christophe Nicolle, 12th International Conference on Design & Decision Support Systems in Architecture and Urban Planning, August 27-29, 2012, Eindhoven, Pays-Bas

Poster dans une conférence Internationale

Interoperability between GIS and BIM, Clément Mignard, Gilles Gesquière, Christophe Nicolle, International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, KMIS 2011, October 26-29, 2011, Paris, France

Invitation à un colloque national

Modélisation, échange et visualisation d'objets 3D du bâtiment et d'objets urbains géo-référencés ; Application à la Gestion Technique de Patrimoines Immobiliers et Urbains, Clément MIGNARD, Les Halles géomatiques de Rhône-Alpes, 2009