

N° d'ordre :

École Doctorale Mathématiques, Sciences de l'Information et de  
l'Ingénieur

---

UdS – INSA – ENGEES

**THÈSE**

Présentée pour obtenir le grade de

**Docteur de l'Université de Strasbourg**

**Discipline : Sciences de l'Ingénieur**

**Spécialité : Géomatique - Informatique**

Par

**Chokri KOUSSA**

**Implantation d'un système d'information géographique 3D sur  
Internet pour la gestion des modèles urbains**

Soutenue publiquement le 28 Janvier 2011

**Membres du jury**

Président : **M. Robert LAURINI**, Professeur des universités, INSA de Lyon

Rapporteur : **Mme. Anne RUAS**, Directrice de recherche HDR, IGN

Examineur : **M. Gilles HALIN**, Maître de conférences HDR, ENSA de Nancy

Examineur : **M. Mathieu KOEHL**, Maître de conférences, INSA de Strasbourg

Directeur de thèse : **M. Pierre GRUSSENMEYER**, Professeur des universités, INSA de Strasbourg



**REMERCIEMENTS**

*Je tiens à remercier, tout d'abord, la "Région Alsace" d'avoir assuré le financement de cette thèse de doctorat.*

*Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre durant ces trois années de thèse au sein de l'équipe LSIT/TRIO, en particulier le groupe PAGE.*

*Monsieur Pierre GRUSSENMEYER, le directeur de thèse, pour son accueil, ses conseils et la confiance qu'il m'a accordé dès mon arrivée dans son équipe. Je le remercie également pour sa collaboration inestimable, son soutien et pour toutes les discussions instructives et fructueuses que nous avons eu durant toute la durée de mon travail de recherche.*

*Monsieur Mathieu KOEHL, le co-encadrant de la thèse, pour avoir su me guider avec attention et gentillesse pendant ces trois années, ses qualités scientifiques et humaines, son encouragement, sa disponibilité, ses nombreux conseils avisés, et ses remarques ont largement contribué à l'aboutissement de cette thèse. Je lui adresse, par ces quelques mots, ma plus profonde gratitude.*

*Tous les membres du groupe PAGE de l'équipe LSIT/TRIO qui m'ont aidé, de près ou de loin, à mener à bien mes travaux de thèse, en particulier Bernard et Samuel ainsi que mes collègues les doctorants Ibrahim, Hakim, Mostafa et Majd.*

*Je dédie cette thèse à mes chers parents qui tiennent une place immense dans mon cœur. Papa, tu es une vraie école de la vie, je ne cesse d'apprendre tous les jours avec toi. Maman, une femme aussi adorable que toi je n'en connais pas, tu as toujours été là pour moi, et à aucun moment tu n'as cessé de me couvrir de ta tendresse.*

*Parce qu'une thèse ne se fait pas seulement par le travail qui suit, je voudrais remercier mes frères et sœurs, mes beaux frères et belles sœurs, mes neveux et nièces pour le soutien et la confiance qu'ils m'ont toujours accordés. Mes amis, ceux de loin comme ceux de tout près, occuperont toujours une place privilégiée.*

*Je dédie tout particulièrement cette thèse à la plus belle femme de ma vie pour son soutien et son encouragement constant et surtout pour sa patience et sa tendresse.*

*Dans cette épreuve où seuls le courage et la maîtrise des connaissances ne suffisent pas, la force qui donne l'impulsion ne peut provenir que de Dieu. Comme toute production scientifique exécutée "pour la gloire", cette entreprise est le fait de Dieu.*



**RESUME**

Grâce à des facteurs multiples (progrès continu des outils informatiques en termes de logiciels et matériel, puissance de plus en plus remarquable des outils d'acquisition des données spatiales, généralisation des applications et données spatiales pour un usage de plus en plus étendu, etc.), on voit naître des applications spatiales de plus en plus sophistiquées que ce soit aux niveaux des outils libres ou commerciaux. Les Systèmes d'Information Géographiques (SIG) se révèlent être des « consommateurs » incontournables des données géo-référencées. Ils constituent des outils très adéquats, voire les plus appropriés, pour l'exploitation de tel type de données. En effet, ils proposent des fonctionnalités diverses qui se focalisent principalement autour de la gestion, l'interrogation et la représentation des données spatiales.

Les SIG ont commencé à se généraliser réellement pendant les années 1990. Cependant à l'époque, vu les limites des outils informatiques, les SIG s'intéressaient principalement à des données spatiales 2D. Vers la fin des années 1990 et pendant la première décennie du 21<sup>ème</sup> siècle, les développeurs des SIG se sont orientés vers la 3D pour la représentation des données spatiales dans le SIG. Le concept de SIG 3D est ainsi devenu omniprésent et un axe de recherche très répandu dans la communauté scientifique.

Comme le progrès des SIG est lié de très près au progrès des technologies informatiques, Internet est devenu rapidement un moyen très prometteur pour la diffusion en ligne des applications SIG. De ce fait, les SIG 2D sont aujourd'hui nombreux sur Internet. Les SIG 3D restent par contre encore assez rares. C'est dans ce contexte que s'inscrivent nos travaux de thèse. En effet, l'objectif principal consiste à développer le prototype d'un SIG 3D accessible via Internet.

Les SIG étant des outils multidisciplinaires faisant appel à des concepts et technologies très variés (modélisation des données, bases de données, outils de développement, 3D, etc.), nos travaux de thèse vont porter essentiellement sur les concepts basiques des SIG 3D : la modélisation tridimensionnelle des données spatiales, la modélisation des Bases de Données Spatiales (BDS, l'intégration des données spatiales dans la BDS, etc.), mener une réflexion sur les fonctionnalités d'interrogation des données spatiales, le développement d'une application Web permettant d'accéder en ligne à tous les services proposés par le SIG.

Quant aux technologies à utiliser pour le développement du SIG, nos choix se sont orientés principalement vers les outils libres. L'objectif consiste à étudier les différentes technologies mise en œuvre ainsi que leur combinaison pour la mise en place d'un SIG 3D fonctionnel et accessible via Internet.

**Mots clés** : Données spatiales, SIG, 3D, modélisation, représentation des données spatiales, analyse spatiale, Bases de Données Spatiales, Internet.

**ABSTRACT**

Due to a lot of reasons, *i.e.* continuous progress of computer tools in terms of software and hardware, the development of powerful spatial data acquisition tools, the generalization of spatial data and applications for an extended use, etc. a lot of spatial applications, more and more sophisticated, are carried out either as free or commercial tools. The Geographic Information Systems (GIS) are proving to be major consumers of geo-referenced data. They consist in adequate, or even the most appropriate, tools for using such kind of data. Indeed, they offer various features that focus especially on the management, interrogation and representation of spatial data.

The real beginning of GIS generalization was in the 1990s. But then, given the limits of computer tools, GIS were mainly interested in 2D spatial data. While at the late 1990s and during the first decade of the 21st century, GIS developers are oriented to the 3rd dimension for spatial data representation. The concept of 3D GIS then became ubiquitous and a widespread research topic around the world.

As the progress of GIS is related mainly to advances in computer technology, Internet has become fastly a very promising way for the online dissemination of GIS applications. Thus, the 2D GIS are now multiple over Internet, while 3D GIS are rare. It is in this context that joins our thesis work. Indeed, the main objective is to develop a 3D GIS prototype accessible via Internet.

Because GIS are multidisciplinary tools since they rely on very different concepts and technologies (data modeling, databases, development tools, 3D, etc.), Our thesis work will focus on the basic concepts of 3D GIS, *i.e.* three-dimensional spatial data modeling, spatial databases modeling (BDS, spatial data integration in the BDS), carrying out a reflection on the querying functionality of spatial data, a Web application development to access online all services offered by the GIS, etc.

As for the technologies to be used for the development of GIS, our choices were oriented mainly to the free tools. The objective is to study the various technologies implemented and their combination for the establishment of a functional 3D GIS accessible over Internet.

**Keywords:** Spatial data, GIS, 3D, modeling, spatial data representation, spatial analysis, spatial databases.

## ACRONYMES

Ce tableau indique une liste non exhaustive d'acronymes (ainsi que leurs significations) les plus utilisés dans ce mémoire de thèse.

<b>Acronyme</b>	<b>Signification</b>
<b>2D</b>	2 Dimensions
<b>3D</b>	3 Dimensions
<b>API</b>	Application Programming Interface
<b>BD</b>	Base de Données
<b>BDS</b>	Base de Données Spatiale
<b>CAO</b>	Conception Assistée par Ordinateur
<b>IHM</b>	Interaction Homme Machine
<b>ISO</b>	International Organization for Standardization
<b>OGC</b>	Open Geospatial Consortium
<b>SGBD</b>	Système de Gestion de Bases de Données
<b>SGBDS</b>	Système de Gestion de Bases de Données Spatiales
<b>SI</b>	Système d'Information
<b>SIG</b>	Système d'Information Géographique
<b>SQL</b>	Structured Query Language

## PREAMBULES

### AIDE AU LECTEUR : LES STYLES TYPOGRAPHIQUES UTILISES

Ceci est le style typographique de l'introduction ainsi que du corps de chaque chapitre.

*Ceci est le style typographique utilisé pour représenter la finalité ainsi que le résumé du chapitre.*

« *Ceci est le style typographique utilisé pour représenter une citation d'un auteur donné.* ».

Ceci est le style typographique utilisé pour représenter une formule, un algorithme ou un code informatique.

---

Ceci est le style typographique pour représenter une note de bas de page pour définir un concept évoqué dans le corps du texte.

## TABLE DES MATIERES

Remerciements .....	1
Résumé .....	3
Abstract .....	4
Acronymes.....	5
Préambules .....	6
Table des matières .....	7
Liste des figures.....	13
Liste des tableaux .....	17
<b>Introduction générale</b> .....	19
1. Contexte & Problématique .....	19
1.1. Contexte.....	19
1.2. Problématique.....	20
2. Objectifs .....	21
3. Organisation du mémoire .....	23
<b>CHAPITRE.1 Définition des besoins</b> .....	25
Résumé .....	26
Introduction .....	26
1. Application SIG.....	26
1.1. Besoins et domaines d'application .....	26
1.2. Fonctionnalités SIG.....	27
1.3. Hypothèses .....	28
2. Modélisation .....	29
2.1. Type des données .....	29
2.2. Modélisation des données.....	29
3. Utilisateurs du SIG .....	30
Conclusion.....	30
<b>SECTION.I Etat de l'art</b> .....	31
<b>CHAPITRE.2 La 3D, Concepts &amp; Modélisation</b> .....	31
Résumé .....	32
Introduction .....	32
1. Un peu d'histoire de la 3D.....	33
2. Quelques terminologies .....	35
2.1. Le concept de « <i>modélisation</i> » .....	35
2.2. Le terme " <i>dimension</i> " .....	35
2.3. Les concepts de « <i>2D</i> », « <i>2.5D</i> » et « <i>4D</i> » .....	35

2.4. Réalité virtuelle (RV) .....	36
2.5. Réalité augmentée .....	37
3. Modélisation géométrique de la 3D.....	37
3.1. Définition.....	37
3.2. Modélisation filaire : Exemple Fil de fer.....	38
3.3. Modélisation surfacique .....	39
3.3.1. Modélisation polygonale .....	39
3.3.2. B-Rep (Boundary Representation) .....	40
3.3.3. NURBS (Non Uniform Rational Basic Spline).....	41
3.4. Modélisation volumique.....	42
3.4.1. CSG (Constructive Solid Geometry).....	43
3.4.2. RBSO (Regularized Boolean Set Operation) .....	44
3.4.3. SPR (Spatial Partitioning Representation).....	44
3.4.4. Octrees.....	45
3.5. Modélisation hybride : surfacique & volumique.....	46
3.6. Modélisation paramétrique.....	46
Conclusion.....	47
<b>CHAPITRE.3 L’information spatiale, concepts &amp; modélisation .....</b>	<b>49</b>
Résumé .....	50
Introduction .....	50
1. Définitions .....	50
1.1. Information spatiale.....	50
1.2. Modes de représentation des données spatiales.....	51
2. Modélisation géométrique des données spatiales (2D et 3D).....	51
2.1. Normes Internationales.....	52
2.1.1. ISO ( <i>International Organization for Standardization</i> ).....	52
2.1.2. GML (Geography Markup Language).....	53
2.1.3. IFC (Industry Foundation Classes).....	54
2.2. Modèles d’échange et de représentation des données spatiales.....	54
2.2.1. CityGML (City Geography Markup Language).....	54
2.2.2. COLLADA (Collaborative Design Activity) .....	56
2.2.3. KML (Keyhole Markup Language).....	57
2.3. Modèles issus de la recherche .....	58
2.3.1. 3D FDS (Formal Data Structure).....	58
2.3.2. TEN (TEtrahedral Network).....	59
2.3.3. SSM (Simplified Spatial Model) .....	60
2.3.4. UDM (Urban Data Model) .....	60

2.4. Bilan sur la modélisation géométrique .....	61
3. Modélisation topologiques de données spatiales (2D et 3D).....	61
3.1. Définitions .....	61
3.2. Modèles et opérateurs topologiques .....	62
3.3. Matrice des neufs intersections (9I).....	65
3.4. La topologie dans les outils spatiaux .....	70
Conclusion.....	73
<b>CHAPITRE.4 Système d'Information Géographique (SIG).....</b>	<b>75</b>
Résumé .....	76
Introduction .....	76
1. Définitions et terminologies .....	76
1.1. SIG (Système d'Information Géographique).....	76
1.2. SIGM (Système d'Information Géographique Multimédia) .....	79
2. Base de données spatiale (BDS).....	79
3. SGBD spatiaux .....	82
4. SIG et l'open source .....	84
5. SIG et les outils commerciaux.....	86
6. SIG et Internet .....	88
Conclusion.....	92
<b>SECTION.II Modélisation &amp; Intégration des données spatiales dans le SIG.....</b>	<b>93</b>
Introduction générale de la section II .....	93
<b>CHAPITRE.5 Modélisation des données spatiales pour le SIG 3D .....</b>	<b>95</b>
Résumé .....	96
Introduction .....	96
1. Modélisation conceptuelle de données .....	96
1.1. Modélisation géométrique .....	97
1.1.1. Liste de critères.....	97
1.1.2. Représentation des objets 3D .....	98
1.1.2.2. Modélisation surfacique.....	99
1.1.2.3. Modélisation volumique .....	101
1.1.2.4. Notre choix : Modélisation hybride .....	102
1.2. Modélisation topologique .....	105
1.2.1. Deux types de topologies.....	105
1.2.2. Choix topologiques.....	105
1.2.3. Modélisation de la topologie .....	106
1.3. Problématique de modélisation attributaire .....	108
2. Modélisation logique .....	108

2.1. Différents types de modélisation logique .....	108
2.2. Comparaison et choix du modèle logique .....	108
2.3. Représentation du modèle logique .....	110
3. Modélisation physique.....	112
3.1. Choix du SGBD.....	112
3.2. Base de données spatiale multi-niveaux.....	113
3.3. Représentation du modèle physique .....	114
3.4. Optimisation .....	115
Conclusion.....	116
<b>CHAPITRE.6 Intégration &amp; création des jeux de données spatiales .....</b>	<b>117</b>
Résumé .....	118
Introduction .....	118
1. Définitions .....	118
1.1. Formats de données .....	119
1.2. Fichiers CAO.....	119
1.2.1. DXF (Drawing eXchange Format) .....	119
1.2.2. DWG (DraWinG) .....	119
1.2.3. COLLADA (Collaborative Design Activity) .....	119
1.3. Fichiers SIG ( <i>SHP</i> ).....	119
2. Tests et choix des formats .....	120
2.1. Dxf2postgis.....	120
2.2. Shp2pgsql .....	121
2.3. Collada2pgsql.....	122
3. Processus d'intégration.....	125
3.2. Intégration des données géométriques/Topologiques.....	126
3.3. Intégration des données attributaires .....	127
Conclusion.....	128
<b>SECTION.III Mise en place d'un SIG 3D accessible via Internet .....</b>	<b>129</b>
Introduction générale de la section III.....	129
<b>CHAPITRE.7 Représentation des données spatiales dans le SIG 3D.....</b>	<b>131</b>
Résumé .....	132
Introduction .....	132
1. L'organisation des jeux de données.....	132
1.2. Création des couches spatiales .....	133
1.3. Visualisation des couches spatiales .....	136
1.4. Outils IHM (Interaction Homme Machine).....	138
1.4.1. Etape d'analyse .....	139

1.4.2. Etape de conception.....	139
1.5. Les niveaux de détails .....	142
2. Moteur 3D .....	145
3. Comparaison avec les outils commerciaux .....	148
Conclusion.....	149
<b>CHAPITRE.8 Interrogation des données spatiales .....</b>	<b>151</b>
Résumé .....	152
Introduction .....	152
1. Traitement des couches spatiales.....	152
1.1. Opérations sur les couches spatiales.....	153
1.2. Stockage des couches spatiales.....	157
2. Traitement des objets spatiaux .....	159
2.1. Requêtes topologiques.....	159
2.2. Requêtes attributaires .....	165
3. Positionnement du travail par rapport aux outils existants.....	176
Conclusion.....	177
<b>CHAPITRE.9 Diffusion &amp; exploitation des données spatiales sur Internet .....</b>	<b>179</b>
Résumé .....	180
Introduction .....	180
1. Définitions .....	180
1.1. Concepts de SIG .....	180
1.2. Concepts d’Internet .....	181
2. Motivations de la diffusion en ligne du SIG.....	181
3. Objectifs du SIG Web .....	182
4. Architecture de l’application SIG Web .....	183
5. Internet au service du SIG .....	187
6. Problèmes rencontrés.....	189
Conclusion.....	190
<b>SECTION.IV Bilan sur l’application SIG &amp; perspectives.....</b>	<b>193</b>
<b>CHAPITRE.10 Bilan sur l’application SIG &amp; Perspectives.....</b>	<b>193</b>
Résumé .....	194
Introduction .....	194
1. Bilan sur l’application SIG .....	194
1.1. La modélisation conceptuelle du SIG Web .....	194
1.1.1. Modélisation des données spatiales .....	194
1.1.2. Modélisation de la Base de Données Spatiale (BDS).....	195
1.1.3. Modélisation des traitements .....	195

1.2. Evaluation technique du SIG Web .....	195
1.2.1. Le code .....	196
1.2.2. L'application SIG .....	198
1.2.3. L'application Web .....	199
2. Perspectives .....	199
2.1.1. Perspectives techniques/technologiques .....	199
2.1.2. Perspectives fonctionnelles.....	201
2.1.3. Perspectives d'utilisation.....	203
2.1.4. CityGML .....	203
Conclusion.....	204
<b>Conclusion générale.....</b>	<b>207</b>
Références bibliographiques .....	209
Liste des publications .....	217
Annexes .....	219

## LISTE DES FIGURES

<i>Fig2.1.</i> Quelques exemples d'évolution de la 3D au fil du temps .....	34
<i>Fig2.2.</i> Exemples de rendu réaliste(a) et de rendu non réaliste(b) des bâtiments .....	34
<i>Fig2.3.</i> Exemple d'une application de Réalité Virtuelle créé par le Laboratoire de Perception et Contrôle du Mouvement en Environnement Virtuel (LPCM).....	37
<i>Fig2.4.</i> Exemple de modélisation "Fil de Fer" (Eglise canoniale de NIEDERMUNSTER).....	38
<i>Fig2.5.</i> Exemple d'ambiguïté d'interprétation d'un modèle filaire [MARANZANA, 2007] .....	38
<i>Fig2.6.</i> Composants d'une structure polygonale : 1 : un nœud, 2 : un arc et 3 : une face.....	39
<i>Fig2.7.</i> Opération de chanfrein sur un cube : (a) suivant un coin, (b) suivant une arête.....	39
<i>Fig2.8.</i> Exemple d'une représentation B-Rep d'un bâtiment.....	40
<i>Fig2.9.</i> Exemple d'une moto modélisée avec des primitives B-Rep [GEUZAIN, <i>et al</i> , 2009].....	41
<i>Fig2.10.</i> Exemple d'une surface NURBS .....	41
<i>Fig2.11.</i> Exemple d'une représentation CSG d'un bâtiment utilisant un arbre d'opérations booléennes.....	43
<i>Fig2.12.</i> Exemple d'une pièce mécanique modélisée avec des primitives solides CSG [DUBUISSON, 2007]	43
<i>Fig2.13.</i> Utilisation de la méthode SPR pour la construction d'un tore [VAN DAM, <i>et al</i> , 1991].....	45
<i>Fig2.14.</i> Exemple d'un arbre Octree correspondant à des volumes composés de voxels .....	45
<i>Fig2.15.</i> Exemple de modélisation hybride B-Rep/CSG d'un bâtiment .....	46
<i>Fig2.16.</i> Exemple d'une modélisation paramétrique de deux bâtiments .....	47
<i>Fig3.1.</i> Exemple d'un modèle vecteur et du modèle raster correspondant .....	51
<i>Fig3.2.</i> Schémas conceptuel géométrique (à gauche) et topologique (à droite) de l'ISO 19107 .....	52
<i>Fig3.3.</i> Diagramme de classes géométriques de l'ISO 19107.....	53
<i>Fig3.4.</i> Modèle géométrique : primitives complexes et agrégats [OGC, 2008].....	55
<i>Fig3.5.</i> Illustration des différents niveaux de détails de CityGML [OGC, 2008].....	55
<i>Fig3.6.</i> Modèle du bâtiment de CityGML [OGC, 2008].....	56
<i>Fig3.7.</i> Schéma géométrique/topologique [COLLADA, 2008].....	57
<i>Fig3.8.</i> Structure du schéma KML.....	58
<i>Fig3.9.</i> Modèle 3D FDS [MOLENAAR, 1990] .....	59
<i>FIG3.10.</i> Modèle 3D TEN intégré [QINGQUAN, <i>et al</i> , 2000] .....	59
<i>Fig3.11.</i> Simplified Spatial Model [ZLATANOVA, 2000].....	60
<i>Fig3.12.</i> UDM (Urbain Data Model) .....	60
<i>Fig3.13.</i> Modèle topologique générique [ISO, 1999] .....	63
<i>Fig3.14.</i> Modèle topologique relationnel [HOEL, <i>et al</i> , 2003] .....	63
<i>Fig3.15.</i> Structure d'un polyèdre pour des données topologiques [KHUAN, <i>et al</i> , 2008.a] .....	64
<i>Fig3.16.</i> Relation volume-volume [ZLATANOVA, 2000] cité dans [KHUAN, <i>et al</i> , 2008.a].....	64
<i>Fig3.17.</i> Modèle topologique de données [KHUAN, <i>et al</i> , 2008.b] .....	65
<i>Fig3.18.</i> Matrice des 9-Intersections.....	66
<i>Fig3.19.</i> Matrice des 9-Intersection pour des objets n-simplex [WEI, <i>et al</i> , 1998] .....	66

<b>Fig3.20.</b> Diagramme ER du modèle topologique [WEI, <i>et al</i> , 1998] .....	67
<b>Fig3.21.</b> Structure B-Rep modifiée pour supporter le partage des même primitives par plusieurs objets. ....	68
<b>Fig3.22.</b> BB-Rep (Binary B-Rep) : Les tables qui apparaissent en rouge sont celles nécessaires pour la détermination des relations des 9-intersections [ELLUL, <i>et al</i> , 2008].....	69
<b>Fig3.23.</b> MBB-Rep (Modified Binary B-Rep) : Les tables qui apparaissent en rouge sont celles nécessaires pour la détermination des relations des 9-intersections [ELLUL, <i>et al</i> , 2008].....	69
<b>Fig3.24.</b> Diagramme UML GeoTen [LACHANCE, <i>et al</i> , 2006] .....	70
<b>Fig3.25.</b> Matrice 9I correspondant à l'opérateur "contained in" de TQuery [LACHANCE, <i>et al</i> , 2006] .....	70
<b>Fig3.26.</b> Relations topologiques traitées par ORACLE .....	71
<b>Fig3.27.</b> Eléments topologique d'ArcGIS et leurs relations .....	72
<b>Fig3.28.</b> Exemple d'une relation "ST_TOUCHES" de PostGIS .....	72
<b>Fig4.1.</b> Exemple de superposition de couches spatiales dans un SIG 3D [DENEGRÉ, <i>et al</i> , 1996].....	77
<b>Fig4.2.</b> Exemple d'une analyse spatiale : Recherche de voisinage et de zones de densité dans .....	78
<b>Fig4.3.</b> Les notions graphiques de l'approche MODUL-R [BEDARD, <i>et al</i> , 1992] .....	80
<b>Fig4.4.</b> Exemple d'utilisation du modèle Geo-ER [HADZILACOS, <i>et al</i> , 1997] .....	80
<b>Fig4.5.</b> Intégration de données thématiques, spatiales et temporelles [BONAN, <i>et al</i> , 2002] .....	81
<b>Fig4.6.</b> Type de données spatiales d'Oracle.....	82
<b>Fig4.7.</b> SDO_Geometry d'Oracle spatial .....	83
<b>Fig4.8.</b> Exemple d'une application SIG réalisée avec QGIS .....	85
<b>Fig4.9.</b> Exemple d'un modèle 3D et une analyse de la luminosité avec l'extension 3D Analyst et rendu dans ArcScene [ESRIFRANCE, 2010] .....	86
<b>Fig4.10.</b> Exemple d'une carte MapInfo : Aéroport de Zürich avec routes de vol et objets 3D extrudés .....	87
<b>Fig4.11.</b> Le SIG Web pour les municipalités [MAINSTREET, 2009].....	89
<b>Fig4.12.</b> Quelques applications de l'outil "Global mapper" [GLOBALMAPPER, 2010] .....	90
<b>Fig4.13.</b> Architecture simplifiée d'un SIG Web .....	91
<b>Fig5.1.</b> Un bâtiment représenté en tant qu'une collection de surfaces.....	98
<b>Fig5.2.</b> Un bâtiment représenté en tant qu'une collection de solides.....	99
<b>Fig5.3.</b> Deux représentations des murs d'un bâtiment selon le niveau de détail .....	99
<b>Fig5.4.</b> Modèle conceptuel géométrique.....	104
<b>Fig5.5.</b> Les différentes primitives topologiques simples.....	105
<b>Fig5.6.</b> Exemple de topologie entre deux solides qui se touchent .....	105
<b>Fig5.7.</b> Modèle conceptuel géométrique topologique.....	107
<b>Fig5.8.</b> Schéma illustratif de la représentation d'une table objet-relationnelle.....	110
<b>Fig5.9.</b> Modèle logique de données .....	111
<b>Fig5.10.</b> Base de données spatiale multi-niveaux .....	113
<b>Fig5.11.</b> Schéma illustratif de la BDS.....	114
<b>Fig5.12.</b> Exemple d'implémentation d'une BDS dans PostgreSQL/PostGIS.....	115
<b>Fig6.1.</b> Modèles 2D et 3D pour tester l'outil "dxf2postgis" .....	120

<i>Fig6.2.</i> Modèles 3D pour tester l'outil "shp2pgsql" .....	121
<i>Fig6.3.</i> Modèles 3D pour tester l'outil "collada2pgsql" .....	122
<i>Fig6.4.</i> Processus d'intégration de données spatiale dans le SIG .....	125
<i>FIG6.5.</i> Une capture écran de l'application permettant l'intégration de données.....	126
<i>Fig6.6.</i> Algorithme de création des solides 3D à partir des polygones PostGIS.....	127
<i>Fig6.7.</i> Une capture d'écran de l'application génératrice des solides .....	127
<i>Fig7.1.</i> Exemple d'une géométrie (un bâtiment) et sa représentation en PostGIS (the_geom) en tant qu'une collection de surfaces .....	133
<i>Fig7.2.</i> Interface graphique permettant de créer une couche spatiale .....	134
<i>Fig7.3.</i> Structure de la BDS "Bâtiments" .....	135
<i>Fig7.4.</i> Exemple d'une requête pour créer une couche spatiale représentant les Façades et les Toits des Bâtiments de la BDS "Bâtiments" .....	135
<i>Fig7.5.</i> Exemple d'une requête contenant un prédicat sur une table de la BDS .....	136
<i>Fig7.6.</i> Processus de visualisation d'un modèle 3D.....	137
<i>Fig7.7.</i> Algorithme de téléchargement de données spatiales vers la mémoire tampon.....	138
<i>Fig7.8.</i> Architecture globale de l'application.....	140
<i>Fig7.9.</i> Modèle conceptuel des interfaces graphiques de l'application.....	140
<i>Fig7.10.</i> Exemple d'un modèle d'interaction pour la création d'une couche spatiale .....	141
<i>Fig7.11.</i> Quelques éléments graphiques proposés par les interfaces du SIG .....	141
<i>Fig7.12.</i> Exemple d'une visualisation B_Rep d'un modèle 3D.....	142
<i>Fig7.13.</i> Exemple d'une visualisation filaire d'un modèle 3D de l'église canoniale de NIEDERMUNSTER..	142
<i>Fig7.14.</i> Représentation B-Rep (à gauche) et B-Rep_filaire (à droite) de Châtel-sur-Moselle (galeries de la forteresse) .....	143
<i>Fig7.15.</i> Extrusion des polygones B-Rep pour obtenir des solides CSG.....	143
<i>Fig7.16.</i> Correction des formes invalides .....	144
<i>Fig7.17.</i> L'algorithme d'extrusion des polygones B-Rep pour obtenir des solides CSG .....	144
<i>Fig7.18.</i> Exemple d'une visualisation B_Rep_CSG d'un modèle 3D .....	145
<i>Fig7.19.</i> Exemple d'objets 3D de différents types rendus par le moteur 3D .....	146
<i>Fig7.20.</i> Exemple d'un objet représenté selon trois différents niveaux de transparence .....	147
<i>Fig7.21.</i> Fonctionnement du moteur 3D .....	148
<i>Fig8.1.</i> Schéma illustratif des opérations booléennes : union, intersection et différence entre deux couches spatiales .....	153
<i>Fig8.2.</i> Organigramme d'une opération booléenne entre deux requêtes SQL correspondant à deux couches spatiales .....	154
<i>Fig8.3.</i> Représentation graphique d'une opération booléenne entre deux couches spatiales.....	154
<i>Fig8.4.</i> Représentation graphique d'une opération booléenne entre plusieurs couches spatiales .....	154
<i>Fig8.5.</i> Quelques exemples de symboles utilisés pour dessiner une opération booléenne.....	155
<i>Fig8.6.</i> Algorithme de calcul d'une opération booléenne entre deux couches spatiales.....	156
<i>Fig8.7.</i> Exemple d'une requête d'union entre trois couches spatiales : Façades, Toits et MNT .....	157

<i>Fig8.8.</i> Méta-Modèle d'une couche spatiale .....	158
<i>Fig8.9.</i> Modèle logique relationnel d'une couche spatiale.....	159
<i>Fig8.10.</i> Schéma illustratif d'un solide qui contient un autre .....	162
<i>Fig8.11.</i> Méta-modèle d'une opération topologique.....	163
<i>Fig8.12.</i> Organigramme de l'algorithme topologique.....	163
<i>Fig8.13.</i> Schéma illustratif de la méthode permettant de vérifier si un point est l'intérieur d'un solide .....	164
<i>Fig8.14.</i> Algorithme Test_point_dans_solide .....	165
<i>Fig8.15.</i> Exemple d'une opération topologique sur deux bâtiments 3D .....	165
<i>Fig8.16.</i> Affichage graphique du modèle conceptuel d'une BDS (exemple de la BDS "Bâtiments").....	166
<i>Fig8.17.</i> Exemple d'un affichage de contenu et des dépendances de la table "Bâtiment" .....	167
<i>Fig8.18.</i> Affichage d'informations attributaires concernant un objet spatial sélectionné par l'utilisateur ...	168
<i>Fig8.19.</i> L'interface permettant de créer une requête pour chercher des objets spatiaux .....	169
<i>Fig8.20.</i> L'outil graphique permettant de créer un opérateur géométrique.....	170
<i>Fig8.21.</i> Schéma illustratif de la traduction d'un schéma de base de données en graphe.....	171
<i>Fig8.22.</i> Exemple illustratif du calcul du plus court chemin .....	172
<i>Fig8.23.</i> Exemple d'une requête portant sur des propriétés géométriques, topologiques et attributaires .....	173
<i>Fig8.24.</i> Le résultat de la requête décrite dans la figure ( <i>Fig8.23</i> ) .....	173
<i>Fig8.25.</i> Exemple d'une requête attributaires permettant de retourner un résultat sous forme de solides (droite) et sous forme de surfaces (gauche).....	174
<i>Fig8.26.</i> Exemples de quelques enregistrements des tables (a) : "batiment", (b) : "type_chauffage" .....	175
<i>Fig8.27.</i> Interface graphique permettant de représenter la BDS selon son contenu.....	175
<i>Fig8.28.</i> Résultat de la requête représentée dans la figure ( <i>Fig8.27</i> ).....	176
<i>Fig9.1.</i> Architecture simplifiée du SIG Web .....	186
<i>Fig9.2.</i> Diagramme de séquences de la communication entre un client Web et le serveur Web dans le cadre du SIG Web .....	187
<i>Fig9.3.</i> Mise en ligne d'une couche spatiale et les différentes alternatives de son partage .....	188
<i>Fig9.4.</i> Interface graphique pour la visualisation d'une couche spatiale stockée dans la BDS .....	188
<i>Fig9.5.</i> Une capture d'écran de l'application SIG Web .....	189
<i>Fig9.6.</i> Exemple d'un fichier JNLP utilisé dans le cadre du SIG Web.....	190
<i>Fig10.1.</i> Statistiques du code Java du SIG Web .....	196
<i>Fig10.2.</i> Dépendances entre les packages du projet SIG Web.....	197
<i>Fig10.3.</i> Courbes décrivant la performance de l'application SIG Web .....	197
<i>Fig10.4.</i> La configuration de la machine sur laquelle nous avons développé le SIG Web .....	198

## LISTE DES TABLEAUX

<b>Tab2.1</b> De la 2D à la 3D [GERBE, 2002].....	36
<b>Tab2.2</b> Comparaison des résultats d'une opération d'intersection entre des opérations booléennes ordinaires et celles régularisées.....	44
<b>Tab3.1</b> Primitives géométrique et les primitives topologiques associées .....	62
<b>Tab3.2</b> Différentes conditions d'opérations topologiques [KHUAN, <i>et al</i> , 2008.a].....	65
<b>Tab3.3</b> La matrice des 9-intersection appliquée sur deux polygones 2D [CODEHAUS, 2009] .....	66
<b>Tab3.4</b> Table Relations topologiques 3D de la matrice 9-intersections [ZLATANOVA, 2000].....	67
<b>Tab4.1</b> Comparaison entre les SGBD spatiaux Oracle, PostgreSQL/PostGIS et MySQL.....	84
<b>Tab4.2</b> Caractéristiques techniques de quelques SIG libres.....	85
<b>Tab4.3</b> Comparaison des principales solutions SIG.....	88
<b>Tab4.4</b> Un petit comparatif entre des SIG Web .....	91
<b>Tab5.1</b> Comparaison des méthodes de modélisation surfacique B-Rep et NURBS .....	100
<b>Tab5.2</b> Comparaison des méthodes de modélisation volumique CSG, SPR et Octree .....	101
<b>Tab5.3</b> Comparaison du stockage de données pour les bâtiments des <i>Figures 5.1</i> et <i>5.2</i> .....	104
<b>Tab5.4</b> Comparaison entre différents types de modèle logiques.....	109
<b>Tab5.5</b> Comparaison de quelques SGBD spatiaux.....	113
<b>Tab6.1</b> Résultats d'exportation des fichiers DXF vers PostGIS .....	121
<b>Tab6.2</b> Résultats d'exportation des fichiers SHP vers PostGIS .....	122
<b>Tab6.3</b> Résultats d'exportation des fichiers COLLADA vers PostGIS .....	124
<b>Tab6.4</b> Comparaison des outils d'intégration des fichiers <i>SHP</i> , <i>DXF</i> et <i>COLLADA</i> .....	124
<b>Tab7.1</b> Les résultats d'extrusion de deux surfaces interconnectées selon l'angle d'intersection.....	143
<b>TAB8.1</b> Différents types de relations topologiques entres deux solides 3D .....	160
<b>Tab8.2</b> Tableau des relations topologiques entre des objets 3D .....	161
<b>Tab10.1</b> Aperçu de quelques fonctionnalités qui existent ou non dans le SIG Web.....	198
<b>Tab10.2</b> Tableau de comparaison entre les fichiers sources de notre SIG et CityGML .....	204



## INTRODUCTION GENERALE

On remarque, de plus en plus, un intérêt grandissant pour les systèmes d'information géographique (SIG) qui se révèlent être de puissants outils d'analyse et d'aide à la décision concernant des phénomènes relatifs à une localisation dans l'espace. L'importance des SIG réside dans leurs capacités avancées en termes de saisie, de stockage, de traitement et de représentation des données spatialement référencées. Les premiers SIG s'intéressaient aux données spatiales bidimensionnelles. De ce fait, les SIG 2D connaissent aujourd'hui une maturité relativement avancée. Des nouveaux besoins sont apparus depuis notamment en ce qui concerne le traitement de données 3D et la diffusion de ces mêmes données tridimensionnelles à travers Internet. Au delà des données, les utilisateurs souhaitent également disposer de fonctionnalités avancées permettant d'effectuer des traitements SIG à distance et notamment sur des plateformes fondées sur Internet. Dans ce cadre, l'objectif de cette thèse consiste à mettre en place un prototype de SIG 3D accessible via Internet permettant une exploitation tridimensionnelle, entres autres, de modèles urbains (principalement les extérieurs des bâtiments représentés par des surfaces).

### 1. CONTEXTE & PROBLEMATIQUE

#### 1.1. CONTEXTE

Les outils spatiaux, en particulier les SIG, constituent un domaine multidisciplinaire (sciences de l'informatique, sciences spatiales, modélisation, etc.). Ils font intervenir des concepts et des technologies très variés. De tels concepts et technologies peuvent être intégrés dans un même système pour des objectifs communs. Nous évoquons, dans cette partie, l'ensemble de concepts, que nous jugeons pertinents, pour expliquer le contexte général de nos travaux de thèse. Les concepts évoqués ci-dessous ne sont pas exhaustifs, d'une part, et ne peuvent pas être expliqués en détail dans cette introduction générale, d'autre part.

- **3D (3 Dimensions)**

La 3D, de plus en plus utilisée par les communautés d'utilisateurs, permet de représenter des objets, réels ou virtuels, en termes de largeur, hauteur et profondeur. Ce concept est souvent associé aux sciences de l'ordinateur. En effet, les outils informatiques jouent un rôle extrêmement important quant à la création et à l'exploitation d'applications 3D grâce aux différentes techniques telles que l'utilisation des matériaux, textures, luminosité, algorithmes de rendu 3D, effets spéciaux, animations, etc.

Au delà des spécificités "techniques" parfois compliquées à expliquer, la 3D a envahi la quasi-totalité des domaines de notre vie quotidienne. On parle de la 3D dans les jeux vidéo, au cinéma, dans les géoportails, en CAO (Conception Assistée par Ordinateur) de pièces mécaniques, en imagerie médicale et dans une liste très étendue d'autres domaines d'application. Nous intégrons également cette 3D, pas encore arrivée à maturité, dans le développement des différentes fonctionnalités de notre SIG.

- **Modélisation spatiale**

La modélisation est une problématique centrale dans le domaine des données spatiales. Cela est dû essentiellement à leurs particularités (données complexes, généralement très volumineuses, ayant un format particulier, etc.). La modélisation des données est essentielle au développement des bases de données, en particulier les bases de données spatiales, et des systèmes d'information, en particulier des SIG. Cette modélisation est la première étape dans le cycle de développement d'un SIG. Elle joue un rôle incontournable pour la compréhension, perception, structuration, interprétation et manipulation des données spatiales utilisées pour la création des modèles spatiaux.

Un concepteur de données spatiales possède généralement trois alternatives pour la mise en place d'un modèle de données spatiales : i) Utiliser intégralement un modèle spatial existant ou suivre une démarche définie par une approche de modélisation de données spatiales. ii) Modifier ou adapter un modèle spatial existant. iii) Concevoir son propre modèle en utilisant ses propres techniques de

modélisation. Le choix d'une solution dépend de plusieurs facteurs tels que les besoins, le niveau de détail, les possibilités de stockage, les dimensions, etc. La dimension des données spatiales (2D ou 3D, voire 4D) dépend, à son tour, des types d'objets spatiaux à représenter.

- **SIG**

Le concept de SIG est relativement récent car il n'a commencé à se généraliser vraiment qu'à partir des années 1990. Un SIG est matérialisé essentiellement par l'ensemble de logiciels, matériel et données spatiales et non spatiales. Il s'articule autour des fameux 5A : Acquisition, Archivage, Accès, Analyse et Affichage. Ces 5A montrent nettement la portée des SIG et la richesse des fonctionnalités qu'ils doivent accomplir.

Par rapport aux outils de CAO ou autres outils spatiaux, le SIG présente une multitude d'avancées. En effet, outre le stockage et la gestion des données spatiales, un SIG doit pouvoir représenter les données spatiales pour en faciliter l'interprétation. Un SIG est de ce fait souvent qualifié d'outil puissant d'analyse et de prise de décision.

Un ensemble de concepts doivent être intégrés pour le développement de tout SIG à savoir :

- Données : les SGBDS (Système de Gestion de Bases de Données Spatiales), les BDS (Bases de Données Spatiales), modélisations spatiales, etc.
- Traitements : un visionneur 2D/3D, des techniques d'IHM (Interaction Homme Machine), etc.

- **Internet**

Internet a envahi tous les domaines de notre vie quotidienne. On parle aujourd'hui du commerce électronique, d'accès et de gestion des comptes bancaires sur Internet, des jeux vidéo sur Internet, des encyclopédies électroniques, des bibliothèques en ligne, des géo-portails, etc.

La portée d'Internet n'est plus limitée à des domaines particuliers ou à des personnes particulières. C'est un réseau ouvert à tout le monde pour différents types d'applications. Pour ce faire, plusieurs technologies sont fondées sur le réseau Internet permettant ainsi de diffuser en ligne des types très variés d'applications. En faisant le lien entre le concept d'Internet et les concepts vus ci-dessus, il est possible aujourd'hui d'accéder à un SIG 3D sur Internet.

## 1.2. PROBLEMATIQUE

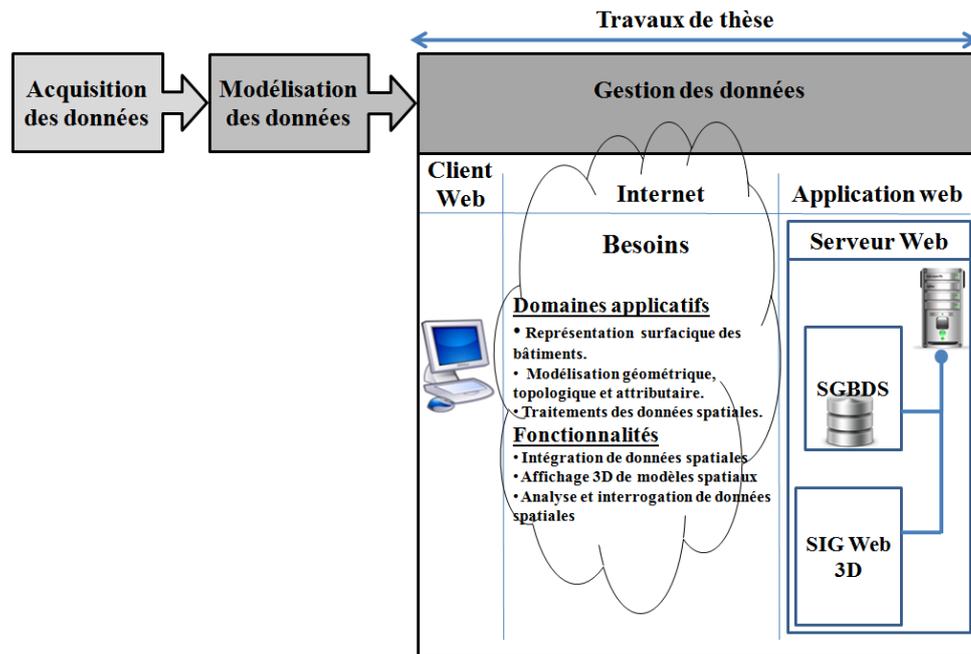
Cette thèse est réalisée au sein du groupe Photogrammétrie et Géomatique (PAGE) du laboratoire LSIT UMR 7005, équipe TRIO.

Les travaux de l'équipe PAGE portent sur les trois domaines suivants :

- L'acquisition des données spatiales.
- La modélisation de ces données spatiales acquises à travers un ensemble de modèles (2D et 3D) structurés.
- La gestion des données spatiales pour l'exploitation optimisée de ces modèles.

Les travaux de cette thèse s'inscrivent dans le troisième domaine. Cette thèse a pour problématique de base le développement d'un SIG 3D pour la gestion, l'analyse et la représentation des données géo-référencées acquises et modélisées dans les deux premières thématiques. Cette gestion en aval est très importante pour l'exploitation des données ainsi que pour la valorisation des modèles.

Malgré l'existence d'un nombre important d'outils (SIG et non SIG) libres, nous avons tenté d'offrir un prototype de SIG traduisant une réflexion sur les fonctionnalités essentielles des SIG 3D où l'utilisateur serait guidé dans ses choix mais resterait libre dans ses décisions. Le non recours aux outils existants pour le développement du SIG 3D est justifié par deux raisons: i) Le développement d'un nouveau prototype offre plus de flexibilité et de liberté pour la mise en place des différentes fonctionnalités attendues du SIG. ii) Le SIG va constituer un prototype sur lequel pourront se greffer des travaux futurs au sein de l'équipe LSIT-TRIO. La figure (*Fig1*) résume sous la forme d'un schéma simplifié, la problématique de la thèse.



*Fig 1.* Schéma simplifié illustrant la problématique de la thèse située par rapport aux autres travaux de l'équipe PAGE

Les SIG 2D, locaux ou en ligne, ont atteint un niveau de maturité relativement élevé. En particulier sur Internet, il existe une multitude d'applications SIG 2D proposant une bibliothèque très riche de services s'appliquant dans des domaines assez diversifiés tels que la gestion territoriale, l'urbanisme, l'optimisation des itinéraires, etc. Cependant, les SIG 3D ne sont pas encore très présents sur le net. A l'exception de quelques SIG Web, la 3D se limite essentiellement à la visualisation d'objets 3D avec des fonctions limitées d'analyse. C'est dans ce cadre que s'inscrivent nos travaux de thèse qui portent sur la proposition d'un SIG 3D accessible sur Internet pour la gestion des modèles 3D, en particulier des modèles urbains.

## 2. OBJECTIFS

A l'issue du contexte et de la problématique expliqués ci-dessus, nous définissons les objectifs majeurs des travaux menés dans le cadre de cette thèse. En effet, nous organisons ces objectifs dans les différentes rubriques expliquées ci-dessous.

### • Utilisateurs du SIG

Comme toute application informatique, en particulier toute application SIG, il est primordial de définir ses utilisateurs finaux ou potentiels. L'application SIG développée dans le cadre de cette thèse est destinée à tout utilisateur désirant :

- La visualisation des objets spatiaux dans des milieux urbains que ce soit des bâtiments isolés, en particulier des bâtiments ou sites historiques, ou un ensemble de bâtiments,
- La création aisée des jeux de données spatiales pour des fins d'analyse et d'interrogation,
- La gestion de ces données dans une Base de Données Spatiales.

Les utilisateurs peuvent changer selon les besoins attendus du SIG, selon les données (des bâtiments moins détaillés pour des urbanistes et des bâtiments très détaillés pour des architectes, par exemple), etc.

- **Modélisation des données spatiales**

Pour suivre la démarche habituelle de conception des systèmes d'information, la modélisation spatiale constitue l'objet de nos premiers travaux en matière de réflexion sur le SIG.

En effet, la modélisation doit poursuivre les objectifs suivants :

- Les objets spatiaux, en particulier les bâtiments, doivent être représentés à travers des modèles tridimensionnels. La représentation 3D doit également pouvoir représenter les objets avec différents niveaux de détails.
- La modélisation doit porter sur les propriétés géométriques, topologiques et attributaires (dites aussi thématiques ou sémantiques) des objets spatiaux.
- Pour mettre en place le modèle spatial 3D, il convient de se référer aux divers travaux entrepris dans ce domaine pour pouvoir ensuite proposer une modélisation sur cette base.
- Un des objectifs majeurs de nos travaux est de pouvoir représenter des modèles simples de bâtiments à l'aide de solides 3D et de les stocker dans un SGBDS.
- La modélisation 3D d'objets spatiaux doit également tenir compte des spécificités d'un SGBDS particulier, notamment en termes de capacités de stockage et manipulation d'objets 3D.
- Enfin, il est très important que la modélisation respecte les normes.

Sur la base des choix pris dans le cadre de la modélisation, un modèle spatial 3D doit être défini en suivant une démarche de modélisation d'objets spatiaux du SIG.

- **Représentation des données spatiales**

Parmi les objectifs principaux de notre SIG 3D, nous considérons que la représentation d'objets spatiaux ainsi que les résultats des requêtes spatiales sur les objets facilitent leur compréhension, interprétation, analyse, etc. Un système de couches va être mis en place pour représenter les différents modèles 3D à travers un ensemble de couches spatiales superposées. La représentation doit être capable d'afficher tout objet 3D pouvant être modélisé par des polygones, et en particulier des bâtiments.

Les outils 3D actuels offrent des possibilités avancées en termes d'affichage d'objets 3D. Un moteur 3D doit être alors mis en place utilisant des technologies adaptées aux besoins attendus du SIG (gestion, analyse et représentation des données spatiales). Le moteur 3D doit offrir des fonctionnalités souples en termes de navigation 3D ainsi que de manipulation d'objets 3D.

- **Fonctionnalités du SIG Web**

Pour la définition des différentes fonctionnalités du SIG, nous suivons une approche ascendante. En effet, de telles fonctionnalités ne peuvent pas être définies exhaustivement *a priori*. Elles sont plutôt définies au fur et à mesure de l'avancement des travaux de thèse. Cependant, nous commençons par la définition des fonctionnalités essentielles.

Comme la représentation des données se base sur un système de couches, il est très important que le SIG fournisse des méthodes pour la fusion des couches spatiales (union, différence, intersection, etc.) pour déduire d'autres couches spatiales plus « sophistiquées ».

- L'interrogation topologique constitue une des plus importantes fonctionnalités d'un SIG. Elle consiste essentiellement à identifier les relations de voisinage entre des objets spatiaux. Comme ces derniers sont tridimensionnels, il faut mettre en place une méthode d'interrogation topologique 3D à travers Internet.
- Outre les données géométriques et topologiques, un SIG contient des données attributaires pour la description sémantique des objets spatiaux. L'interrogation spatiale peut également comporter des propriétés attributaires pour identifier des objets spatiaux dans les modèles 3D.
- Le système d'interrogation doit pouvoir porter, dans la même requête, sur des propriétés géométriques, topologiques et attributaires.

- Pour faciliter l'interrogation des modèles 3D, un langage graphique peut être très utile. En effet, au lieu de composer manuellement des requêtes complexes (pouvant être une source d'erreur), on peut mettre en place un langage graphique facile à percevoir et à utiliser.
- Des procédures d'analyse spatiale doivent être aussi mises en place. Il s'agit d'utiliser des graphes et des tableaux personnalisés par les utilisateurs.

Ces fonctionnalités peuvent être étendues en améliorant les fonctionnalités déjà fournies dans les outils spatiaux existants ou en proposant des nouvelles fonctionnalités.

- **Exploitation du SIG Web**

A l'ère des NTIC (Nouvelles Technologies de l'Information et de la Communication), l'information est de plus en plus accessible. Internet a révolutionné les outils de communication et est devenu un moyen essentiel pour l'échange et le partage de données à travers le monde. Tout d'abord réservées pour des applications locales et des utilisateurs spécifiques, les données spatiales sont maintenant généralisées pour des géo-applications complexes et disponibles en ligne pour un nombre illimité d'utilisateurs.

Ainsi, les objectifs de l'application SIG Web peuvent-ils être résumés dans les points suivants :

- L'application doit pouvoir autoriser l'accès à distance aux différentes fonctionnalités du SIG. L'utilisation des différentes applications du SIG Web doit ressembler à une application locale.
- La manipulation des données spatiales peut être matérialisée par un travail individuel ou collectif, plusieurs utilisateurs doivent pouvoir collaborer pour exécuter des tâches communes sur des mêmes jeux de données.
- Chaque utilisateur doit posséder un compte pour manipuler à distance des données spatiales stockées sur un serveur Web. Cela permet à un utilisateur de stocker ses lots de données spatiales ainsi que les résultats d'analyse pour une réutilisation future ou pour les partager avec d'autres utilisateurs.

### 3. ORGANISATION DU MEMOIRE

Le présent rapport de thèse est organisé à travers quatre sections principales : la première met l'accent sur l'état de l'art, la deuxième section, quant à elle, s'intéresse à la modélisation et l'intégration des données spatiales dans le SIG, alors que la troisième section décrit le développement du SIG Web 3D, et enfin, la quatrième section constitue un bilan sur le travail mené dans le cadre de cette thèse et propose des perspectives complémentaires.

Avant d'aborder les différentes sections mentionnées ci-dessus, nous proposons, tout d'abord, un chapitre intitulé "*Définition des besoins*". Dans ce chapitre nous expliquons les besoins attendus du SIG en termes de domaines d'application, de modélisation et enfin d'utilisateurs.

Dans la section "*Etat de l'art*", un intérêt particulier est accordé au contexte général du travail en termes des concepts qui touchent au sujet de thèse. En effet, cette section est composée de trois chapitres. Le premier, intitulé "*La 3D, Concepts & Modélisation*", aborde le concept de la 3D en termes d'histoire, de définitions ainsi que d'approches de modélisation. Le deuxième chapitre, qui se nomme "*L'information Spatiale, Concepts & Modélisation*", s'intéresse au concept de données spatiales en évoquant quelques définitions ainsi qu'un aperçu des principales approches de modélisation géométriques et topologiques des données spatiales. Le troisième chapitre, qui s'intitule "*Systèmes d'Information Géographique*" évoque le sujet des SIG par rapport à un ensemble de définitions et de terminologie ainsi que d'explications des principaux concepts relatifs.

A partir de la section "*Modélisation & intégration de données spatiales dans le SIG 3D*", nous commençons à expliquer nos travaux menés dans le cadre de cette thèse. Cette section est composée de deux chapitres. Le premier, intitulé "*Modélisation des données spatiales 3D*", décrit la procédure de modélisation 3D des données spatiale en s'appuyant sur les trois niveaux de conception de bases de données: conceptuel, logique et physique. Alors que le deuxième chapitre, qui se nomme "*Intégration*"

*& Création des jeux de données spatiales*", décrit la démarche d'intégration de données provenant des fichiers SIG ainsi que des fichiers CAO dans la BDS.

Dans la section "*Mise en place d'un SIG 3D accessible via Internet*", nous nous lançons dans la description de la procédure de développement du SIG Web. Cette section est composée, à son tour, de trois chapitres. Le premier, intitulé "*Présentation des données spatiales dans le SIG 3D*", explique comment les données spatiales sont organisées à travers le SIG pour être exploitées par les utilisateurs. Le deuxième chapitre, qui se nomme "*Interrogation de données spatiales*", présente nos propositions en termes d'interrogation topologique et attributaire des modèles 3D. Alors que le troisième chapitre, intitulé "*Diffusion et exploitation de données sur Internet*", explique comment les diverses fonctionnalités du SIG sont diffusées en ligne sur Internet.

La dernière section s'intitule "*Bilan sur l'application SIG & Perspectives*". Elle constitue un bilan de l'application SIG développée dans le cadre de cette thèse en termes de résultats, de modélisation, d'implication technique et technologique, etc. Sur la base de ce bilan, nous recensons quelques perspectives pour l'amélioration et l'enrichissement des fonctionnalités du SIG Web.

---

## CHAPITRE.1 DEFINITION DES BESOINS

Résumé.....	26
Introduction.....	26
<b>1. Application SIG</b> .....	26
1.1. Besoins et domaines d'application.....	26
1.2. Fonctionnalités SIG.....	27
1.3. Hypothèses.....	28
<b>2. Modélisation</b> .....	29
2.1. Type des données.....	29
2.2. Modélisation des données.....	29
<b>3. Utilisateurs du SIG</b> .....	30
Conclusion.....	30

## RESUME

*Avant de rentrer en détail dans l'explication des différents travaux menés dans le cadre de thèse, il est nécessaire d'introduire les besoins attendus du prototype du SIG. Cela constitue l'objectif de ce chapitre. Pour ce faire, nous évoquons les fonctionnalités du SIG, les données exploitées par ce SIG et enfin ses utilisateurs potentiels.*

## INTRODUCTION

Avant la mise en place de toute application informatique, en particulier toute application SIG, les besoins doivent être bien délimités. Cela peut apporter une valeur ajoutée en matière de définition des capacités du SIG ainsi que ses limites. Une application SIG ne peut pas être universelle. Elle est généralement mise en place pour répondre à des besoins particuliers. En effet, l'objectif de ce travail consiste à implanter un SIG pour la gestion et la manipulation des objets spatiaux principalement des bâtiments dans des milieux urbains.

Ce chapitre introduit alors les besoins attendus de l'application SIG qui doit être développée dans le cadre de ce travail de thèse. En effet, nous expliquons respectivement les fondements de l'application SIG en termes de domaines d'application, d'hypothèses initiales et de fonctionnalités à développer, le type et la modélisation des données exploitées par ce SIG et enfin les utilisateurs et les différentes alternatives d'utilisation des fonctionnalités du SIG.

### 1. APPLICATION SIG

De nos jours, une multitude de domaines impliquent des données spatiales, généralement associées à des données thématiques, pour des fins assez diversifiées. Pour ce faire, il existe une large variété de solutions spatiales (SIG et non SIG) libres ou commerciales proposant divers services.

Dans cette partie, nous expliquons les besoins et les domaines dans lesquels peut être utilisé le prototype du SIG ainsi que ses principales fonctionnalités.

#### 1.1. BESOINS ET DOMAINES D'APPLICATION

Comme nous l'avons expliqué dans "*l'introduction générale*", les travaux de l'équipe PAGE sont fondés principalement autour des trois domaines suivants : l'acquisition, la modélisation et la gestion des données géo-localisées. Les travaux de cette thèse s'inscrivent dans le domaine de gestion des données spatiales par le biais d'un prototype de SIG. Nous classifions les principaux besoins attendus de ce prototype selon différentes rubriques.

- **Recherche scientifique**

Malgré l'existence de plusieurs solutions SIG libres sur lesquelles nous pourrions greffer notre propre réflexion dans le domaine du SIG, nous avons favorisé le développement de notre propre prototype pour plusieurs raisons :

- Le développement du SIG est un bon moyen pour appréhender, d'une part, les spécificités techniques d'un logiciel de SIG, et d'autre part, ses spécificités fonctionnelles.
- Mener une réflexion sur les fonctionnalités existantes des SIG et les alimenter par d'autres fonctionnalités traduisant le travail de recherche de cette thèse.
- Le prototype du SIG réalisé dans le cadre de cette thèse constituera un noyau sur lequel pourront être fondés des travaux futurs de l'équipe PAGE.

- **Domaines applicatifs**

Le SIG est généralement créé pour des fins assez précises et pour des domaines d'application bien définis. Cependant, une application SIG peut évoluer au cours du temps pour élargir les domaines d'application et pour tenir compte des nouveaux besoins. Cela constitue l'objectif de notre SIG. En effet, la solution SIG développée dans le cadre de cette thèse a pour objectif la gestion des objets spatiaux principalement de type (enveloppe de) bâtiment dans des milieux urbains. Ces objets spatiaux peuvent être des bâtiments isolés, en particulier des bâtiments historiques, ou un groupe de bâtiments représentant une partie d'une ville. Le but initial ne consiste pas à gérer des grandes villes contenant des milliers de bâtiments, mais plutôt des ensembles contenant quelques centaines de bâtiments pour des projets assez précis. Comme le développement du SIG est réalisé dans le cadre d'un travail de recherche, les domaines d'application peuvent être enrichis au fur et à mesure de la mise en place d'autres projets de l'équipe PAGE.

Concrètement, le SIG développé dans le cadre de cette thèse peut être appliqué pour :

- La visualisation des modèles urbains représentés par une méthode surfacique.
- La mise en place des petits projets nécessitant la gestion des objets de type bâtiment et la création d'un ensemble de requêtes topologiques et attributaires pour interroger les modèles spatiaux.
- Selon le type et la précision des données initiales, le SIG peut être appliqué à des contextes différents comme par exemple l'étude des bâtiments en matière de déperdition de chaleur, d'émission de CO<sub>2</sub>, etc.

Cependant, le SIG ne prend pas en compte :

- La simulation des phénomènes naturels tels que les incendies, les inondations, etc.
- La représentation des objets naturels complexes tels que la végétation, les forêts, etc.
- Le rendu des données rasters.

## **1.2. FONCTIONNALITES SIG**

L'alimentation du SIG par des nouvelles fonctionnalités est subjective. D'une part, elle dépend des besoins attendus de ce SIG, et d'autre part, de telles fonctionnalités ne peuvent pas être définitives car les besoins, les données, les attentes des utilisateurs, etc. sont susceptibles de changer à tout moment. L'objectif de cette thèse consiste à développer un prototype de SIG accessible sur Internet dans lequel nous distinguons deux types de fonctionnalités : fonctionnalités de base et nos contributions.

- **Fonctionnalités de base**

Il s'agit de fonctionnalités qu'on trouve dans la plupart des solutions SIG. Il est primordial, tout d'abord, de développer ces fonctionnalités pour pouvoir y ajouter, par la suite, les autres fonctionnalités traduisant nos contributions. Ces fonctionnalités de base s'inscrivent dans les domaines suivants :

- La gestion des données spatiales : Le stockage des données utiles pour le SIG est une fonctionnalité primordiale pour tout SIG. Ces données sont généralement stockées dans un SGBDS (Système de Gestion de Bases de Données Spatiales). Dans le cadre de ce travail, nous faisons recours au SGBD libre PostgreSQL avec sa couche spatiale PostGIS.
- L'interrogation et l'analyse des données spatiales : C'est la raison d'être de tout SIG et la propriété principale qui le distingue des outils de DAO (Dessin Assisté par Ordinateur). Une

fonctionnalité particulière, développée dans le cadre de ce travail et qu'on trouve dans la plupart des SIG, consiste à combiner deux ou plusieurs couches spatiales pour former d'autres couches spatiales plus sophistiquées.

- La représentation des données spatiales : L'outil SIG permet de transformer les données spatiales stockées dans des fichiers ou dans une BDS en un ensemble de modèles virtuels faciles à percevoir par la vision humaine. Il permet également de représenter les résultats d'analyse en utilisant différentes techniques (couleurs, textures, transparence, etc.). Les modèles spatiaux sont généralement représentés à travers un système de couches superposées.

- **Contributions**

Après avoir développé les fonctionnalités de base, nous pouvons y ajouter de nouvelles fonctionnalités traduisant notre réflexion dans le domaine du SIG. Nos contributions portent principalement sur les problématiques suivantes :

- La gestion des données spatiales : Nos choix portent essentiellement sur des SGBDS libres. Pour ce faire, nous avons adopté le SGBDS PostgreSQL/PostGIS car il propose une multitude de fonctions spatiales. Une BDS est également mise en place pour permettre le stockage des données spatiales et non spatiales. Et comme PostGIS ne permet pas jusqu'à sa version actuelle (1.5) le stockage des solides 3D, nous avons développé une fonction pour la création et la gestion des solides 3D en se basant sur des identifiants uniques.
- L'interrogation et l'analyse des données spatiale : Outre les fonctionnalités de base, nous proposons un ensemble de fonctionnalités traduisant nos travaux de recherche. L'objectif est d'assister les utilisateurs lors de l'interrogation de la BDS et les modèles spatiaux (interrogation géométrique, topologique et attributive). En effet, plusieurs interfaces graphiques doivent être développées pour la création des couches spatiales et l'interrogation attributive des modèles spatiaux. Et pour les requêtes booléennes de combinaison des couches spatiales et les requêtes topologiques, un langage graphique est développé pour faciliter les tâches des utilisateurs.
- La représentation des données spatiales : Un moteur 3D doit être développé pour rendre les modèles spatiaux, d'une part, et les résultats d'interrogation de ces modèles, d'autre part.

Les contributions seront expliquées en détails dans la section intitulée "*Mise en place d'un SIG 3D accessible via Internet*".

### **1.3. HYPOTHESES**

Avant d'entamer la phase du développement du SIG, nous avons fixé quelques hypothèses relatives principalement aux données ainsi que l'application SIG elle-même.

- **Données**

La qualité des données spatiales joue un rôle central tant au niveau de l'évaluation qu'au niveau du test du SIG en matière de capacités et de limites. Nous avons fixé alors quelques hypothèses concernant les données utiles pour notre prototype du SIG.

- Le SIG ne sera pas capable, dans sa version actuelle, d'éditer ou de corriger les données sources en termes de géométrie et de topologie. En effet, il suppose que ces données ont été validées géométriquement avant leur intégration.
- Les données sources proviennent principalement des fichiers SIG, en particulier des fichiers *Shapefile* (ESRI), et des fichiers DAO, en particulier des fichiers *DWG* et *DXF*.

- Pour leur exploitation par le SIG, les données spatiales sont transformées à partir de leurs formats de fichier en un format de base de données.

- **Application**

Concernant l'application SIG, les principales hypothèses se rapportent aux :

- Fonctionnalités : Le SIG doit disposer des fonctionnalités de base proposées par la plupart des SIG, en particulier le stockage et la gestion, la représentation et l'analyse de données.
- Développement : Toutes les fonctionnalités du SIG sont entièrement développées dans le cadre de cette thèse.
- Technologies : Toutes les technologies utilisées doivent provenir du monde de l'open source pour des raisons de minimisation des coûts mis en place pour le développement du prototype SIG.

## **2. MODELISATION**

### **2.1. TYPE DES DONNEES**

Les outils SIG existants utilisent généralement des données spatiales stockées dans des fichiers de formats connus. Ces données peuvent provenir des fichiers SIG ou des fichiers DAO. Le prototype du SIG développé dans le cadre de cette thèse utilise à son tour des fichiers SIG, en particulier des *Shapefile* (ESRI), et des fichiers DAO, en particulier des fichiers *DWG* (DraWinG - Autodesk) et *DXF* (Data eXchange File - Autodesk).

Ces fichiers consacrent généralement plus d'intérêt aux données géométriques qu'aux données topologiques et attributaires. Les fichiers testés sont des fichiers représentant particulièrement des bâtiments historiques et quelques quartiers des villes y compris des MNT (Modèles Numériques de Terrain). Le SIG doit être alors capable de représenter tout objet spatial à travers l'ensemble de surfaces composant son enveloppe.

La plupart des fichiers sources utilisés pour tester le prototype de SIG :

- Ne contiennent pas des données topologiques surtout pour la représentation des bâtiments à travers des solides 3D.
- Ne contiennent pas assez de données attributaires très utiles pour le SIG.

Malgré ces limites, l'outil du SIG doit disposer des fonctionnalités pour :

- L'interprétation des solides 3D à partir des données initialement représentées par des surfaces.
- L'alimentation ou l'enrichissement des données géométriques / topologiques par des données attributaires pour améliorer les capacités d'interrogation et d'analyse.

Pour optimiser leur représentation et interrogation, les fichiers sources seront intégrés une structure de base de données. Pour ce faire, tout un processus d'intégration va être mis en place. Il n'est pas prévu de procédure de correction de géométries invalides. Nous partons du principe que les données initiales sont bien structurées et ne contiennent pas des erreurs géométriques.

### **2.2. MODELISATION DES DONNEES**

La modélisation spatiale porte généralement sur les propriétés géométriques et topologiques, voire attributaires, des objets spatiaux. En effet, un modèle spatial traduit la perception du concepteur des objets spatiaux. Il existe une multitude de modèles spatiaux proposés par des organismes

internationaux tels que l'ISO (International Organization for Standardization) et l'OGC (Open Geospatial Consortium) ainsi que les communautés des chercheurs.

Chaque modèle spatial utilise généralement un ensemble de primitives géométriques et topologiques pour représenter les objets du monde réel. Les primitives utilisées ainsi que leur combinaison dépendent des objets à représenter (objets naturels, objets artificiels, bâtiments, etc.).

La modélisation spatiale constitue la première phase dans le cycle du développement de notre outil SIG. Cette modélisation utilise des primitives géométriques devant être en accord avec les normes existantes, d'une part, et facile à implémenter dans un SGBDS, d'une autre part. Pour ce faire, un ensemble de méthodes de représentation surfacique et volumique des objets du monde réel seront étudiées pour en choisir les méthodes les plus appropriées pour nos modèles spatiaux urbains.

La modélisation spatiale sera réalisée à travers les trois niveaux de modélisation :

- Conceptuel : Il s'agit de la description à travers un ensemble de figures illustratives et de modèles conceptuels notre perception des objets spatiaux, en particulier des bâtiments.
- Logique : Il s'agit d'implémenter les modèles conceptuels dans une structure de base de données. A ce stade, les modèles peuvent être simplifiés pour une optimisation du stockage des données dans la BDS.
- Physique : C'est la dernière étape de modélisation et qui consiste à implémenter le modèle logique selon les spécificités d'un SGBD particulier. Pour ce faire, un ensemble de SGBDS libres seront étudiés pour le stockage des données utiles au prototype de SIG.

### 3. UTILISATEURS DU SIG

Il existe plusieurs catégories d'utilisateurs du SIG. L'outil SIG est, *a priori*, créé pour un usage privé au sein de l'équipe PAGE. Cependant, il est créé pour qu'il soit utilisé par différents types d'utilisateurs :

- Des utilisateurs qui créent (acquièrent) les données et désirent afficher les modèles spatiaux correspondants.
- Des utilisateurs qui utilisent ses modèles que ce soit dans un contexte purement spatial ou dans un contexte relatif à une localisation dans l'espace.

Avec le déploiement de notre outil SIG sur Internet, la communauté des utilisateurs du SIG est susceptible de s'agrandir de façon relativement importante. Pour ce faire, il est difficile de préciser d'une manière exhaustive leur nombre ou leur typologie.

Parmi les motivations du déploiement du SIG sur Internet est de favoriser son utilisation collective ou coopérative. En effet, sur un même projet peuvent travailler différents types d'utilisateurs (ceux qui créent les données, ceux qui alimentent le SIG par des données thématiques, ceux qui interrogent les données pour en dégager des interprétations, des décisions, etc.).

### CONCLUSION

Avant de rentrer dans les détails de l'état de l'art ainsi que les étapes du développement du prototype SIG, nous avons proposé ce chapitre pour expliquer les besoins, les domaines d'application et les utilisateurs de ce prototype. La section "*Etat de l'art*" reprend les principales méthodes de modélisation géométrique de la 3D, la modélisation géométrique et topologique des données spatiales (standards et travaux de recherche) et enfin les concepts relatifs au SIG (définitions, SGBDS, open source, etc.). Cette section explique les concepts fondamentaux liés aux travaux de cette thèse. Les sections suivantes décrivent nos contributions en termes de modélisation spatiale et du développement des différentes fonctionnalités du SIG.

## SECTION.I ETAT DE L'ART

## CHAPITRE.2 LA 3D, CONCEPTS &amp; MODELISATION

Résumé .....	32
Introduction .....	32
<b>1. Un peu d'histoire de la 3D</b> .....	<b>33</b>
<b>2. Quelques terminologies</b> .....	<b>35</b>
2.1. Le concept de « <i>modélisation</i> » .....	35
2.2. Le terme " <i>dimension</i> " .....	35
2.3. Les concepts de « 2D », « 2.5D » et « 4D » .....	35
2.4. Réalité virtuelle (RV) .....	36
2.5. Réalité augmentée .....	37
<b>3. Modélisation géométrique de la 3D</b> .....	<b>37</b>
3.1. Définition .....	37
3.2. Modélisation filaire : Exemple Fil de fer .....	38
3.3. Modélisation surfacique .....	39
3.3.1. Modélisation polygonale .....	39
3.3.2. B-Rep (Boundary Representation) .....	40
3.3.3. NURBS (Non Uniform Rational Basic Spline) .....	41
3.4. Modélisation volumique .....	42
3.4.1. CSG (Constructive Solid Geometry) .....	43
3.4.2. RBSO (Regularized Boolean Set Operation) .....	44
3.4.3. SPR (Spatial Partitioning Representation) .....	44
3.4.4. Octrees .....	45
3.5. Modélisation hybride : surfacique & volumique .....	46
3.6. Modélisation paramétrique .....	46
Conclusion .....	47

« *Un petit croquis vaut mieux qu'un long discours* »

(NAPOLEON BONAPARTE)

## **RESUME**

*La 3D, une notion de plus en plus familière aussi bien pour des utilisateurs potentiels que pour des néophytes, est maintenant omniprésente dans plusieurs domaines de notre vie quotidienne. Son essor phénoménal est dû en premier lieu au progrès technologiques fabuleux des outils informatiques. L'objectif majeur de la 3D consiste à construire un ensemble de modèles virtuels pour représenter des objets ayant ou non une existence physique. Un ensemble de méthodes sont proposées pour modéliser la géométrie des objets tridimensionnels. Chaque méthode est caractérisée par une perception particulière de tels objets. Le degré de réalisme varie d'une méthode à l'autre, l'approche la plus réaliste est celle qui imite au mieux des objets tridimensionnels.*

## **INTRODUCTION**

En modifiant notre perception du monde et en offrant des expériences interactives inédites, la 3D est par excellence une technologie au service de l'innovation. Si elle a déjà bouleversé le monde du jeu vidéo, le cinéma d'animation ou encore le secteur industriel, la 3D sort de son univers d'initiés pour révolutionner de plus en plus de domaines [BLINDENBERG, *et al*, 2007]. La modélisation tridimensionnelle (3D) est la représentation de l'espace qui nous entoure d'une manière se rapprochant de la perception humaine. Il existe deux principales interprétations du terme « 3D ». La première perçoit la 3D comme étant la *hauteur*, la *largeur* et la *profondeur* d'un objet donné. Alors que la deuxième la perçoit comme étant un ensemble de coordonnées *x*, *y* et *z* dans un repère à trois axes perpendiculaires. C'est "... *tout objet ou espace possédant une longueur, une largeur et une épaisseur. Sur un écran, une image en 3D ajoute donc le relief aux images 2D. Le relief peut cependant n'être qu'un leurre visuel, par exemple pour les calculs 3D des interfaces graphiques*" [PHPMYVISITES, 2010].

*On entend par «objet», un élément de la réalité qui est cartographié et par le «nombre de dimensions» de cet objet, le nombre d'axes (ou de coordonnées) nécessaires pour localiser des points les uns par rapport aux autres (cet objet étant alors considéré comme un univers spécifique). Il s'agit de l'espace occupé par l'objet lui-même (ex. largeur, longueur, épaisseur) et non l'espace occupé par son rectangle englobant 2D ou 3D. Ainsi, indépendamment des axes de l'univers du système, un objet ponctuel n'aura aucune dimension (0D), un objet linéaire aura une seule dimension soit sa longueur (1D), un objet surfacique aura deux dimensions (par exemple, une largeur et une longueur) et conséquemment une surface calculable (2D). Un objet volumétrique aura trois dimensions occupant un espace ayant des valeurs de longueur, largeur et hauteur (3D) et conséquemment un volume calculable [LARRIVEE, *et al*, 2006].*

Les notions d'"*image en relief*" et "*image 3D*" ont été souvent confondues. L'image en relief est obtenue grâce à des procédés de filtres optiques (par exemple, anaglyphes), qui donnent une illusion de profondeur de champ, ainsi que de relief. Alors que l'image 3D est une image définie sur trois axes que l'on pourrait appeler longueur, hauteur et profondeur ; le monde que nous percevons est aussi en trois dimensions, puisqu'il est défini sur ces trois axes [MULTIMANIA, 2010].

Pendant ces dernières années, la 3D a envahi des domaines assez diversifiés pour des fins assez variées. On parle de la 3D dans les logiciels de CAO (Conception Assistée par Ordinateur), dans le cinéma 3D, dans la simulation des phénomènes du monde réel, dans le domaine des données spatiales, dans les jeux vidéo, etc. C'est une représentation qui essaye d'imiter, selon le niveau de détail et le

réalisme attendu du modèle 3D, des objets du monde qui nous entoure. Ce chapitre aborde le sujet de la 3D en décrivant brièvement son historique, en définissant quelques termes relatifs au domaine de la 3D et en expliquant les principales approches de modélisation géométrique et topologique de la 3D.

## 1. UN PEU D'HISTOIRE DE LA 3D

La 3D a souvent été reliée aux sciences de l'ordinateur pour des disciplines diverses comme la CAO (Conception Assisté par Ordinateur), le graphisme 3D, les jeux vidéo, le cinéma 3D, la simulation, etc. Vue à l'écran d'un ordinateur, la 3D est exprimée par un ensemble d'images donnant une impression de relief.

Revenons un peu dans l'histoire de l'infographie et de la 3D en se basant sur quelques découvertes technologiques et avancées matérielles qui ont révolutionné l'industrie 3D selon [SHADOWS, 2010].

Pendant les années 1950, les premiers jeux graphiques sont apparus. L'année 1952 marque l'apparition du premier jeu graphique sur ordinateur appelé *OXO*. Un peu plus tard en 1958, William HIGINBOTHAM a proposé son jeu "*Tennis for two*". Il s'agit d'un jeu de tennis fonctionnant avec un ordinateur relié à un oscilloscope.

Pendant les années 1960, le terme "*computer graphics*", signifiant infographie, a été inventé. Cette décennie a connu une succession de plusieurs innovations très importantes. En 1962, Pierre BEZIER a inventé la "*courbe de Bézier*". Il s'agit de courbes paramétriques servant, au départ, à la conception de pièces automobiles. Les courbes de Bézier seront utilisées plus tard en 3D pour tracer des courbes et des surfaces. En 1963, Ivan SUTHERLAND a développé le logiciel "*Sketchpad*" qui consistait à créer interactivement des dessins à l'aide d'un crayon optique. En 1966, Ralph BAER a inventé un boîtier se raccordant aux télévisions qui constituait une ébauche de console de jeux.

Pendant les années 1970, plusieurs inventions relatives aux jeux graphiques sont apparues. En 1972, "*Magnavox Odyssey*", première véritable console de jeu, a été construite et dont le modèle était conçu par Ralph BAER. En 1976, Richard T. HEFFRON a utilisé pour la première fois une image de synthèse dans son film "*Future world*" pour modéliser un visage et une main animés.

Pendant les années 1980, le cinéma 3D a connu un essor formidable avec l'apparition d'une série de films entièrement en 3D ou des films contenant quelques séquences et objets en 3D tels que "*La fin d'une ère*" en 1982, "*The Last Starfighter*" en 1984, "*Knick Knack*" en 1989, etc.

Pendant les années 1990, la 3D a commencé à se développer réellement, notamment avec l'apparition des ordinateurs de plus en plus puissants. Outre les films 3D et les jeux vidéo qui ont été réalisés pendant cette décennie, quelques avancées considérables ont été mises en place en termes d'API (Application Programming Interface) pour la programmation 3D tels que "*Allegro (Atari Low-Level Game Routines)*" créé par Shawn HARGREAVES pour "*Atari-ST*" en 1990, OpenGL (Open Graphics Library) développé par "*Silicon Graphics*" en 1992, "*DirectX*" développé par "*Microsoft*" en 1995, etc.

Pendant les années 2000, la 3D a pris une « autre dimension » en se généralisant sur d'autres domaines notamment les sites web, les villes en 3D, le domaine spatial, les simulations, etc. Elle est devenue un axe de recherche très répandu dans le monde entier. Dans le domaine spatial, la 3D a été employée pour combler les insuffisances constatées des données spatiales 2D. Des nouvelles terminologies et technologies ont également été inventées tels que les GPS (Global Positioning System) 3D, les télévisions 3D, les géo-portails 3D, etc. Il est maintenant possible de visualiser toute une ville en 3D sur un petit téléphone portable.

La figure (Fig2.1) montre quelques exemples d'évolution de la 3D au cours du temps en termes de capacités de visualisation.



Voiture 3D (1967)



3D PacMan (1986)



Jurassic Pac (1997)



The Snog (2004)

**Fig2.1.** Quelques exemples d'évolution de la 3D au fil du temps

La création d'une scène 3D est matérialisée par plusieurs étapes partant de l'étape de modélisation, passant par l'étape d'animation jusqu'à l'étape de visualisation ou de rendu 3D de cette même scène.

La modélisation des données 3D est un processus consistant à représenter schématiquement ou mathématiquement, à travers un modèle, les composants d'un objet du monde réel. Tel modèle doit se rapprocher le maximum des caractéristiques physiques de l'objet en question. Il existe une multitude de méthodes pour la modélisation 3D telles que la modélisation surfacique, volumique, etc.

L'animation consiste à définir comment un objet donné se déplace et se déforme au cours du temps. Il s'agit d'une description temporelle de tel objet. Il existe deux principaux types d'animations : i) Les animations prédéfinies dont l'animation est définie une fois pour toute et ne peut pas être changée telles que les animations dans les films 3D. ii) Les animations dynamiques ou interactives qui dépendent des IHM (Interactions Homme-Machine) que l'on retrouve dans les jeux vidéos.

La visualisation ou le rendu d'une scène 3D est le processus de traduction ou de transformation du modèle 3D, défini lors de l'étape de modélisation, en un ensemble d'images, affichées à l'écran, donnant une impression de relief. Il existe deux principaux types de rendus : i) Rendu photo-réaliste (PR) qui consiste à reproduire l'apparence du monde réel avec la plus grande fidélité possible. Il traduit les objets du monde réel en intégrant tellement de détails (textures, lumière, ombre, réflexions, etc.) de telle façon qu'on n'arrive pas à distinguer l'image artificielle de celle prise par une caméra élaborée [OSTROMOUKHOV, 2002]. ii) Rendu non photo-réaliste (NPR) qui consiste à reproduire les objets du monde réel par des dessins 2D ou 3D. Le terme de "*Rendering Non Réaliste*" désigne les techniques logicielles de rendu (et par extension de traitement d'images) qui donnent à des images réelles ou de synthèse un style graphique non réaliste, généralement inspiré des techniques traditionnelles (cartoon, gravure, peinture, etc.) [FOURMIROUGE, 2010]. La figure (Fig2.2) suivante montre un exemple d'un bâtiment PR (a) et un bâtiment NPR (b).



(a)



(b)

**Fig2.2.** Exemples de rendu réaliste(a) et de rendu non réaliste(b) des bâtiments

(Sources : (a) NP3D.fr

(b) Banque d'images 3D Google Sketchup)

La 3D, a connu, surtout pendant les deux dernières décennies, un essor fabuleux. Elle était employée dans des nombreux domaines à des fins assez diversifiées. La 3D aboutit à une représentation plus réaliste ainsi que des calculs plus avancés sur des objets du monde réel. Un objet donné pourra être alors vu sous tous les angles ce qui donne une flexibilité très importante pour dégager des interprétations, effectuer des calculs assez précis, etc. Que ce soit pour des logiciels de CAO, de simulation, des outils spatiaux, etc. la modélisation de données 3D est une étape primordiale qui précède les étapes d'animation et de rendu de la scène.

## 2. QUELQUES TERMINOLOGIES

Avant d'aborder le sujet de modélisation tridimensionnelle, il est très important d'éclaircir quelques notions relatives au domaine de la modélisation des objets du monde réel.

### 2.1. LE CONCEPT DE « MODELISATION »

On entend par "modèle" « *ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose* ». Il s'agit d'une « *catégorie, variété particulière, définie par un ensemble de caractères et lesquelles peuvent se rapporter à des faits ou objets réels* ». On utilise également le terme modèle dans le cas d'une « *représentation simplifiée d'un processus, d'un système* », ou en mathématique dans le cas d'un « *modèle formé par description mathématique et destiné à simuler un tel processus* ». [ROBERT, 2008].

Un modèle est donc une représentation simplifiée d'une réalité, et dans « *la mesure où il reflète le monde tel que nous le percevons, il n'est pas, la plupart du temps, unique* ». [DENEGRE et al, 1997].

Un modèle est le résultat d'une opération de "modélisation". Ainsi, nous pouvons définir la "modélisation" comme « *Opération par laquelle on établit le modèle d'un système complexe, afin d'étudier plus commodément et de mesurer les effets sur ce système des variations de tel ou tel de ses éléments composants* » [GIRAUD, et al, 1974]. La modélisation « *... ce n'est donc rien d'autre que la pensée organisée en vue d'une finalité pratique. Modèle est synonyme de théorie, mais avec une connotation pratique : un modèle, c'est une théorie orientée vers l'action qu'elle doit servir* » [GOLLE, 2004].

### 2.2. LE TERME "DIMENSION"

Le terme dimension est défini comme la « *grandeur réelle, mesurable, qui détermine la portion de l'espace occupée par un corps [...] les dimensions d'un objet, d'un meuble : largeur, longueur, épaisseur, hauteur, profondeur* » [ROBERT, 2008]. C'est « *la mesure de chacune des grandeurs nécessaires à l'évaluation des figures et des solides (longueur, largeur, hauteur ou profondeur* ». En mathématique, la dimension d'un espace est « *le nombre de paramètres nécessaire pour situer un point de cet espace [...] espace à une, deux, trois dimensions : la ligne, la surface, l'espace. Espace à dimension  $n$  est le produit cartésien  $\mathbb{R}^n$ , dont l'élément quelconque  $(x_1, \dots, x_n)$  est appelé point* » [LAROUSSE, 2008].

### 2.3. LES CONCEPTS DE « 2D », « 2.5D » ET « 4D »

La modélisation 2D est un moyen relativement simple pour dessiner des graphiques pouvant être utiles dans différents domaines tels que les plans des bâtiments, les diagrammes conceptuels, les réseaux, etc. Si on utilise la définition du terme "dimension" établie ci-dessus, la 2D peut être définie comme tout objet ne comportant que les dimensions de la longueur et la largeur. La 2D est notamment utile pour la modélisation simplifiée des objets qui sont, en général, tridimensionnels. Il s'agit,

généralement, d'établir une projection de tels objets dans un plan 2D. En informatique, même la représentation 3D est en réalité une extension de la 2D du fait que les supports utilisés ne disposent que de deux dimensions. C'est l'interprétation visuelle du modèle qui le qualifie de bidimensionnel ou tridimensionnel. La modélisation 2.5D est une représentation intermédiaire entre la 2D et la 3D. Le principe consiste à exprimer la hauteur des objets par l'ajout d'une et une seule valeur Z (altitude) à chaque point X, Y. Toute géométrie 2.5D peut être convertie en une géométrie 2D en éliminant la coordonnée Z de chaque sommet de l'objet. [GERBE, 2002] décrit dans le tableau (Tab2.1) les différentes dimensions entre la 2D et la 3D.

Dimension	Description	Exemples de SIG
2D <sup>1/4</sup>	Le Z des objets est géré comme un attribut des objets, ce qui le limite à quelques valeurs possibles par objet (Zmin, Zmax par exemple). Les fonctions topologiques, d'analyse ou de visualisation n'utilisent pas le Z	MapInfo (MapViewer)
2D <sup>1/2</sup>	Tous les points de chaque objet sont stockés avec leurs trois coordonnées : X, Y, Z. Le SIG manipule des points et des lignes 3D, mais contient différentes restrictions : - Pas de surfaces 3D (mais possibilité d'avoir des surfaces 2D avec des limites en 3D). - Un seul Z par point 2D. - Primitives géométriques 3D mais moteur topologique 2D (lignes 3D mais les intersections ou les distances sont calculées en 2D)	ArcGIS (3D Analyst) GeoConcept (Virtual Geo)
2D <sup>3/4</sup>	Gère les surfaces 3D et le moteur topologique inclus les fonctions 3D. Les calculs de distances et d'intersections sont effectués en 2D.	Cadcorp (SIS)
3D	Gère des données 3D (jusqu'aux volumes tels que les sphères) y compris la sémantique. Possède des capacités d'analyse 3D.	?

**Tab2.1** De la 2D à la 3D [GERBE, 2002]

Parfois, la définition d'un objet du monde réel peut dépasser sa description géométrique et/ou topologique. En effet, tel objet peut évoluer dans le temps. De ce fait, l'ajout d'une dimension temporelle aux trois dimensions géométriques permettra de décrire l'évolution ou la déformation d'un objet au cours du temps. Ainsi, la 4D est définie, par extension de la définition de la 3D, comme la largeur, la longueur, la hauteur et le temps utilisés pour la description d'un objet donné. Le fait que l'unité principale pour la mesure des distances est le mètre et celle pour la mesure du temps est la seconde, nous laisse réfléchir sur la manière de leur combinaison. Comment peut-on combiner des dimensions n'ayant pas la même unité de mesure ? A la fin du 19<sup>ème</sup> siècle, les physiciens, tels que HELMHOLTZ, ce sont intéressés à la théorie de RIEMANN qui a proposé une 4<sup>ème</sup> dimension dans laquelle les objets doivent disparaître pour réapparaître ailleurs. La dimension temporelle décrit, à tout moment, les caractéristiques d'un objet donné telles que sa position dans l'espace, son orientation, ses apparences physiques, son existence, etc.

#### 2.4. REALITE VIRTUELLE (RV)

Il existe plusieurs mots clés pour définir la RV tels que l'interaction, temps réel, monde artificiel, virtualité, etc. La RV est « *un outil de plus en plus utilisé pour étudier la cognition, il permet en effet de fournir une situation expérimentale pleinement contrôlée avec un contexte riche. De plus, cette technique permet des interactions actives entre le sujet et les stimuli* » [PLANCHER, et al, 2007]. Il s'agit de mettre à disposition de l'utilisateur un ensemble de techniques et outils informatiques lui aidant d'interagir en temps réel avec un environnement virtuel imitant un phénomène réel. Mais quel est le rapport entre la RV et la 3D ? La RV peut se servir de la 3D pour la modélisation de l'environnement virtuel. Avec la 3D, on peut construire des objets relativement réalistes.



**Fig2.3.** Exemple d'une application de Réalité Virtuelle créé par le Laboratoire de Perception et Contrôle du Mouvement en Environnement Virtuel (LPCMV)

Le VRML (Virtual Reality Markup Language) est un format structuré pour la description et la présentation 3D des objets réels ou virtuels. Il propose une multitude d'instructions pour décrire les primitives géométriques, les textes, les images, les lumières, les sons, etc. C'est un langage qui consiste à créer des scènes 3D statiques ou dynamiques.

## 2.5. REALITE AUGMENTEE

La réalité augmentée (RA) est une extension de la RV. C'est une association du virtuel et du réel. La RA peut être définie comme « *un paradigme d'interaction qui est né de la volonté de fusionner les capacités de traitements informatiques et notre environnement physique de tous les jours* ». Mais qu'est-ce que la RA comparée à la RV ? « *En RV, l'utilisateur est complètement plongé dans un monde reconstitué avec des données informatiques : l'utilisateur est coupé du monde réel. L'immersion en RV est donc totale, à l'opposé de la RA dont l'essence même est de maintenir l'utilisateur au contact de son environnement réel* » [DUBOIS, et al, 2000]. La notion de RA consiste donc à compléter le monde réel qui nous entoure par des éléments fictifs ou virtuels n'ayant pas une existence physique. L'application de la RA a envahi des domaines assez divers tels que l'industrie, l'apprentissage, la simulation, etc.

## 3. MODELISATION GEOMETRIQUE DE LA 3D

### 3.1. DEFINITION

La géométrie est « *la sciences de figures de l'espace physique [...] En mathématique, la géométrie est l'étude des relations entre points, droites, courbes, surfaces et volumes de l'espace réel* » [ROBERT, 2008]. Pour EUCLIDE, la géométrie est la science des figures de l'espace. Alors que pour Felix Christian KLEIN (1871), c'est l'étude des invariants d'un groupe de transformations de l'espace. Il existe plusieurs types de géométries telles que la géométrie Euclidienne, la géométrie analytique, la géométrie algébrique, etc.

La modélisation 3D dépend des objets à représenter. Elle permet de définir les formes des objets du monde réel généralement par un ensemble de "*primitives géométriques*". Le terme "*primitif*" ou "*primitive*" signifie « [...] *qui est à son origine ou près de son origine [...] Qui est la source, l'origine (d'une autre chose de même nature* » [ROBERT, 2008]. Ainsi, le terme "*primitive géométrique*" dans le domaine de l'infographie et des systèmes CAO signifie tout objet géométrique atomique. On peut alors définir une primitive géométrique comme un composant élémentaire faisant partie d'un objet géométrique plus complexe.

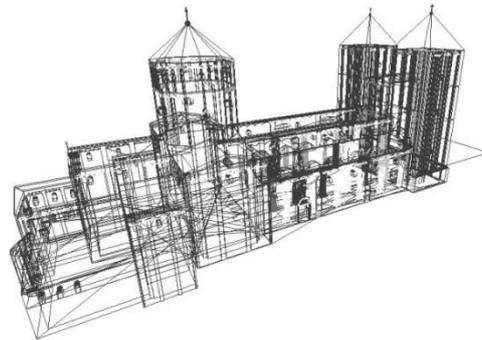
La définition de primitives géométriques dépend de la méthode de modélisation utilisée. Par exemple, l'approche de modélisation par frontières (B-Rep ou Boundary Representation) définit un ensemble de primitives géométriques telles que les points, les lignes, les polygones, etc. L'approche de modélisation par solide (CSG ou Constructive Solid Geometry) utilise quant à elle d'autres types de primitives géométriques comme les cubes, les sphères, les cônes, etc.

Il existe différentes approches et méthodes pour modéliser la géométrie 3D. Chaque approche utilise une représentation particulière des objets du monde réel. Ces méthodes peuvent être classées en cinq familles : Modélisation filaire, modélisation surfacique, modélisation volumique, modélisation hybride (surfacique et volumique) et modélisation paramétrique.

### 3.2. MODELISATION FILAIRE : EXEMPLE FIL DE FER

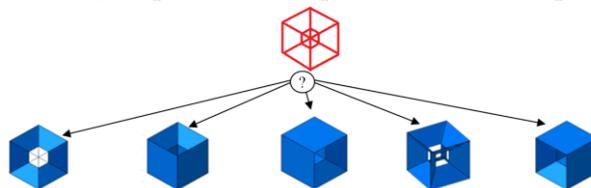
Comme son nom l'indique, la modélisation "*Fil de Fer*" se sert de lignes ou d'arcs pour la représentation des objets du monde réel. Elle consiste à tracer les arcs représentant l'intersection des surfaces mitoyennes ou relier les différents sommets de l'objet par des lignes ou des courbes. Les surfaces d'un objet modélisé par la méthode "*Fil de Fer*" sont alors représentées par leurs contours. Toutes ces surfaces sont visibles y compris les surfaces du côté opposé et les surfaces situées à l'intérieur de l'objet qui sont censées être cachées.

La figure (Fig2.4) suivante montre un exemple d'une modélisation "*Fil de Fer*" de l'église canoniale de NIEDERMUNSTER affichée par le logiciel SIG 3D développé dans le cadre de cette thèse.



**Fig2.4.** Exemple de modélisation "*Fil de Fer*" (Eglise canoniale de NIEDERMUNSTER)

La modélisation "*Fil de Fer*" est une technique utilisée par la plupart des logiciels de modélisation 3D. Elle permet de visualiser d'une manière très simple les objets du monde réel. Comparée à la modélisation surfacique ou par solide, la méthode "*Fil de Fer*" peut être qualifiée comme la méthode la moins complexe pour la représentation des objets géométriques. Vu que les "*Fils de Fer*" sont simples et faciles à calculer, ils sont souvent utilisés pour le rendu d'objets complexes. En cas de nécessité de plus de détails dans le modèle, les fils de fer sont complétés par l'ajout de surfaces et textures. La modélisation "*Fil de Fer*" est avantageuse par rapport aux méthodes de modélisation par surface ou par solide en termes de temps de calcul notamment pour manipuler, modifier ou stocker des formes géométriques. Cependant, les modèles "*Fils de Fer*" peuvent être ambigus du fait que deux observateurs du même modèle peuvent avoir des interprétations différentes. La figure (Fig2.5) montre un exemple d'un modèle filaire ayant plusieurs interprétations visuelles possibles.



**Fig2.5.** Exemple d'ambiguïté d'interprétation d'un modèle filaire [MARANZANA, 2007]

Les modèles assez complexes contenant un nombre important de lignes risquent de ne plus être clairs pour l'observateur, voire parfois impossibles à percevoir.

### 3.3. MODELISATION SURFACIQUE

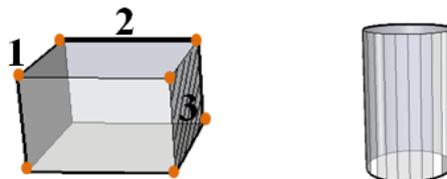
Quelque soit l'approche utilisée pour la modélisation 3D, l'apparence physique d'un objet donné peut être relativement la même ou, le cas échéant, comparable. Mais ce qui distingue une approche de modélisation d'une autre ce sont les primitives géométriques utilisées ainsi que les combinaisons utilisées pour former des objets 3D simples ou complexes. Parmi les approches de modélisation les plus répandues, nous trouvons la modélisation surfacique qui modélise les objets à travers les surfaces composant leurs enveloppes.

#### 3.3.1. MODELISATION POLYGONALE

Comme son nom l'indique, la modélisation polygonale se sert de polygones pour représenter les objets du monde réel. Les polygones sont utilisés depuis les débuts de l'infographie vu leur simplicité en termes de calcul, de rendu et de stockage. Malgré l'apparition des nouvelles méthodes de modélisation telle que les NURBS, on peut se poser la question : pourquoi utilise-t-on encore les polygones ? En fait, les polygones permettent d'exploiter au maximum une simplicité de calcul, une rapidité de modélisation, une facilité de modélisation d'objets complexes, une flexibilité, etc. [RUSSO, 2006].

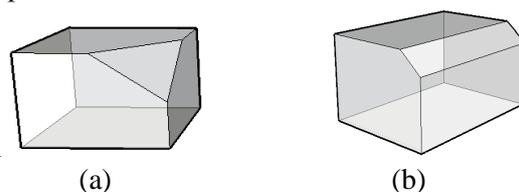
Chaque polygone est décrit par la liste de sommets et d'arêtes qui le composent. Un polygone peut contenir des surfaces planes appelées "*faces*". Il est alors possible de créer, avec cette méthode, des faces simples ou des collections de faces constituant un maillage. Les polygones constituant un objet donné peuvent être orientés. On peut dans ce cas distinguer les différentes faces de chaque polygone et, par extension, définir l'intérieur et l'extérieur de chaque objet.

La structure polygonale consiste en trois sous-éléments basiques : les *nœuds*, les *arcs*, les *faces* ou polygones (*Fig2.6*).



**Fig2.6.** Composants d'une structure polygonale : 1 : un nœud, 2 : un arc et 3 : une face

La méthode de modélisation polygonale propose également un ensemble d'opérations ou d'outils de transformation sur les polygones. Parmi ces opérations, les plus utilisées sont l'extrusion, la coupe, l'attachement et le détachement, la soudure, le lissage de polygones, la duplication et le miroir, le chanfrein, etc. Par exemple, une opération d'extrusion sur un polygone donné suivie par une opération de coupe résulte en la création de deux volumes. Par contre, une opération de chanfrein consiste en une opération de coupe en un coin ou une arête d'un volume donné (*Fig2.7*).

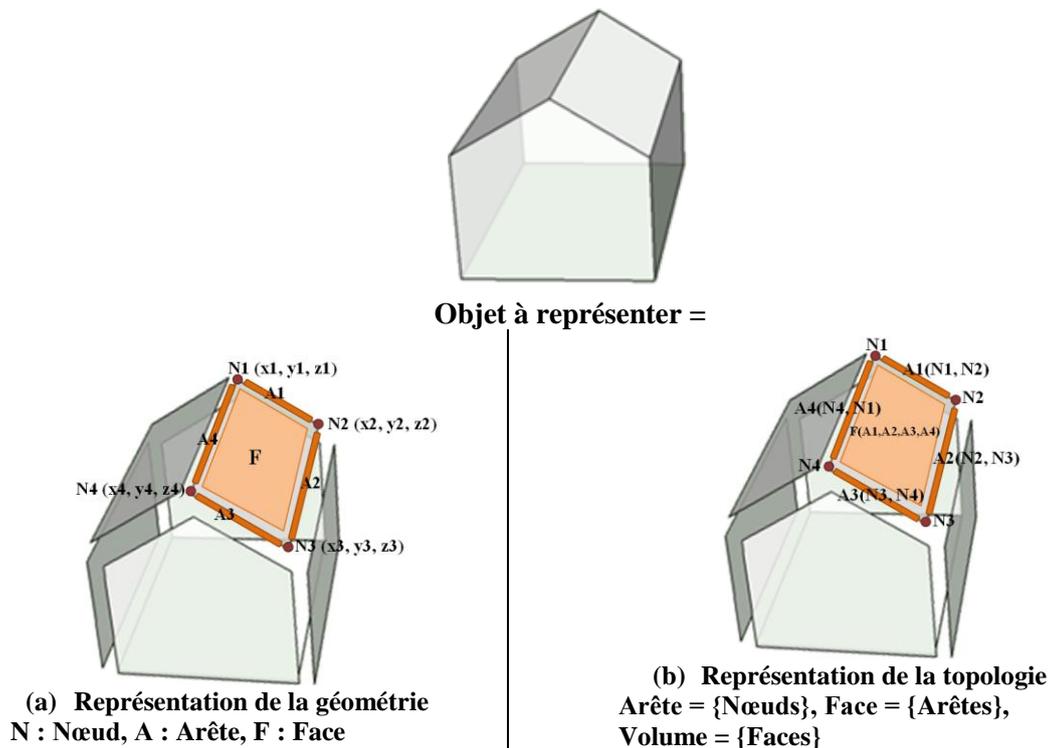


**Fig2.7.** Opération de chanfrein sur un cube : (a) suivant un coin, (b) suivant une arête

La technique de modélisation par polygones est souvent utilisée dans les logiciels 3D tels que 3DSMax, MAYA, 3DLUVR, etc. L'avantage majeur des polygones est la rapidité de représentation d'objet par rapport aux autres méthodes de modélisation. Néanmoins, cette méthode présente quelques inconvénients. La modélisation polygonale n'est pas très adaptée pour représenter, avec précision, des surfaces non planes. Il faut utiliser beaucoup de polygones pour représenter approximativement ce type de surfaces. Cela peut avoir des conséquences négatives en termes de taille et stockage d'objets complexes.

### 3.3.2. B-REP (BOUNDARY REPRESENTATION)

B-Rep désigne en anglais "*Boundary Representation*" ce qui signifie en français "Modélisation par Frontières" ou "Modélisation par Frontières Surfacciques" [RAMOS, 2003], ou encore "Modélisation par les bords" [BRANDEL, 2009]. B-Rep est une branche de la modélisation surfaccique dans laquelle un objet est décrit en termes de surfaces composant ses frontières : *nœuds*, *arêtes* et *faces* [JARROUSH, 2004]. Cette méthode représente alors les objets du monde réel par leurs limites. B-Rep distingue la géométrie de la topologie d'un objet donné. La géométrie décrit la forme et la position de chaque nœud, arête et face. Par exemple, la géométrie d'un nœud est définie par sa position dans l'espace généralement exprimée par ses coordonnées  $(x, y, z)$ . Par contre, la topologie exprime les relations de connexion entre les faces, arêtes et nœuds [ZLATANOVA, *et al*, 2002] cité dans [KOUSSA, *et al*, 2009]. Par exemple, un bâtiment peut être modélisé à travers un ensemble de surfaces (représentant les façades et les toits) partageant des nœuds et des arêtes communes tel que représenté dans la figure (Fig2.8).



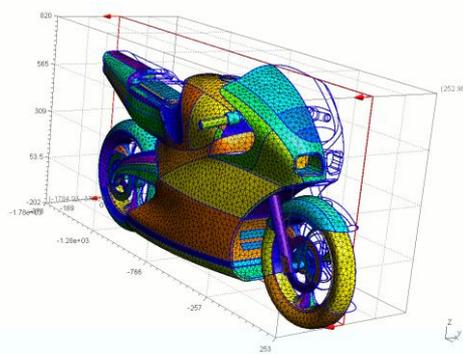
**Fig2.8.** Exemple d'une représentation B-Rep d'un bâtiment

Pour assurer la validité topologique des objets solides construits avec l'approche B-Rep, on utilise les opérateurs d'EULER dans la formule suivante [THALMANN, *et al*, 2003].

$$V - E + F = 2(S - H) + R$$

Formule d'Euler ( $V$  : le nombre de sommets,  $E$  : le nombre d'arêtes,  $F$  : le nombre de faces,  $S$  : le nombre de composants disjoints,  $H$  : le nombre de trous, et  $R$  le nombre de cavités dans une face)

L'approche de modélisation B-Rep présente plusieurs avantages. Elle décrit de manière explicite et détaillée les objets du monde réel en termes de nœuds, arêtes et faces. C'est une approche capable de représenter des objets complexes avec une définition exacte de toutes les surfaces du modèle final. L'avantage majeur d'une telle approche est la double présentation géométrie-topologie des différentes primitives dans un même modèle. B-Rep offre également la possibilité d'effectuer des opérations complexes sur des objets 3D telles que le calcul de volumes. Cependant, B-Rep nécessite un volume de stockage de données très important. Des procédures d'optimisation ou de simplification doivent être mises en place pour accélérer l'interrogation de modèles 3D. B-Rep n'est pas très adaptée pour la modélisation des objets contenant des surfaces non planes. Pour imiter de tels objets, il faut utiliser un maximum de surfaces. Cela peut dégrader les capacités de calcul et de ce fait alourdir de manière préjudiciable l'exploitation des modèles 3D. Cette approche est utilisée dans les logiciels de CAO tels qu'ArchiCAD, AutoCAD, etc. La figure (Fig2.9) montre une moto modélisée à travers un ensemble de primitives B-Rep réalisé avec le logiciel Gmsh.

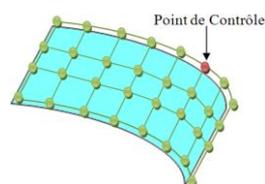


**Fig2.9.** Exemple d'une moto modélisée avec des primitives B-Rep [GEUZAIN, *et al*, 2009]

### 3.3.3. NURBS (NON UNIFORM RATIONAL BASIC SPLINE)

NURBS est l'acronyme de "*Non Uniform Rational Basic Spline*" ce qui signifie en français Spline Basique Rationnelle Non Uniforme. Les NURBS appartiennent à la famille des courbes (ou surfaces) définies par des polynômes. L'idée sous jacente est d'imposer des contraintes à une courbe (ou surface) par le biais de points de contrôle [LAFERTE, *et al*, 2009]. La modélisation par NURBS consiste en un ensemble ou un réseau de courbes interpolées entre les différents points de contrôle. Pour interpoler telles courbes, on peut soit utiliser des algorithmes NURBS (tel que l'algorithme de Boor), manipuler les tangentes des courbes de Bézier, etc.

La figure (Fig2.10) suivante montre une représentation simplifiée d'une surface NURBS.



**Fig2.10.** Exemple d'une surface NURBS

Formellement, NURBS peut être représentée mathématiquement à l'aide de la formule de Cox-De Boor suivante :

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i}$$

Avec : les  $P_i$  sont les points de contrôle, les  $w_i$  les poids, et les  $N_{i,p}(u)$  sont les fonctions de base B-Splines de degré  $p$  définies sur un vecteur nodal non-uniforme :

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\}$$

La modélisation d'un objet géométrique en utilisant des courbes NURBS consiste en un maillage permettant de subdiviser une surface en un ensemble de sous-surfaces dépendant de la complexité de la courbe. Par exemple, une surface plane ou quasi-plane aura un nombre de subdivisions qui tend vers le nul.

NURBS est utilisée dans les logiciels de CAO pour la modélisation d'objets 3D, pour la compression d'images, etc. C'est une technique de modélisation très précise car elle représente mathématiquement des objets géométriques. Elle est capable de décrire des objets très complexes même d'une manière approximative. C'est une technique qui aboutit à une construction simple d'objets géométriques ainsi qu'une implémentation facile. En effet, les NURBS offrent plusieurs avantages qui les rendent attractives pour les applications de modélisation. Elles sont plus générales que les courbes de Bézier ou B-Spline. Leur évaluation est simple, rapide et stable du point de vue calculatoire. Elles offrent une représentation mathématique commune pour des surfaces de formes libres ainsi que des formes analytiques couramment utilisées telles que les quadriques naturelles, les torii, les surfaces extrudées et les surfaces de révolution. Les courbes NURBS sont faciles à modifier à travers la manipulation des points de contrôle, les poids, les nœuds, etc. Cependant, cette approche présente quelques inconvénients. Un mauvais choix des poids peut mener à un mauvais paramétrage des courbes et des surfaces. Cette approche nécessite un stockage plus important pour définir des formes géométriques traditionnelles telles que les cercles, les sphères, etc. Quelques algorithmes sont numériquement instables tels que l'algorithme de "Inverse Point Mapping". De plus, quelques algorithmes marchent mieux avec d'autres formes que les NURBS tels que le calcul de la courbe d'intersection de deux surfaces, etc. [WEN, *et al*, 2005].

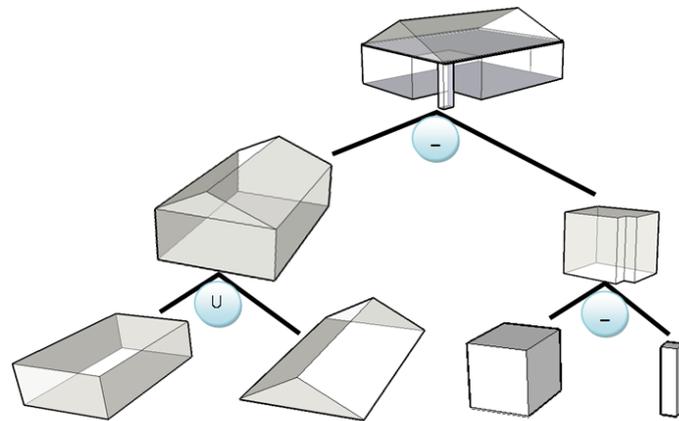
Il existe d'autres méthodes de modélisation de la 3D telles que la "*Modélisation par Subdivision Surfacique*", "*Modélisation par Surfaces Implicites*", etc. Par exemple, la méthode de modélisation par subdivision surfacique est en quelque sorte un mélange des deux méthodes de modélisation polygonale et les NURBS. Le principe de la méthode est de subdiviser une surface ou une partie d'une surface afin d'y ajouter plus de détails. Cette méthode est massivement utilisée dans les logiciels de modélisation 3D tels que Blender, Maya, 3DSMax, Lightwave, Softimage, Modo, etc.

### 3.4. MODELISATION VOLUMIQUE

Alors que la modélisation surfacique utilise des surfaces pour représenter la géométrie des objets 3D, la modélisation volumique se sert de solides. Le principe consiste à combiner des objets simples pour former des objets plus complexes. La combinaison des différents objets s'effectue généralement à l'aide des opérations booléennes.

### 3.4.1. CSG (CONSTRUCTIVE SOLID GEOMETRY)

CSG désigne en anglais "*Constructive Solid Geometry*" ce qui signifie en français "*Géométrie de Construction de Solides*". C'est une branche de la modélisation volumique dans laquelle un objet du monde réel est représenté par un ensemble d'objets élémentaires appelés solides. CSG se sert donc de primitives solides telles que les sphères, les cubes, les cônes, etc. pour la construction d'objets complexes. La construction de tels objets se base sur des opérations booléennes entre des objets solides. Les opérations booléennes sont l'intersection ( $\cap$ ), l'union ( $\cup$ ) et la différence ( $-$ ). L'union de deux objets produit un objet qui englobe l'espace occupé par ces deux objets. L'intersection produit un objet qui réunit l'espace où les deux objets se chevauchent. La différence est un opérateur qui dépend de l'ordre des opérandes. Il s'agit du premier objet privé de l'espace partagé par les deux objets [E.KOLB 1994]. La figure (Fig2.11) montre un bâtiment modélisé à travers un ensemble de primitives CSG.

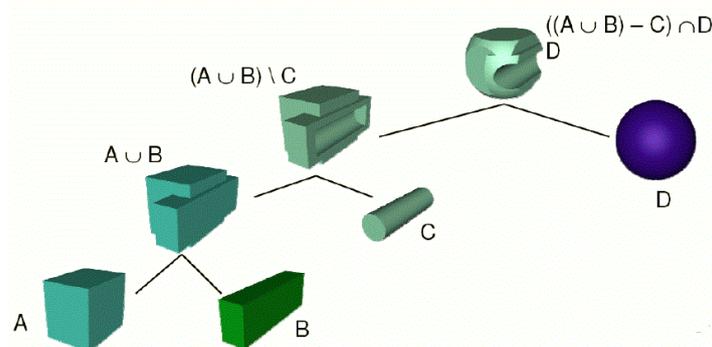


**Fig2.11.** Exemple d'une représentation CSG d'un bâtiment utilisant un arbre d'opérations booléennes

Comme CSG utilise des primitives solides élémentaires telles que les cubes, les sphères, les cônes, etc. pour la construction d'objets complexes, il est possible de calculer la surface globale de tels objets car il existe une équation de calcul de surface pour chaque primitive élémentaire. Par exemple l'équation de calcul de la surface d'une sphère est :  $S = 4\pi r^2$  (avec  $r$  est le rayon de la sphère), la surface d'un cylindre :  $S = 2\pi r^2 + 2\pi rh$  où  $r$  est le rayon et  $h$  la hauteur du cylindre, etc.

CSG est utilisée par plusieurs outils de CAO tels qu'AutoCAD qui propose trois opérations booléennes sur des primitives CSG : union, intersection et différence.

La figure (Fig2.12) suivante montre un modèle CSG d'une pièce mécanique réalisée à travers trois opérations booléennes sur des primitives solides simples et composées.

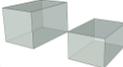
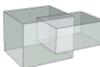


**Fig2.12.** Exemple d'une pièce mécanique modélisée avec des primitives CSG [DUBUISSON, 2007]

L'approche de modélisation CSG a plusieurs avantages. La modélisation d'un objet à travers un ensemble de blocs élémentaires est sémantiquement plus significative que la modélisation par des surfaces B-Rep car la plupart des objets du monde réel sont en réalité composés de solides élémentaires et non pas de surfaces. La modélisation en utilisant des primitives solides et des opérations booléennes est plus intuitive que la détermination de toutes les surfaces d'un objet donné (sauf si l'ensemble de primitives n'est pas approprié pour l'objet à modéliser) [BRENNER, 2004]. Les objets construits par cette méthode ont des frontières réalistes quelque soit le niveau de détail. Cependant, on peut atteindre facilement des opérations de construction complexes ce qui peut dégrader les capacités de calcul du rendu. De plus, les primitives CSG et les opérations booléennes sont parfois insuffisantes pour construire certains objets complexes tels que les objets naturels comme les arbres. Le nombre de primitives d'un objet donné peut être très important ce qui peut avoir des conséquences négatives sur l'algorithme de rendu.

### 3.4.2. RBSO (REGULARIZED BOOLEAN SET OPERATION)

RBSO, acronyme de "*Regularized Boolean Set Operation*" signifiant opérations booléennes régularisées, est une technique de modélisation des objets du monde réel qui combine des objets simples pour former des objets plus complexes. La combinaison s'effectue à l'aide d'un ensemble d'opérations booléennes dites régularisées : union, intersection et union. Un opérateur booléen est dit régularisé s'il génère des objets solides, alors qu'un opérateur booléen ordinaire peut engendrer des objets non solides tels que les surfaces, les lignes, les points, etc. Pour distinguer les opérateurs régularisés des opérateurs booléens, on utilise deux notations différentes :  $\cup^*$ ,  $\cap^*$ ,  $-^*$  pour les opérations régularisées et  $\cup$ ,  $\cap$ ,  $-$  pour les relations ordinaires. Le tableau suivant (Tab2.2) montre une comparaison des résultats d'une opération d'intersection entre des opérations booléennes ordinaires et celles régularisées.

	Opération booléenne ordinaire	Opération booléenne régularisée
	$\bullet$	$\emptyset$
	$-$	$\emptyset$
		$\emptyset$
		

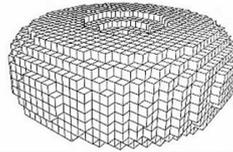
**Tab2.2** Comparaison des résultats d'une opération d'intersection entre des opérations booléennes ordinaires et celles régularisées

Les opérations booléennes régularisées sont appliquées sur l'intérieur des objets et négligent leurs enveloppes. Alors que les opérations booléennes ordinaires non régularisées tiennent en compte les bords des objets.

### 3.4.3. SPR (SPATIAL PARTITIONING REPRESENTATION)

SPR, acronyme de "*Spatial Partitioning Representation*" signifiant représentation par partition de l'espace, est une technique de modélisation dont le principe consiste à décomposer un solide en un

ensemble de solides élémentaires adjacents et ne se chevauchant pas. Tels solides élémentaires sont plus primitifs, et pas nécessairement de même type que le solide à décomposer. Les solides primitifs peuvent varier notamment en termes de type, de taille, de position, de paramètres et d'orientations, etc. [JARROUSH, *et al*, 2004]. Le principe de SPR constitue la base d'autres méthodes de modélisation de solides telles que BSP (Binary Spatial Partitioning), SOE (Spatial Occupancy Enumeration), Octrees.

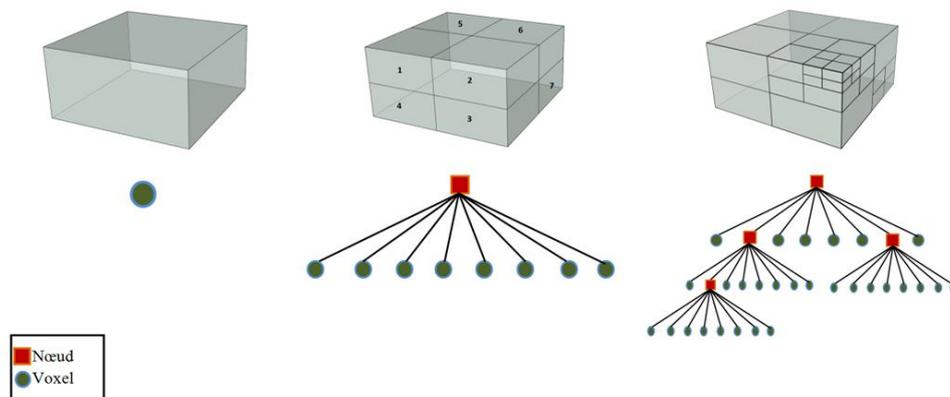


**Fig2.13.** Utilisation de la méthode SPR pour la construction d'un tore [VAN DAM, *et al*, 1991].

La figure (Fig2.13) montre un exemple d'application de la SPR en utilisant des voxels<sup>1</sup> pour la représentation d'un tore. SPR est avantageuse du point de vue de la simplicité, de la facilité de distinction de l'intérieur et l'extérieur des solides, du calcul approximatif du volume du solide, etc. Cependant, on peut atteindre facilement un nombre gigantesque de solides primitifs élémentaires pour la représentation d'un solide. On constate également que l'apparence visuelle des solides modélisés par cette méthode n'est pas très réaliste.

#### 3.4.4. OCTREES

Octrees, une variante du SPR, est une technique de modélisation basée sur une structure de type arbre dans laquelle chaque nœud peut avoir  $2^k$  enfants (dimension 2 : 4 fils, dimension 3 : 8 fils). C'est une description de la distribution des objets dans une scène à partir d'une subdivision récursive de l'espace. La figure (Fig2.14) suivante montre un exemple d'un arbre Octree correspondant à des volumes composés de voxels.



**Fig2.14.** Exemple d'un arbre Octree correspondant à des volumes composés de voxels

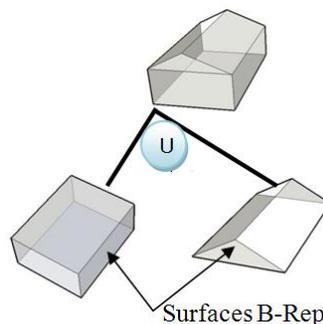
Les Octrees ont plusieurs avantages. Ils permettent de gérer des modèles 3D d'une manière efficace en représentant hiérarchiquement une scène 3D à travers un arbre. Les Octrees sont capables de représenter des objets volumiques à travers une répartition des voxels. Cette répartition offre une flexibilité de placement et de construction de volumes en utilisant un parcours simple. Néanmoins, cette approche souffre de quelques inconvénients. Les arbres utilisés pour la distribution des objets ne sont pas forcément équilibrés ce qui pourra alourdir leur parcours pour des scènes complexes. Cette méthode nécessite un espace de stockage important pour des scènes volumineuses ou complexes.

<sup>1</sup> Le voxel (contraction de « volumetric pixel ») est un pixel en 3D. C'est le plus petit élément de volume indivisible dans un système à trois dimensions.

Il existe d'autres méthodes de modélisation volumique telles que "*Sweep Representation*", "*Polyèdre et formule d'Euler*", "*Nonpolyhedral B-Rep*", etc. [VAN DAM, *et al*, 1991]. Le choix d'une méthode ou d'une autre se base sur plusieurs critères tels que le contexte de la modélisation, les spécificités des objets à modéliser, le niveau de détail attendu du modèle, la simplicité de la méthode, etc.

### 3.5. MODELISATION HYBRIDE : SURFACIQUE & VOLUMIQUE

Le principe consiste à combiner les deux approches surfacique (exemple B-Rep) et volumique (exemple CSG). L'objectif du couplage des deux méthodes est de cumuler leurs avantages et d'éviter au mieux leurs limitations respectives. La figure (Fig.2.15) suivante montre un exemple d'un bâtiment modélisé avec l'approche hybride B-Rep/CSG.



**Fig2.15.** Exemple de modélisation hybride B-Rep/CSG d'un bâtiment

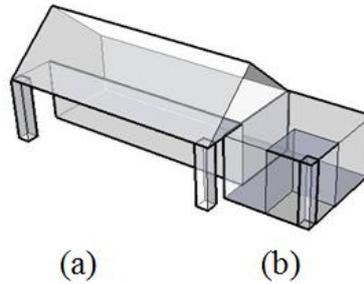
Cette méthode présente plusieurs avantages. Elle représente un objet à travers un arbre CSG et/ou avec des primitives B-Rep. Cela conduit à une souplesse de création d'objets complexes. Avec cette approche, les surfaces des solides sont définies d'une manière exacte. Cette méthode permet également le placement de contraintes géométriques et de connexion entre différentes primitives de l'objet à modéliser. Malgré le couplage des deux approches, cette approche possède quelques inconvénients. Elle nécessite un espace de stockage très important pour pouvoir stocker des informations sur les solides CSG, sur les arbres des opérations booléennes et sur les primitives B-Rep. Pour assurer l'homogénéité du modèle, il est nécessaire de maintenir la cohérence entre les deux représentations. Cette approche peut également alourdir la construction des primitives B-Rep car il faut parcourir l'arbre pour chercher des informations sur des primitives précises.

### 3.6. MODELISATION PARAMETRIQUE

Le principe de la modélisation paramétrique est la définition d'une entité par un ensemble de paramètres (quantitatifs et/ou qualitatifs) pouvant être modifiés facilement. Ces paramètres peuvent porter, par exemple, sur la longueur, les angles, le diamètre, etc. d'un objet géométrique quelconque. La modélisation paramétrique définit les relations entre les composants d'un objet qui peuvent être représentées par exemple par des équations mathématiques. Il est très important de distinguer la modélisation explicite de la modélisation paramétrique. Dans un modèle explicite, une géométrie est stockée avec l'entité correspondante (par exemple, un ensemble de polygones qui représentent le toit d'un bâtiment). Tandis que dans un modèle paramétrique, la géométrie est décrite implicitement à travers les paramètres de l'entité en question (par exemple, la forme d'un toit à deux versants symétriques, est donnée par les hauteurs des avant-toits et des crêtes en combinaison avec la largeur et la hauteur du bâtiment sur lequel il est construit) [GUERCKE, *et al*, 2009].

La figure (Fig.2.16) suivante montre une modélisation paramétrique très simplifiée d'un bâtiment. Soient  $P\{H, HC, HT, TT\}$  l'ensemble de paramètres du modèle avec H est la hauteur du bâtiment, HC

est la hauteur du corps du bâtiment, HT est la hauteur du toit et TT le type du toit. Sachant que H, HC et HT sont des paramètres quantitatifs alors que TT est un paramètre qualitatif.



**Fig2.16.** Exemple d'une modélisation paramétrique de deux bâtiments

Bâtiment a : P{4.5m, 3m, 1.5m, "Toit pente" }

Bâtiment b : P{3m, 3m, 0m, "Toit plat" }

Cette méthode est utilisée par quelques outils CAO tel qu'AutoCAD, MCAD (Mechanical Computer Aided Design), etc. Ces moteurs CAO utilisent généralement un arbre pour stocker les différentes opérations appliquées aux primitives telles que l'extrusion, le balayage, le lissage, etc.

La modélisation paramétrique présente quelques avantages. Avec cette méthode, il est possible d'attribuer quelques informations thématiques au modèle géométrique. Le modèle peut ainsi être bien décrit géométriquement et enrichi sémantiquement. Cependant, tous les paramètres quantitatifs et qualitatifs doivent être définis par le concepteur ce qui pourra être une tâche fastidieuse. Dans le cas d'un modèle complexe, l'arbre des opérations peut être compliqué et long.

## CONCLUSION

Depuis quelques années, la 3D a envahi notre vie quotidienne et elle s'est imposée de manière magistrale dans différents domaines: le cinéma (effets spéciaux et films 3D), la télévision (télévision 3D), les jeux vidéos, Internet (animations 3D), les villes 3D, etc. Son utilisation a apporté une valeur ajoutée en termes de visualisation, animation et manipulation d'objets 3D.

Avant d'être rendus dans un écran d'un ordinateur, les objets 3D composant un modèle doivent être modélisés. La modélisation consiste principalement à décrire formellement les apparences physiques de ces objets dont leur géométrie. En effet, plusieurs approches ont été proposées pour la modélisation géométrique tridimensionnelle. Quelque soit l'approche utilisée, les objets tridimensionnels sont généralement représentés avec des niveaux de précision différents.

Les applications géographiques ont accordé une attention particulière à la 3D pour la représentation des objets spatiaux. La 3D permet aussi bien plus de réalisme pour représenter des objets et phénomènes spatiaux, mais aussi des opérations d'analyse avancées sur de tels objets. Plusieurs méthodes de modélisation de données géo-référencées ont ainsi été proposées autour du concept de la 3D. Ces méthodes sont décrites dans le chapitre suivant.



## CHAPITRE.3 L'INFORMATION SPATIALE, CONCEPTS &amp; MODELISATION

Résumé.....	50
Introduction .....	50
<b>1. Définitions</b> .....	50
1.1. Information spatiale.....	50
1.2. Modes de représentation des données spatiales.....	51
<b>2. Modélisation géométrique des données spatiales (2D et 3D)</b> .....	51
2.1. Normes Internationales.....	52
2.1.1. ISO ( <i>International Organization for Standardization</i> ) .....	52
2.1.2. GML (Geography Markup Language) .....	53
2.1.3. IFC (Industry Foundation Classes).....	54
2.2. Modèles d'échange et de représentation des données spatiales .....	54
2.2.1. CityGML (City Geography Markup Language).....	54
2.2.2. COLLADA (Collaborative Design Activity) .....	56
2.2.3. KML (Keyhole Markup Language).....	57
2.3. Modèles issus de la recherche .....	58
2.3.1. 3D FDS (Formal Data Structure) .....	58
2.3.2. TEN (TEtrahedral Network) .....	59
2.3.3. SSM (Simplified Spatial Model).....	60
2.3.4. UDM (Urban Data Model).....	60
2.4. Bilan sur la modélisation géométrique .....	61
<b>3. Modélisation topologiques de données spatiales (2D et 3D)</b> .....	61
3.1. Définitions.....	61
3.2. Modèles et opérateurs topologiques .....	62
3.3. Matrice des neuf intersections (9I) .....	65
3.4. La topologie dans les outils spatiaux.....	70
Conclusion.....	73

« A l'espace en tant que milieu dans lequel se situe l'ensemble de nos perceptions, l'individu est situé à un point déterminé de l'espace et on a pu soutenir avec de bonnes raisons, que toutes ses sensations ont quelque chose de spatial »

(Emile DURKHEIM)

## RESUME

*Les domaines dans lesquels sont utilisés et exploités des données spatiales sont très nombreux. Les données spatiales ont un usage double. D'une part, elles permettent de localiser un objet ou un phénomène du monde réel sur le globe terrestre. Et d'autre part, elles associent des données thématiques aux objets et aux phénomènes correspondants. Mais avant d'être exploitées pour la visualisation, l'analyse spatiale, etc. ces données doivent être structurées et formalisées par des modèles géométriques, topologiques, etc. Pendant ces dernières années, la modélisation de données spatiales a constitué une problématique de recherche particulièrement active dans laquelle un ensemble de questions a été posé, un ensemble de concepts, de théories et de méthodes a été proposé et développé.*

## INTRODUCTION

De nos jours, l'information spatiale est omniprésente dans une variété de domaines et d'activités relatifs, de près ou de loin, à une localisation sur la terre. Sa mise à disposition à un public, de plus en plus large, a récemment proliféré dans la mesure où la majorité des informations utilisées dans nos activités quotidiennes sont des informations géo-référencées. Les statistiques montrent l'attention croissante accordée à l'information géographique et son importance auprès des communautés d'utilisateurs. Par exemple en 2003, environ 80 % des données administratives, opérationnelles ou stratégiques disponibles et utilisées au quotidien par les autorités nationales contenaient des informations géo-spatiales (adresses, références à des endroits physiques, etc.) [KOONTZ, 2003].

Malgré les nombreux travaux réalisés dans le domaine de la conception et la gestion des Systèmes d'Information (SI), plusieurs questions se sont posées autour de l'information spatiale en termes de modélisation, dimension, stockage, manipulation, etc. Plusieurs standards, approches et travaux de recherche ont traité la gestion de données géo-référencées en proposant une multitude de modèles pour structurer et formaliser les données spatiales en tenant compte des spécificités des objets à modéliser, des phénomènes spatiaux à étudier, des intérêts auprès des utilisateurs finaux, etc.

Dans ce chapitre, l'information spatiale est étudiée. La première partie propose quelques définitions ainsi que quelques concepts relatifs à l'information spatiale. Les deuxième et troisième parties s'intéressent respectivement à la modélisation géométrique et topologique de ces mêmes données spatiales.

## 1. DEFINITIONS

Dans cette partie, une brève définition de l'information spatiale est donnée en la situant par rapport aux concepts relatifs.

### 1.1. INFORMATION SPATIALE

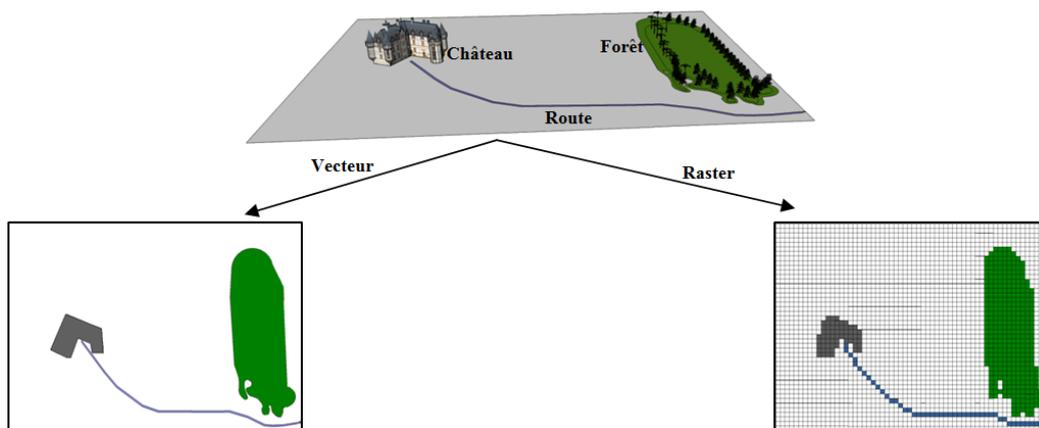
Le terme « *information* » signifie « *renseignements sur quelqu'un, quelque chose* » [ROBERT, 2008]. C'est une ou plusieurs données concernant un fait particulier. Il s'agit d'un « *ensemble de connaissances réunies sur un sujet déterminé* » [CNTRL, 2010]. Et dans le contexte informatique,

l'information signifie « *élément de connaissance susceptible d'être codé pour être conservé, traité ou communiqué* » [LAROUSSE, 2008].

Vu son utilisation interdisciplinaire, la définition de l'information spatiale varie selon le contexte dans lequel elle est étudiée. Mais toutes les définitions partagent le fait qu'une information spatiale est relative à une localisation sur le globe terrestre. Les données géographiques, comme toute donnée informatique gérée dans une base de données, sont généralement caractérisées par des données thématiques (nom, description, etc.), mais également par des informations spatiales (localisation, géométrie, relations spatiales, etc.). La structuration de ces données dépend de leur perception, modélisation, représentation, et également du mode ou format de représentation [SERVIGNE, *et al*, 2006]. L'information géographique consiste à représenter un phénomène ou un objet du monde réel ou virtuel ayant une localisation dans l'espace. Tel objet ou phénomène peut avoir ou non une existence physique. Les données spatiales proviennent des positions et caractéristiques physiques des objets situés sur ou à côté de la surface de la terre. Elles sont issues d'une variété de sources telles que les photographies aériennes, cartes numérisées, GPS (Global Positioning Systems), etc. [GALATI, 2006].

## 1.2. MODES DE REPRESENTATION DES DONNEES SPATIALES.

Il existe deux principaux modes de représentation de données spatiales : le mode vecteur et le mode raster (*Fig3.1*).



*Fig3.1.* Exemple d'un modèle vecteur et du modèle raster correspondant

En mode vecteur, chaque objet est défini indépendamment, de manière explicite, par des points (repérés par des coordonnées X, Y et Z), des lignes (formées de points) et des polygones (formés de lignes), voire des polyèdres (en trois dimensions). Le mode vecteur est basé sur les coordonnées [SERVIGNE, *et al*, 2006].

Le mode raster, quant à lui, se base sur des images provenant soit d'une numérisation d'une carte soit d'une image numérique (telles que les images satellites ou images aériennes, etc.) afin de représenter une zone spatiale. L'unité utilisée est le pixel. Les pixels formant une image sont répartis sur une grille régulière. Les formes géométriques telles que les lignes et les surfaces sont alors représentées par un enchaînement de pixels.

## 2. MODELISATION GEOMETRIQUE DES DONNEES SPATIALES (2D ET 3D)

La phase initiale et indispensable avant la mise en place de tout système d'information (SI) consiste en une modélisation des données. Une modélisation spatiale permet ainsi d'organiser des données spatialement référencées. Les travaux de recherche et de normalisation effectués par l'ISO

(International Organisation for Standardization) – ISO TC211 ou l'OGC (Open Geospatial Consortium) ont proposé divers modèles géométriques, topologiques et même sémantiques.

La modélisation géométrique de données spatiales porte sur les caractéristiques intrinsèques, généralement relatives aux formes des objets spatiaux qu'ils soient simples ou complexes, bidimensionnels ou tridimensionnels. Un ensemble de primitives géométriques est utilisé pour la description de ce type d'objets. Des nombreux modèles spatiaux sont proposés, chacun exprime une structure formelle des objets spatiaux.

## 2.1. NORMES INTERNATIONALES

### 2.1.1. ISO (*INTERNATIONAL ORGANIZATION FOR STANDARDIZATION*)

L'ISO est une organisation qui a défini jusqu'à présent plus de 18000 normes internationales dans des domaines assez diversifiés avec une moyenne de 1100 nouvelles normes chaque année.

Dans le domaine géo-spatial, il existe beaucoup de normes ISO adoptées pour la définition d'autres schémas spatiaux. Ces normes spécifiques sont développées par la TC211 (Technical Committee / Commission Technique) dont la mission principale consiste à définir un ensemble de standards structurés concernant des objets ou des phénomènes directement ou indirectement associés à une localisation dans l'espace. Ces standards définissent, pour des informations géographiques, l'ensemble de méthodes, outils et services pour la gestion de données, l'acquisition, le traitement, l'analyse, l'accès, la présentation et l'échange de données numériquement/électroniquement entre différents utilisateurs, systèmes et localisations. Parmi ces standards, l'ISO 19107 est un ensemble de schémas décrivant des données spatiales et faisant partie des documents de la série 191xx définis par la TC211.

- **ISO 19107**

L'ISO 19107, publié comme norme internationale en 2003, est un ensemble de modèles conceptuels spatiaux. Ces modèles sont créés pour décrire et manipuler les données spatiales 3D ainsi que leurs caractéristiques associées relatives à un phénomène géographique donné.

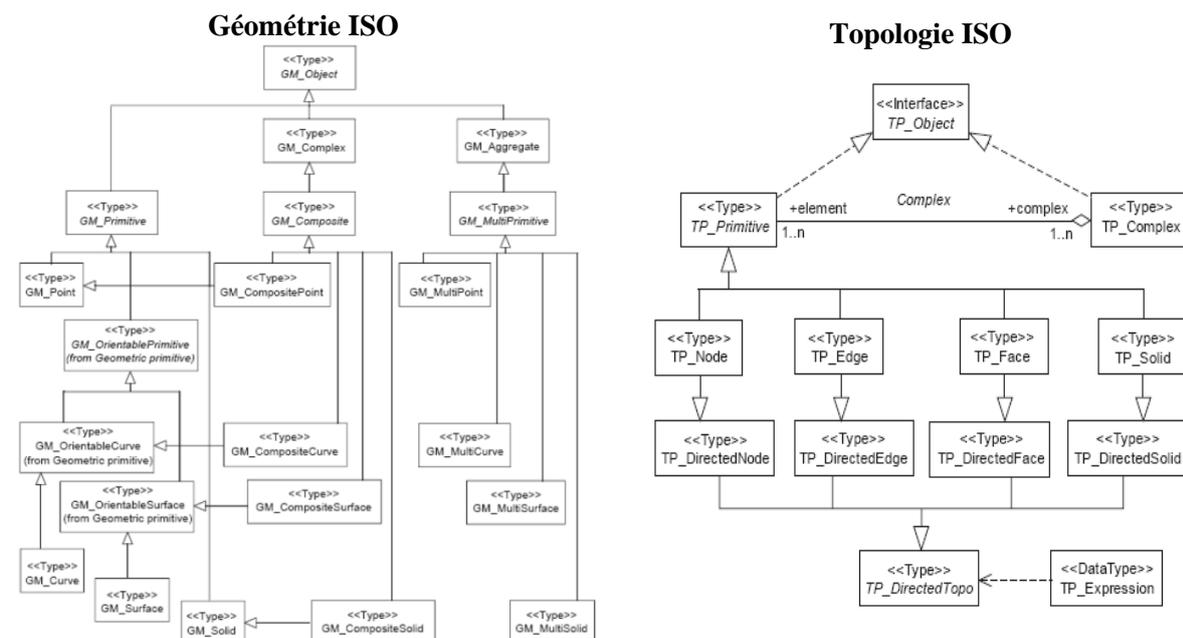


Fig3.2. Schémas conceptuel géométrique (à gauche) et topologique (à droite) de l'ISO 19107

Le concept "feature" (ou "élément" en français) est utilisé par l'ISO 19107 pour représenter de manière abstraite des objets du monde réel. Le concept d'"élément géographique" est utilisé pour représenter tout objet ayant une localisation dans l'espace. L'ISO 19107 définit un ensemble de primitives géométriques dérivées de la classe GM\_OBJECT (*GM\_POINT*, *GM\_CURVE*, *GM\_SURFACE* et *GM\_SOLID*, etc.) ainsi qu'un ensemble de primitives topologiques dérivées de la classe TP\_OBJECT (*TP\_NODE*, *TP\_EDGE*, *TP\_FACE*, *TP\_SOLID*, etc.). Les GM\_OBJECT sont utilisés pour décrire la géométrie d'un objet donné. Tandis que, les TP\_OBJECT décrivent les relations de connectivité entre les GM\_OBJECTS.

L'ISO 19107 propose un diagramme de classes géométriques détaillé dans lequel il décrit les différentes primitives géométriques ainsi que les relations entre elles. La figure (Fig3.3) suivante reprend cette description.

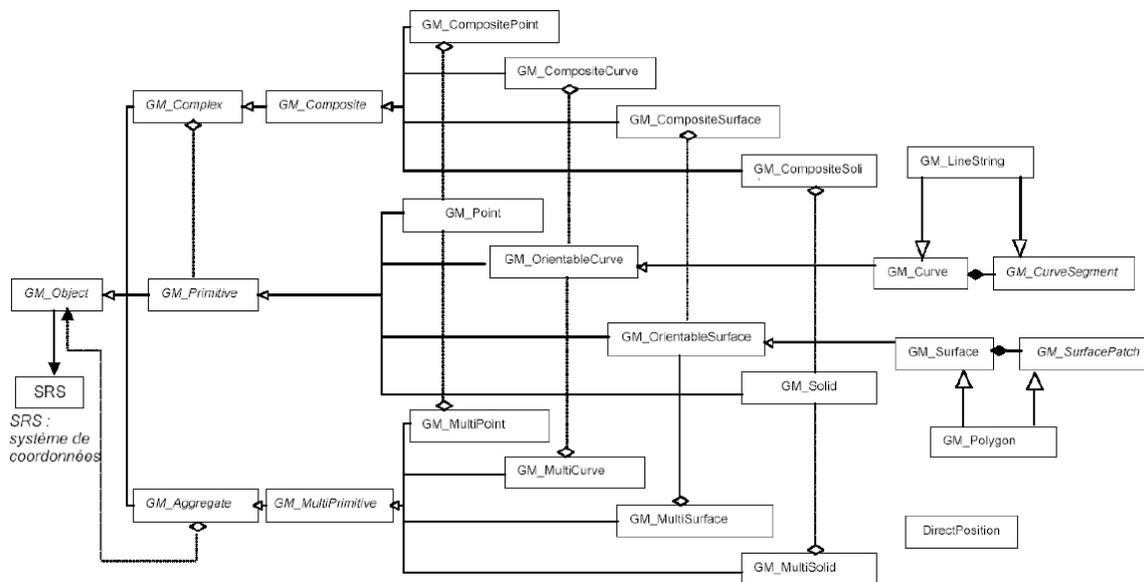


Fig3.3. Diagramme de classes géométriques de l'ISO 19107

La description géométrique des objets du monde réel porte sur leurs positions dans l'espace, leurs orientations, leurs formes, etc. Tandis que, la description topologique porte sur les relations de connectivité entre différents éléments d'un modèle spatial 3D. La géométrie d'un objet donné peut changer si on modifie le système de coordonnées, alors que la topologie reste invariante si l'espace varie notamment dans le cas d'un changement de système de coordonnées.

### 2.1.2. GML (GEOGRAPHY MARKUP LANGUAGE)

GML, acronyme de "Geography Markup Language", est un langage de format XML, conçu pour la description et l'échange de données spatiales. C'est une spécification OGC (Open Geospatial Consortium) supportant des données spatiales et non spatiales et implémentant les concepts de l'ISO 19100. GML peut servir de langage de modélisation de données spatiales tant pour des systèmes géographiques que pour des échanges de données sur Internet. Un document GML au format XML contient deux parties principales : le schéma qui décrit le document et les instances XML respectant la structure définie dans le schéma et contenant des données. Un document GML est donc décrit par un schéma GML. Cela aide des utilisateurs et des développeurs à décrire des ensembles de données géographiques contenant des *points*, *lignes* et *polygones*.

Le standard GML définit la syntaxe du schéma XML, les mécanismes et les conventions pour [OGC, 2007] :

- Fournir une structure pour la description des schémas géo-spatiaux pour l'échange et le stockage d'informations géographiques dans un format XML.
- Définir une description des applications géo-spatiales pour des domaines spécialisés.
- Permettre la création et la maintenance des schémas des applications géographiques.
- Permettre le stockage et l'échange des schémas et des données entre des applications géographiques.

Augmenter la capacité des organisations pour le partage de schémas géographiques et les données qu'ils décrivent.

### **2.1.3. IFC (INDUSTRY FOUNDATION CLASSES)**

IFC, acronyme de "*Industry Foundation Classes*", est un format normalisé en 1997 pour l'échange de données concernant les bâtiments. Il a été conçu afin de mettre en place une structure de données commune assurant l'interopérabilité entre des logiciels dans le domaine de la construction. C'est un format de description, d'échange et de partage de données utilisées dans le secteur des bâtiments. L'échange s'effectue à l'aide des fichiers ISO 10303-21 dits fichiers STEP-21 ou à travers des fichiers XML nommés ifcXML. La spécification IFC est développée et maintenue par *bSI* (building SMART International) et *IAI* (International Alliance for Interoperability).

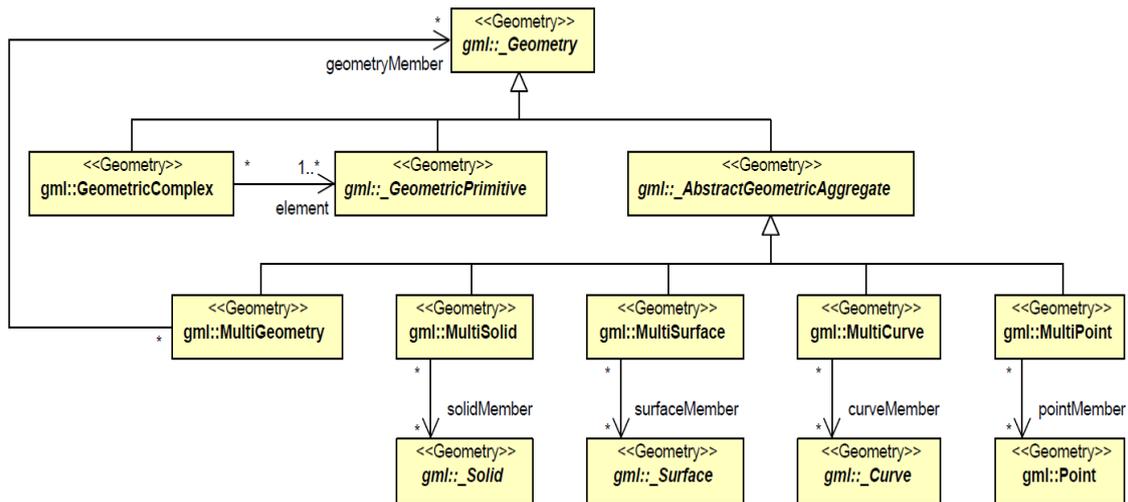
L'IFC est un ensemble de spécifications normalisées pour le BIM (Building Information Models). Il fournit un support pour :

- L'échange de données entre des applications appartenant au secteur AEC/FM (Architecture, Engineering and Construction /Facility Management).
- La proposition d'un modèle pour la description de données spatiales, des bâtiments, des composants des bâtiments, etc.
- La représentation des formes géométriques des objets de type bâtiment ou relatifs aux bâtiments.
- La représentation des relations entre tels objets.
- L'ajout de données attributaires, la classification, accès aux bibliothèques externes, etc.

## **2.2. MODELES D'ECHANGE ET DE REPRESENTATION DES DONNEES SPATIALES**

### **2.2.1. CITYGML (CITY GEOGRAPHY MARKUP LANGUAGE)**

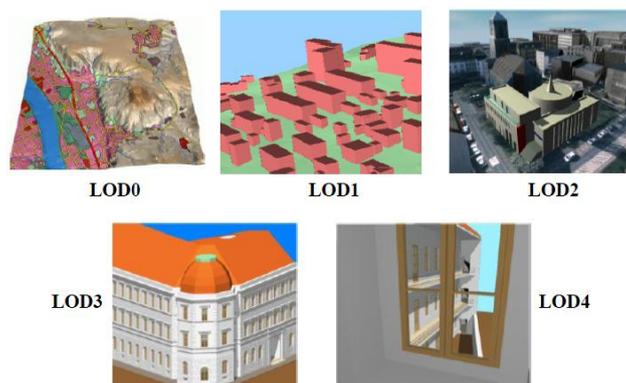
CityGML est un ensemble de modèles spatiaux pour la représentation des objets urbains. Comme son nom l'indique, CityGML se base sur les spécifications du standard GML pour la description et la manipulation d'objets urbains. C'est une spécification OGC dont la première version est apparue en 2007. Il est destiné à être un standard ouvert pour le stockage et l'échange de modèles urbains virtuels 3D. Il inclut des propriétés inspirées des objets topographiques telles que les propriétés géométriques, topologiques, sémantiques et d'apparence. Il est basé sur les spécifications de GML3 dont le modèle géométrique définit un ensemble de primitives géométriques pouvant être combinées pour former des géométries composées, complexes ou des agrégations. Pour chaque dimension, il existe une primitive géométrique : un objet 0D est un *Point*, un objet 1D est une *\_Curve*, un objet 2D est une *\_Surface* et un objet 3D est un *\_Solid* [OGC, 2008]. Comme CityGML peut être appliqué dans divers domaines comme la planification urbaine, la gestion des catastrophes naturelles, le tourisme, etc. un ensemble de modèles de bâtiments, d'apparence, de MNT (Modèle Numérique de Terrain), de transport sont alors proposés. La figure (*Fig3.4*) reprend le modèle géométrique proposé par [OGC, 2008].



**Fig3.4.** Modèle géométrique : primitives complexes et agrégats [OGC, 2008]

Le modèle géométrique présenté dans la figure (Fig3.5) décrit les différentes primitives géométriques ainsi que leurs combinaisons pour former des primitives complexes et des agrégats de type *MultiSolid*, *MultiSurface*, etc.

Outre les modèles géométriques et topologiques, CityGML propose un ensemble de modèles thématiques tel que le modèle de bâtiment décrit dans la figure (Fig3.5) ci-dessous. Actuellement, CityGML constitue le modèle spatial le plus complet dans le domaine des modèles 3D urbains. D'une part, il se base sur des spécifications GML utilisant des concepts ISO, et d'autre part, il inclut une multitude de modèles géométriques, topologiques et thématiques. Il représente des phénomènes géographiques avec des échelles allant du MNT jusqu'aux détails internes d'un bâtiment. En effet, CityGML propose un ensemble de LoD (Level of Detail / niveaux de détail) pour optimiser la gestion des objets. Ainsi, le LoD0 correspond à un MNT 2.5D sur lequel une image aérienne ou une carte peuvent être plaquées. Le LoD1, connu en tant que "modèle des blocs", représente les bâtiments à travers un ensemble de blocs avec des toits plats, alors qu'un bâtiment du LoD2 contient des toits ayant différentes structures et des surfaces thématiquement variées. Le LoD3 fournit un modèle architectural ayant des façades et des toits détaillés, des balcons, des baies et des projections. Des textures hautes-résolution peuvent être plaquées aux différentes structures du LoD3. Le LoD4 complète le modèle LoD3 en permettant d'y ajouter des détails internes tels que des chambres, des portes et fenêtres, des escaliers, etc.



**Fig3.5.** Illustration des différents niveaux de détails de CityGML [OGC, 2008]

La figure (Fig3.6) ci-dessous reprend le modèle du bâtiment de CityGML.

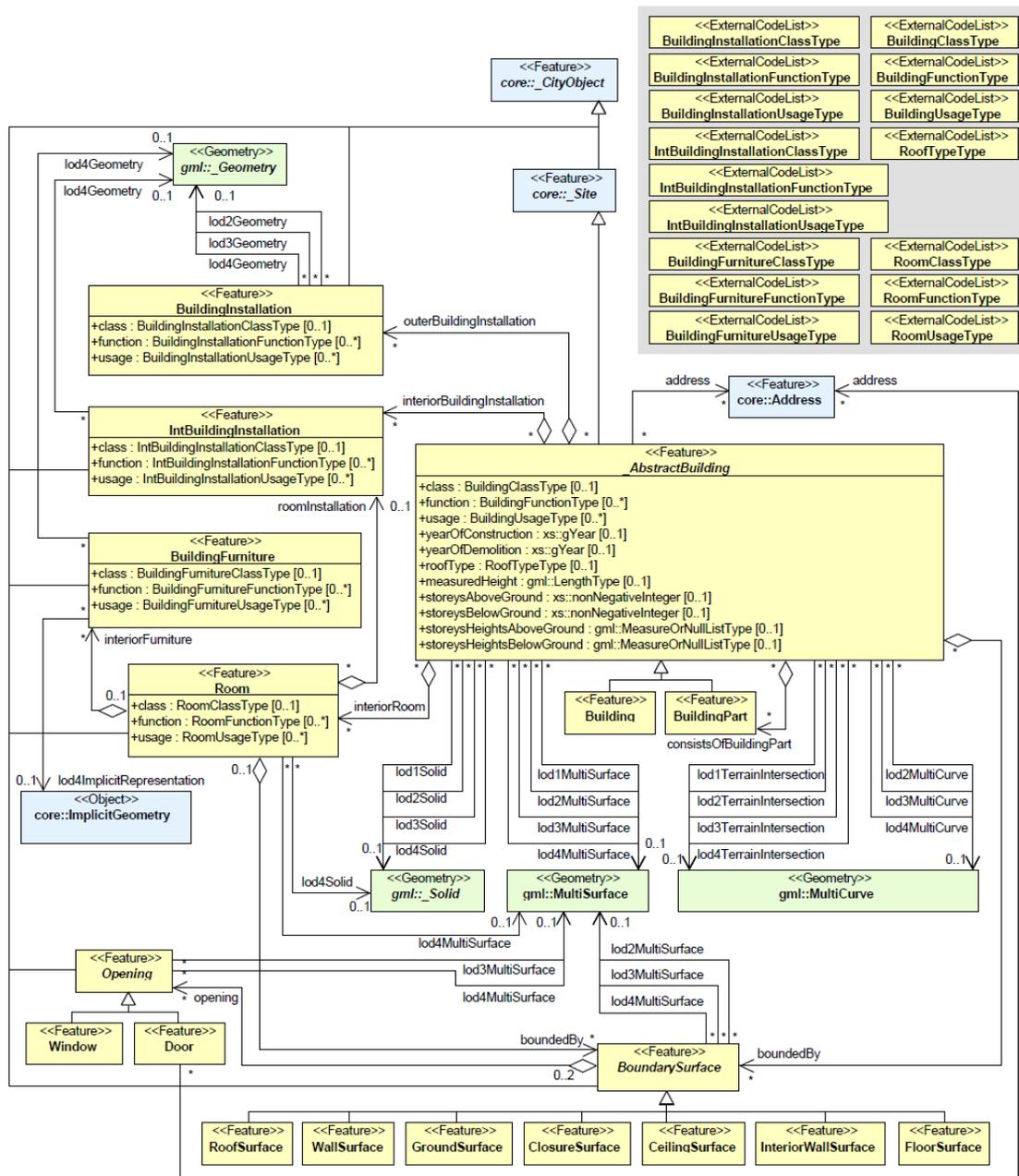
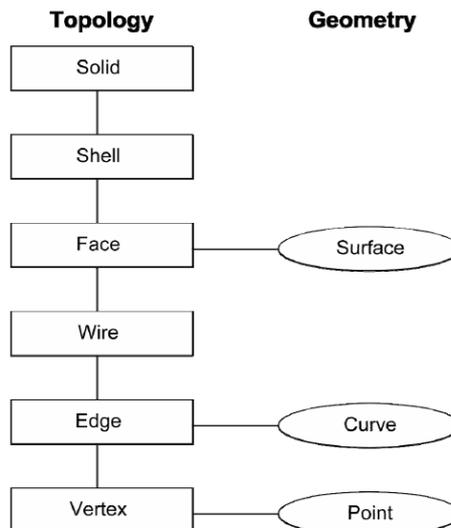


Fig3.6. Modèle du bâtiment de CityGML [OGC, 2008]

### 2.2.2. COLLADA (COLLABORATIVE DESIGN ACTIVITY)

"Collaborative Design Activity" abrégé en *COLLADA*, signifiant Activité de Conception Collaborative, est un modèle de description et d'échange de données 3D entre des applications interactives. COLLADA utilise le format XML pour représenter des objets 3D à travers des polygones, matériaux et shaders et quelques animations [COLLADA, 2008]. Les fichiers COLLADA sont utilisés pour échanger des données sur les objets 3D telles que les transformations (translation, rotation, mise à l'échelle, etc.), matériaux et textures, animations, données de scène (caméra, lumière, etc.), etc. Du point de vue géométrique, COLLADA utilise plusieurs types de primitives telles que les lignes, les meshes (maillages), les polygones, les splines, les triangles, etc.

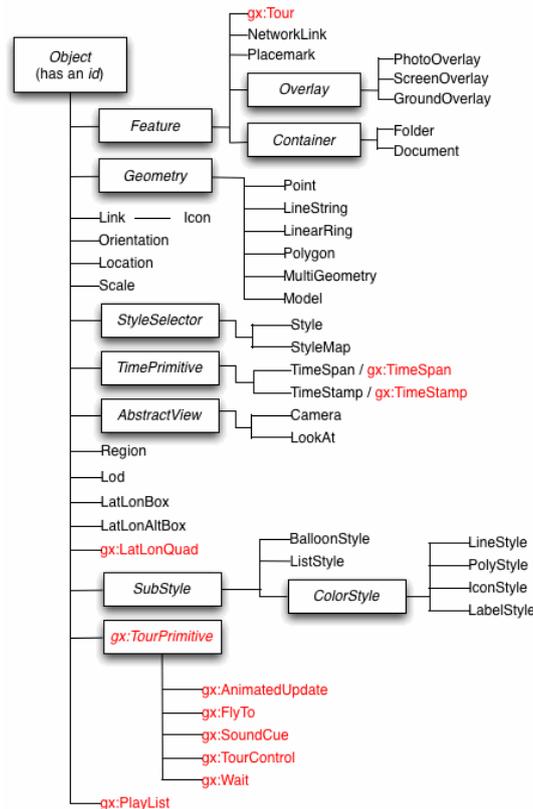


**Fig3.7.** Schéma géométrique/topologique [COLLADA, 2008]

Parmi les primitives géométriques particulières proposées par COLLADA on retrouve la primitive *Wire* (composée d'un ensemble d'arcs) et la primitives *Shell* (composée d'un ensemble de faces). COLLADA utilise la représentation par frontières dans laquelle la topologie et la géométrie sont mixées. En effet, un *nœud* est représenté par un *point*, une *arête* est représentée par un *arc*, et une *face* est représentée par une *surface*.

### 2.2.3. KML (KEYHOLE MARKUP LANGUAGE)

KML, acronyme de "*Keyhole Markup Language*", est un format XML de description et d'échange de données spatiales entre des logiciels issus de la société Google tels que *Google Earth*, *Google Maps*, etc. et d'autres applications telles que *World Wind*. Les fichiers KML utilisent le format COLLADA pour la modélisation et le stockage des caractéristiques géographiques des objets du monde réel tels que les primitives géométriques (points, lignes, polygones, etc.), des modèles 3D, des images, etc.



**Fig3.8.** Structure du schéma KML

KML est une spécification OGC mise en place pour accomplir divers objectifs :

- Mettre en place un seul langage standard international pour l'expression des annotations et des visualisations géographiques des cartes dans les navigateurs Web (2D) et dans les navigateurs du globe terrestre (3D).
- Aligner KML avec les autres standards pour assurer une meilleure exploitation et interopérabilité entre différentes implémentations des navigateurs du globe terrestre.
- Collaborer avec l'OGC pour assurer la conformité de KML.

La figure (Fig3.8) montre les différents composants du schéma KML. Il inclut les différentes propriétés utilisées par KML. Ce schéma contient des caractéristiques thématiques, géométriques (*Point*, *LineString*, *LinRing*, *Polygon*, etc.), localisation, échelle, orientation, LoD, etc.

## 2.3. MODELES ISSUS DE LA RECHERCHE

### 2.3.1. 3D FDS (FORMAL DATA STRUCTURE)

Le FDS ou Structure Formelle de Données est un modèle de données proposé par [MOLENAAR, 1990] pour la représentation de données spatiales. Il se base sur quatre primitives géométriques : *Point*, *Ligne*, *Surface* et *Volume*. Dans ce modèle, la topologie dépend de la géométrie du fait que les relations topologiques sont exprimées à travers les liens entre les primitives géométriques. ISO s'inspire de ce modèle mais il recommande une indépendance totale entre le modèle géométrique et topologique [RAMOS, 2002]. La figure (Fig3.9) montre les primitives géométriques utilisées par ce modèle ainsi que les relations topologiques entre elles telles que « *Is In* », « *Is On* », « *Part Of* », etc.

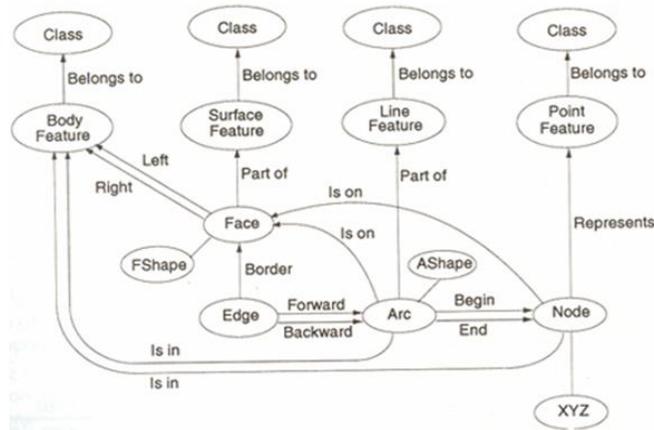


Fig3.9. Modèle 3D FDS [MOLENAAR, 1990]

3D FDS est un modèle vecteur comportant des informations géométriques, topologiques et pouvant contenir des informations sémantiques. Ce modèle est structuré en trois niveaux principaux : le niveau géométrique, objet et thématique. Le niveau géométrique consiste à représenter des informations à propos de la position, la forme, la taille ainsi que les relations entre les objets (topologie). Les trois primitives, *nœud*, *arc* et *face*, introduites dans ce modèle constituent les éléments basiques pour former les objets. Le niveau objet est matérialisé par quatre types d'objets : *point*, *ligne*, *surface* et *volume*. Le dernier niveau comporte les différentes classes thématiques liées aux objets [ZLATANOVA, 1996].

### 2.3.2. TEN (TETRAHEDRAL NETWORK)

TEN, acronyme de "*TETrahedral Network*", est une approche de modélisation de volumes proposée par [PILOUK, 1996]. Elle se base sur quatre types de primitives : *tétraèdre*, *triangle*, *arc* et *nœud*. Un volume est composé de tétraèdres, une surface est composée de triangles, une ligne est composée d'arcs et un point est composé de nœuds. Ces différentes primitives ainsi que les relations entre elles sont décrites dans le modèle présenté dans la figure (Fig3.10) proposé par [QINGQUAN, et al, 2000].

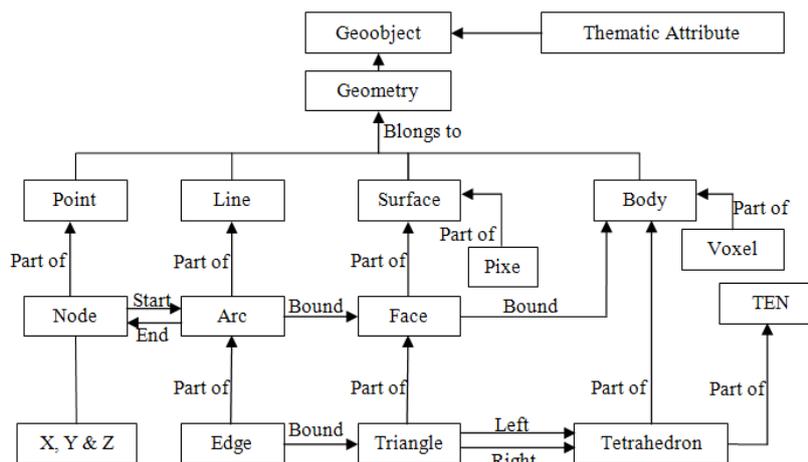


FIG3.10. Modèle 3D TEN intégré [QINGQUAN, et al, 2000]

Du fait que l'espace est complètement subdivisé en tétraèdres, cette subdivision n'est pas adaptée pour des objets artificiels [ZLATANOVA, 1996]. Ceci est particulièrement vrai pour les intérieurs des objets (les bâtiments, par exemple) ainsi que les espaces ouverts.

### 2.3.3. SSM (SIMPLIFIED SPATIAL MODEL)

SSM acronyme de "*Simplified Spatial Model*", signifiant Modèle Spatial Simplifié, été conçu pour servir des applications orientées-Web par plusieurs types de requêtes de visualisation. Ce modèle utilise quatre objets géométriques : *point*, *ligne*, *surface* et *volume*. Mais, il emploie uniquement deux primitives géométriques : *nœud* et *face*.

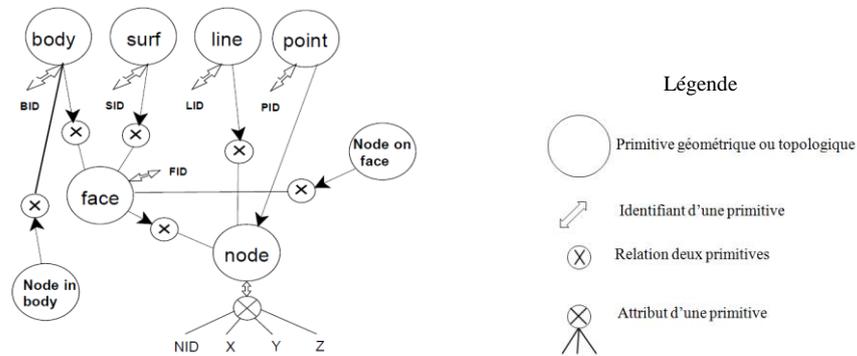


Fig3.11. Simplified Spatial Model [ZLATANOVA, 2000]

La raison de ne pas stocker explicitement les *arcs* est due au simple fait que la relation unique *arc/face* n'a plus de sens car le même *arc* peut être partagé par plus de deux *faces* [ZLATANOVA, et al, 2002].

### 2.3.4. UDM (URBAN DATA MODEL)

UDM acronyme de "*Urban Data Model*" signifiant modèle de données urbain, représente la géométrie d'une surface ou d'un volume par un ensemble de *faces* planes et convexes [COORS, 2002]. Chaque *face* est définie par un ensemble de *nœuds*. Deux faces convexes et planes sont adjacentes si elles partagent au minimum deux *nœuds*. L'orientation d'une face donnée est stockée implicitement. Le schéma conceptuel de ce modèle est décrit dans la figure (Fig3.12) ci-dessous.

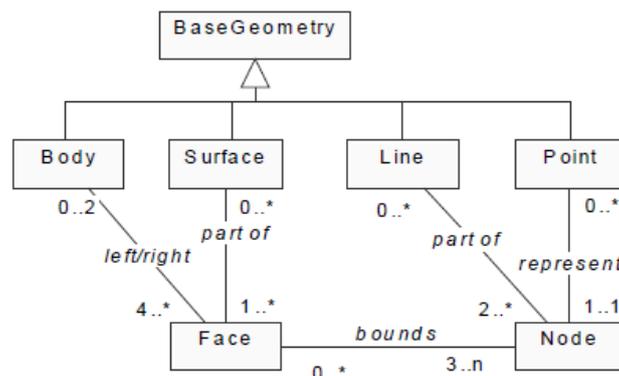


Fig3.12. UDM (Urban Data Model)

Dans la représentation relationnelle de l'UDM, chaque face possédant plus de trois nœuds est décomposée en triangles [ZLATANOVA, et al, 2002]. La primitive *arc* n'est pas stockée explicitement car elle peut facilement être interprétée implicitement en lui associant deux nœuds. Dans le cas des objets complexes, la triangulation de toutes les *faces* les composant engendre un volume important de données.

## 2.4. BILAN SUR LA MODELISATION GEOMETRIQUE

Chacun des modèles mentionnés ci-dessus traduit une perception des données spatiales dans un contexte donnée (UDM et CityGML pour un contexte urbain, KML pour la description et l'échange des données spatiales, etc.).

Plusieurs interprétations peuvent être ainsi dégagées :

- Chaque modèle spatial propose une représentation de la réalité spatiale à travers un ensemble de primitives géométriques pouvant être associées à des primitives topologiques, voire attributaires (tel que CityGML).
- Quelques modèles spatiaux s'inspirent des autres et utilisent des primitives communes comme CityGML qui s'appuie sur des primitives GML, ISO qui s'inspire du 3D FDS, etc.
- Un modèle spatial ne peut pas être universel. Pour ce faire, chaque modèle est appliqué généralement pour des données spatiales particulières.

Un tableau de comparaison des différents modèles spatiaux est décrit dans l'annexe (page 220) [YANBING, *et al*, 2007].

## 3. MODELISATION TOPOLOGIQUES DE DONNEES SPATIALES (2D ET 3D)

La modélisation géométrique s'intéresse aux formes des objets du monde réel ainsi que leurs tailles, leurs localisations dans l'espace, etc. Tandis que, la modélisation topologique complète la géométrie par l'étude des différentes relations de connectivité et d'inclusion entre des objets statiques ou en déformation. En effet, un objet du monde réel peut avoir des relations permanentes avec d'autres objets dont l'étude peut révéler des interprétations, des calculs, des décisions, etc.

### 3.1. DEFINITIONS

La définition de la topologie varie selon le contexte dans lequel elle est étudiée. Etymologiquement, le terme "*topologie*" est la concaténation des deux mots grecs "*topo*" et "*logos*" signifiant respectivement "*lieu*" et "*étude*". En se référant à la géométrie, la topologie est définie comme « *une partie de la géométrie qui étudie les propriétés qualitatives et les positions relatives des êtres géométriques indépendamment de leur forme et leur grandeur* » [ROBERT, 2008]. Mathématiquement, la topologie est définie comme « *L'étude mathématique des propriétés qui sont préservées grâce à des déformations, torsions et étirements d'objets. Le déchirement, cependant, n'est pas autorisé. Un cercle est topologiquement équivalent à une ellipse (dans lequel il peut être déformé par étirement) et une sphère est l'équivalent d'un ellipsoïde* » [MATHWORLD, 2010]. Et dans le contexte spatial, la topologie peut être définie comme « *Les différents objets géographiques interconnectés ce qui constitue un réseau en termes d'adjacence et de connectivité dans l'espace relatif, défini par les relations entre les objets eux mêmes*» [JIANG, *et al*, 2006].

En parlant de la topologie, il est très important de définir la notion d'espace topologique. Un espace topologique est mathématiquement défini comme suit : « *Les espaces topologiques permettent de définir dans un contexte très général des concepts comme la convergence, la continuité et la connexité. Ces concepts apparaissent dans presque toutes les branches des mathématiques, ils sont donc centraux dans la vision moderne des mathématiques. La branche des mathématiques qui étudie ces espaces s'appelle la topologie* » [TECHNOSCIENCE, 2010].

Dans la littérature, à chaque dimension (0D, 1D, 2D ou 3D) correspond une primitive géométrique et une primitive topologique. Le tableau (Tab3.1) reprend ces éléments.

Dimension de la primitive	Primitive géométrique	Primitive topologique
0	Point	Nœud
1	Ligne	Arc
2	Surface	Face
3	Solide	Volume

**Tab3.1** Primitives géométrique et les primitives topologiques associées

L'étude des relations topologiques est très intéressante dans les outils et les applications spatiales. Ces relations offrent des réponses à des questions comme « *quels sont les bâtiments ayant une façade mitoyenne avec un bâtiment donné* » ou « *pour creuser un tunnel à un endroit donné, quelles sont les infrastructures qui peuvent être touchées* », etc.

### 3.2. MODELES ET OPERATEURS TOPOLOGIQUES

Alors que la géométrie repose généralement sur une étude quantitative de l'espace (coordonnées, mesures, etc.), la topologie s'intéresse plutôt à l'étude qualitative de cet espace (relations spatiales entre et inter objets spatiaux). L'étude topologique d'un espace donné dépend principalement du contexte de cette étude (par exemple, l'analyse des relations de voisinage pour étudier les zones présentant un risque d'inondation).

Depuis le début des années 1990, plusieurs modèles topologiques [MOLENAAR, 1990], [PILOUK, 1996], [DE LA LOSA, 2000], [PFUND, 2001], etc. ont été proposés pour structurer les propriétés topologiques ainsi que les relations spatiales entre les objets spatiaux. La structuration de la topologie est utilisée essentiellement pour assurer l'intégrité de données, elle décrit les géométries partagées entre les objets spatiaux [HOEL, *et al*, 2003]. La formalisation des relations topologiques à travers un ensemble de schémas conceptuels ou des métadonnées s'avère être très utile pour effectuer des requêtes topologiques et des analyses spatiales sur des données géo-référencées. Les structures topologiques possèdent généralement des propriétés utiles telles que la partition de l'espace (où les primitives sont connectées mais ne se chevauchent pas), l'orientation, la connectivité, les caractéristiques d'Euler, les relations d'incidence et d'adjacence, etc. [WEI, *et al*, 1998] cité dans [LACHANCE, *et al*, 2006].

[CLEMENTINI, *et al*, 2008] proposent une classification des relations spatiales selon trois catégories géométriques : topologiques, projectives, et métriques. Elles sont basées respectivement sur les propriétés de l'espace topologique, projectif, et euclidien. Comme les relations topologiques, les relations projectives sont de nature qualitative, car elles n'ont pas besoin de mesures précises pour être expliquées. De plus, les relations projectives sont plus précises que les relations topologiques et elles peuvent servir de base pour décrire les relations qui ne sont pas décrites par la topologie (telles que les relations « *à droite de* », « *en avant de* », « *entre* », « *le long de* », « *entouré par* », « *devant* », « *arrière* », « *au nord de* », « *à l'est de* », etc.) [CLEMENTINI, *et al*, 2008].

Comme nous l'avons vu dans le chapitre précédent, ISO 19107 est une norme internationale proposant une multitude de schémas conceptuels pour la description et la manipulation des caractéristiques spatiales des éléments géographiques. Un *élément* est une abstraction d'un objet réel, il est dit géographique s'il est associé à une localisation relative dans la terre. Le standard ISO 19107 décrit à la fois les caractéristiques géométriques d'un objet spatial ainsi que les spécifications topologiques associées. Il distingue des objets géométriques (*GM\_OBJECT*) et des objets topologiques (*TP\_OBJECT*) tel que c'est décrit dans la figure (Fig3.2) ci-dessus. Dans le contexte de l'information géographique, la topologie est employée pour décrire la connectivité d'un ensemble d'objets n-dimensionnel. La topologie fournit des informations sur la connectivité des primitives géométriques qui peuvent être interprétées de la géométrie sous-jacente. Les opérateurs spatiaux sont matérialisés

par des fonctions pour utiliser, interroger, créer, modifier ou supprimer des objets spatiaux. ISO 19107 définit une taxonomie pour standardiser la définition et l'implémentation de ces opérateurs. Les principaux opérateurs sont "equals", "disjoint", "intersects", "touches", "crosses", "within", "contains" et "overlaps".

Les spécifications de ce standard sont utilisées par d'autres schémas standardisés qui portent sur l'information géographique tels que GML, CityGML, etc. Néanmoins, ce modèle de données montre quelques inconvénients : i) La taille des données est considérable par le simple fait que la structure contient des aspects très variés du modèle 3D. ii) La complexité du modèle conduit à une dégradation de la performance des opérations géométriques et topologiques. Telles inconvénients résultent en une implémentation lourde et difficile de l'ISO 19107 [KIM, *et al*, 2009].

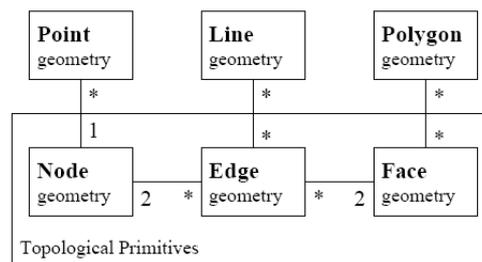


Fig3.13. Modèle topologique générique [ISO, 1999]

[HOEL, *et al*, 2003] propose une extension du modèle topologique générique de [ISO, 1999] décrite dans la figure (Fig3.14).

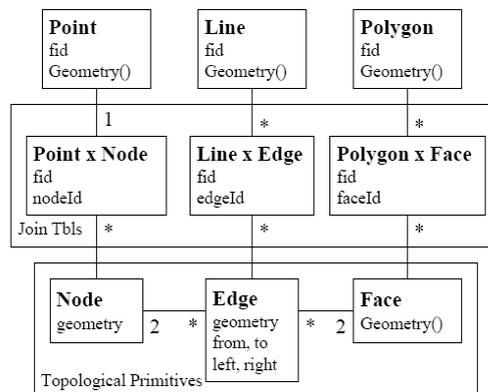


Fig3.14. Modèle topologique relationnel [HOEL, *et al*, 2003]

Ce modèle utilise les contraintes d'intégrité référentielle pour décrire les topologies. Les tables de jointure sont employées pour supporter les relations plusieurs-plusieurs entre les éléments (*features*) et leurs primitives topologiques associées. Ce modèle permet un accès direct aux primitives topologiques (*nœuds*, *arcs* et *faces*) du modèle topologique logique. Les relations topologiques sont explicitement décrites dans les tables de jointure. Cependant, ce modèle révèle quelques désavantages tels que la performance des requêtes typiques, la maintenance de l'intégrité sémantique du modèle, la performance et la complexité des mises à jour [HOEL, *et al*, 2003].

Comme [LACHANCE, *et al*, 2006] définissent une structure topologique adaptée à la modélisation par tétraèdres, [KHUAN, *et al*, 2007] et [KHUAN, *et al*, 2008.a] proposent une structure pour traiter des données 3D appelée '*Polyhedron*' ainsi qu'un ensemble de rôles ou règles pour constituer de tels polyèdres. Pour chaque géométrie d'un polyèdre on associe une topologie. Pour chaque polyèdre, on ne définit que les objets de dimensions inférieures. De ce fait, les ID des faces sont enregistrées

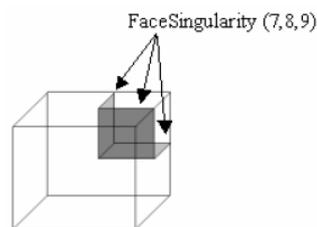
comme étant les données d'entrée pour le polyèdre. La topologie d'un polyèdre est définie alors comme :

`POLYHEDRON (FaceInfo(6,3),Face(1,2,3,4,5,6),FaceSingularity(7,8,9))`

Avec

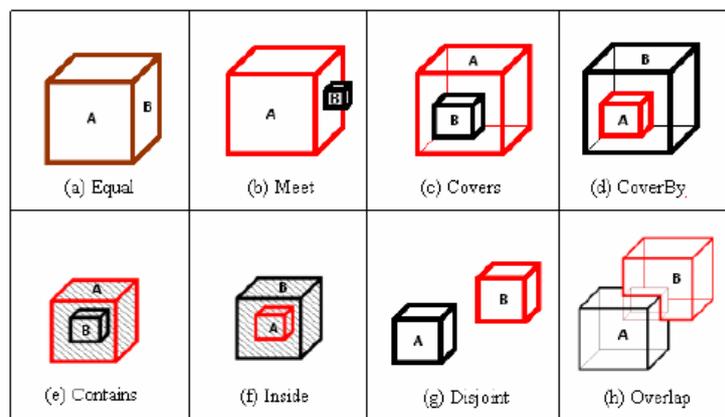
- FaceInfo(6,3) dénote 6 polygones et 3 singularités du polyèdre,
- Face(1,2,3,4,5,6) dénote tous les ID des faces qui composent le polyèdre,
- FaceSingularity(7,8,9) dénote toutes singularités d'un polygone faisant partie du polyèdre. Ces types de faces font références aux faces appartenant à un autre polyèdre qui partage une partie commune avec le polyèdre en question.

Un exemple de deux polyèdres qui se chevauchent est décrit dans la figure (Fig3.15).



**Fig3.15.** Structure d'un polyèdre pour des données topologiques [KHUAN, *et al*, 2008.a]

Les opérations topologiques sont basées sur le modèle des 4-intersections. Ces opérations sont : *Overlap*, *Meet*, *Disjoint*, *Inside*, *Covers*, *CoveredBy*, *Contain*, and *Equal*. Ces opérations sont décrites dans la figure (Fig3.16) ci-dessous.



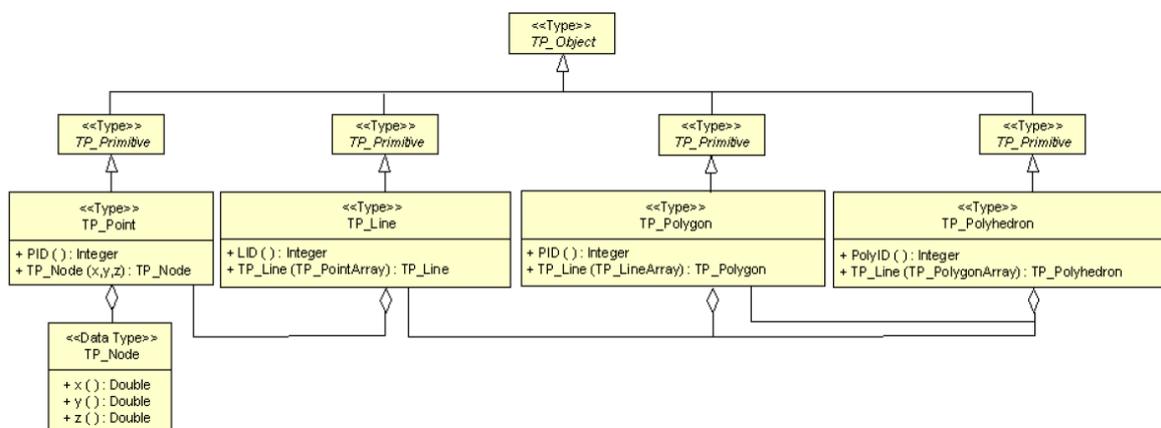
**Fig3.16.** Relation volume-volume [ZLATANOVA, 2000] cité dans [KHUAN, *et al*, 2008.a]

[KHUAN, *et al*, 2008.a] regroupe ces opérations dans un Tableau (Tab3.2) dont les lignes sont les différentes opérations citées ci-dessus et les colonnes sont les différentes possibilités de l'endroit d'un objet par rapport à un autre « *Inside* », « *Outside* » et « *Touch* ».

Opération topologique 3D	Inside	Outside	Touch
Overlap	X	X	✓
Meet	X	✓	✓
Disjoint	✓	X	✓
Inside	✓	X	✓
Covers	✓	X	X
CoveredBy	✓	X	X
Contain	X	✓	X
Equal	✓	✓	✓

**Tab3.2** Différentes conditions d'opérations topologiques [KHUAN, *et al*, 2008.a]

[KHUAN, *et al*, 2008.b] définissent un modèle conceptuel des primitives géométriques *TP\_Point*, *TP\_Line*, *TP\_Polygon* et *TP\_Polyhedron* comme indiqué dans la figure (Fig3.17) ci-dessous.



**Fig3.17.** Modèle topologique de données [KHUAN, *et al*, 2008.b]

Ce modèle topologique propose le stockage de la topologie d'un polyèdre en utilisant une chaîne de caractères. Ce type de stockage est simple à lire mais présente un inconvénient majeur car il est lourd à interpréter par un programme. Il faut adopter un parseur de chaînes de caractères pour déchiffrer les différentes parties de la chaîne de caractères. Cette méthode s'avère être très utile en termes de définition de données géométriques 3D sous forme de polyèdres et d'opérations topologiques correspondantes. En revanche, lors d'un ensemble important de données spatiales (par exemple une ville), le stockage de toutes les informations topologiques entre les objets spatiaux est une procédure très lente et génère un volume supplémentaire de données qui ne seront pas forcément toutes utilisées dans des requêtes topologiques.

### 3.3. MATRICE DES NEUFS INTERSECTIONS (9I)

L'étude des relations topologiques binaires (entre deux objets) est un moyen approprié pour recenser toutes les relations possibles entre tels objets. La matrice des 9-intersections (9I), développée par [J.EGENHOFER, *et al*, 1990], est une technique très répandue dans le domaine de la topologie indiquant toutes les relations d'intersections possibles entre deux objets.

9I distingue trois composants de chaque objet A: i) son intérieur :  $Int(A)$  ou  $A^\circ$ , ii) son extérieure :  $Ext(A)$  ou  $A^1$  et iii) sa frontière :  $Bnd(A)$  ou  $\partial A$ , et examine les possibilités d'intersection entre chaque composant d'un objet A avec chaque composant d'un objet B [ELLUL, *et al*, 2008]. Le résultat d'intersection est une matrice de dimension (3\*3) décrite dans la figure (Fig3.18).

$$R(A,B) = \begin{pmatrix} Int(A) \cap Int(B) & Int(A) \cap Bnd(B) & Int(A) \cap Ext(B) \\ Bnd(A) \cap Int(B) & Bnd(A) \cap Frt(B) & Bnd(A) \cap Ext(B) \\ Ext(A) \cap Int(B) & Ext(A) \cap Bnd(B) & Ext(A) \cap Ext(B) \end{pmatrix}$$

Ou

$$R(A,B) = \begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^\circ & \partial A \cap B^{-1} \\ A^\circ \cap \partial B & A^\circ \cap B^\circ & A^\circ \cap B^{-1} \\ A^{-1} \cap \partial B & A^{-1} \cap B^\circ & A^{-1} \cap B^{-1} \end{pmatrix}$$

**Fig3.18.** Matrice des 9-Intersections

Pour illustrer schématiquement le principe de 9I, le tableau (Tab3.3) suivant montre une application de 9I qui étudie les différentes intersections possibles entre deux polygones 2D.

	Intérieur	Frontières	Extérieur
Intérieur			
Frontières			
Extérieur			

**Tab3.3** La matrice des 9-intersection appliquée sur deux polygones 2D [CODEHAUS, 2009]

[WEI, *et al*, 1998] proposent une application de 9I entre des objets k-simplex avec ( $0 \leq k \leq 3$ ). Les 0-simplex sont des *points*, les 1-simplex sont des *lignes*, les 2-simplex sont des *surfaces* et les 3-simplex sont des *volumes*. Le résultat de 9I est une matrice S de dimension (3\*3).

$$R(A,B) = \begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^\circ & \partial A \cap B^{-1} \\ A^\circ \cap \partial B & A^\circ \cap B^\circ & A^\circ \cap B^{-1} \\ A^{-1} \cap \partial B & A^{-1} \cap B^\circ & A^{-1} \cap B^{-1} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{pmatrix}$$

**Fig3.19.** Matrice des 9-Intersection pour des objets n-simplex [WEI, *et al*, 1998]

Avec  $S_{ij}$  ( $i,j = 1, 2, 3$ ) =  $\emptyset, 0, 1, \dots, \dim(S_{ij})$ .  $\dim(S_{ij})$  est une fonction qui retourne la dimension d'un ensemble de points  $S_{ij}$ .  $\dim(S_{ij})$  peut avoir cinq valeurs possibles  $\emptyset$  (pas d'intersection), 0 (point), 1 (ligne), 2 (surface), 3(volume).  $5^9$  relations topologiques sont alors possibles entre deux k-simplex. Telles relations topologiques peuvent être classées en six classes formalisées mathématiquement comme suit:

- Relation « touches »  
 $\langle A, \text{touch}, B \rangle \Leftrightarrow (A^\circ \cap B^\circ = \emptyset) \wedge (A \cap B \neq \emptyset)$
- Relation « in »  
 $\langle A, \text{in}, B \rangle \Leftrightarrow (A^\circ \cap B^\circ = \emptyset) \wedge (A \cap B = A)$
- Relation « cross »  
 $\langle A, \text{cross}, B \rangle \Leftrightarrow \dim(A^\circ \cap B^\circ) < (\max(\dim(A^\circ), \dim(B^\circ))) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B)$
- Relation « overlap »  
 $\langle A, \text{overlap}, B \rangle \Leftrightarrow (\dim(A^\circ \cap B^\circ) = \dim(A^\circ) = \dim(B^\circ)) \wedge (A \cap B \neq A) \wedge (A \cap B \neq B)$
- Relation « disjoint »  
 $\langle A, \text{disjoint}, B \rangle \Leftrightarrow A \cap B = \emptyset$

- Relation « equal »  
 $\langle A, \text{equal}, B \rangle \Leftrightarrow \langle A, \text{in}, B \rangle \wedge \langle B, \text{in}, A \rangle$

Le modèle topologique de [WEI, *et al*, 1998] est traduit en un diagramme Entité-Relation (ER) tel que repris dans la figure (Fig3.20) suivante.

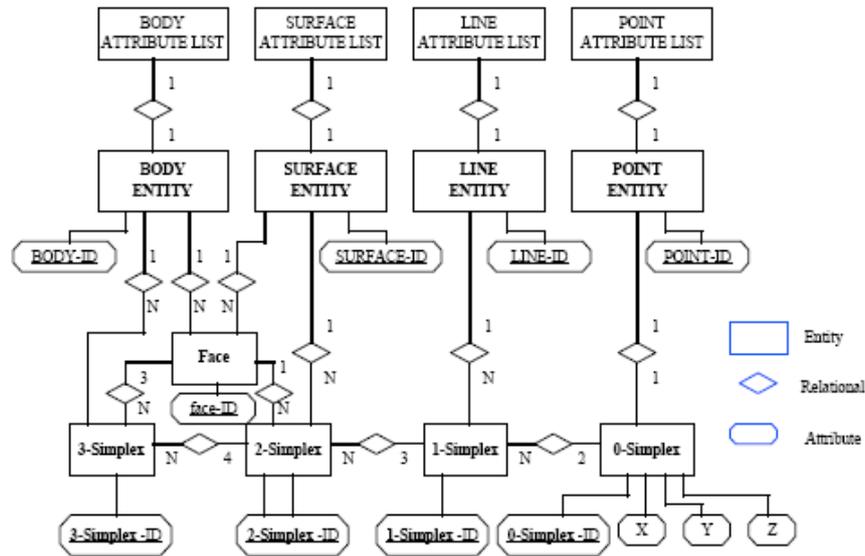


Fig3.20. Diagramme ER du modèle topologique [WEI, *et al*, 1998]

Pour distinguer les différentes relations topologiques décrites dans 9I, [ZLATANOVA, 2000] représente graphiquement une énumération des opérations qui peuvent exister entre des objets simples telles que les relations *ligne-ligne*, *surface-ligne*, *volume-ligne*, *surface-surface*, etc. Alors que [KUFONIYI, 1995] cité dans [ZLATANOVA, 2000] et [ELLUL, *et al*, 2008] élaborent une représentation numérique sur trois chiffres pour chaque relation 9-intersections. [ZLATANOVA, 2000] propose une classification des relations 9-intersections comme décrit dans le tableau (Tab3.4) suivant :

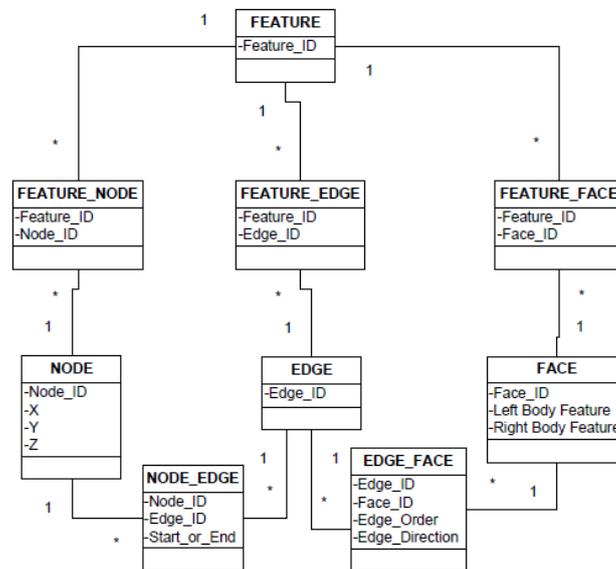
Dimension de l'espace englobant	Dimension d'objets	Nombre de relations
1	Ligne-Ligne	8
2	Ligne-Ligne	33
3	Surface-Ligne	31
2	Surface-Surface	8
3	Volume-	8
3	Volume	38
3	Surface-Surface	19
3	Volume-Ligne	19
	Volume-surface	
	Total	164

Tab3.4 Table Relations topologiques 3D de la matrice 9-intersections [ZLATANOVA, 2000].

Dans un espace 3D, le nombre des combinaisons possibles entre des primitives géométriques peut être contradictoire selon le modèle topologique utilisé [ZLATANOVA, 2000] cité dans [LACHANCE, *et al*, 2006]. Par exemple, la frontière d'une surface peut s'intersecter avec l'intérieur d'une autre dans un espace 3D. Cela n'est pas possible dans un espace 2D. Ainsi les possibilités de combinaison entre des primitives géométriques varient en fonction de leur dimension (2D ou 3D), et le type de leurs frontières (connectées ou non), du modèle topologique utilisé, etc. [LACHANCE, *et al*, 2006]. De plus, l'extérieur d'un objet ne peut pas être formalisé mathématiquement car il englobe un nombre infini de

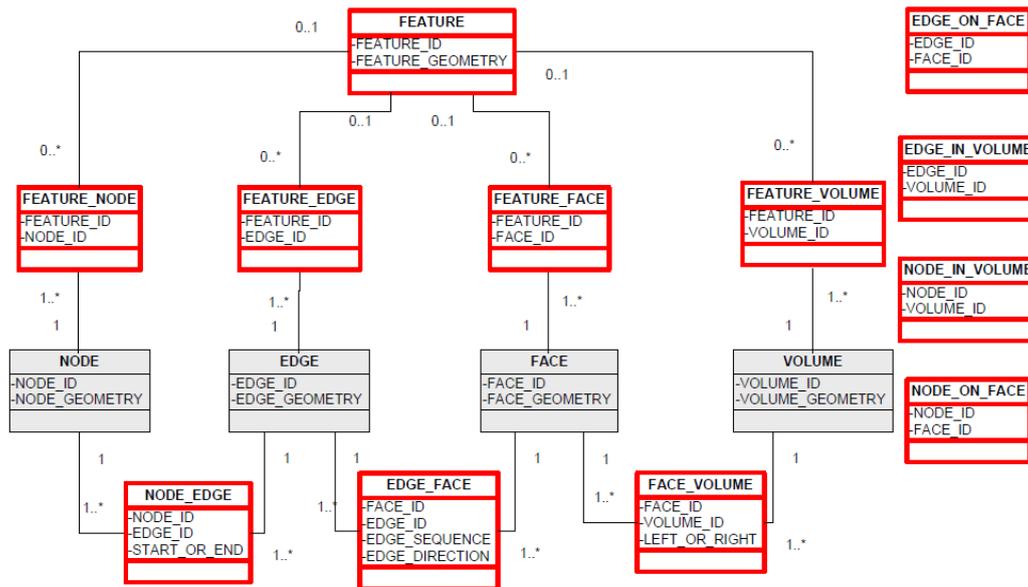
points. L'application de 9I change selon le contexte et les besoins topologiques. Par exemple, le résultat d'une combinaison entre deux objets peut être formalisée par l'existence ou l'inexistence d'une intersection entre eux ( $\neg\emptyset$  ou  $\emptyset$ ) ou par la dimension de l'objet résultat de l'intersection (0 pour les points, 1 pour les lignes, 2 pour les surfaces, 3 pour les volumes et  $\emptyset$  s'il n'y a pas d'intersection).

[ELLUL, *et al*, 2008] classifient les relations topologiques binaires (pouvant apparaître entre deux objets spatiaux) en trois classes principales : les relations d'adjacence (deux objets sont-ils proches), les relations de composition (un objet est-il dans un autre objet), les relations d'intersection (y a-t-il une certaine interaction entre deux objets). Pour identifier de telles relations, [ELLUL, *et al*, 2008] utilisent 9I sur des modèles B-Rep. La frontière d'un objet consiste en un ensemble de primitives topologiques ayant une dimension inférieure à la dimension maximale de cet objet. Un exemple simple est la frontière d'un *volume* peut être caractérisée par des *faces*, *arcs* ou *nœuds*. La structure B-Rep est définie dans la figure (Fig3.21) suivante :



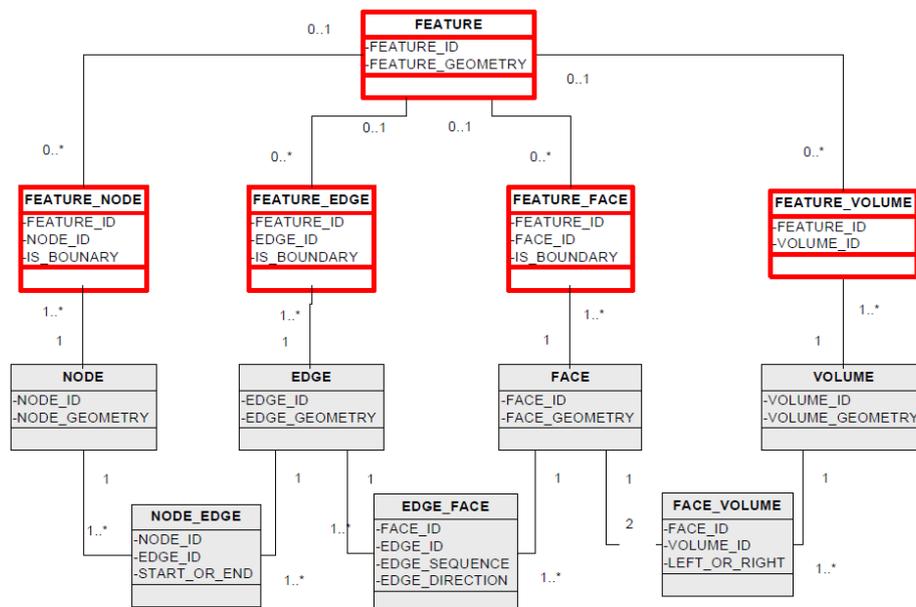
**Fig3.21.** Structure B-Rep modifiée pour supporter le partage des mêmes primitives par plusieurs objets.

Ce modèle décrit la composition d'un objet en termes de primitives topologiques (*nœuds*, *arcs* et *faces*) ainsi que leurs différentes relations topologiques. Malgré le fait qu'on puisse les composer à partir d'un ensemble de *faces*, les *volumes* ne sont pas explicitement décrits dans ce modèle. Ce modèle ne permet pas de représenter les relations topologiques possibles entre deux volumes (un volume est-il composé d'autres volumes, les géométries partagées entre des volumes, etc.). Le modèle BB-Rep (Binary B-Rep), représenté dans la figure (Fig3.22), décrit une extension de B-Rep qui prend en compte des volumes. L'identification des relations d'adjacence, de composition et d'intersection entre des volumes sont désormais possibles.



**Fig3.22.** BB-Rep (Binary B-Rep) : Les tables qui apparaissent en rouge sont celles nécessaires pour la détermination des relations des 9-intersections [ELLUL, *et al*, 2008]

Toutes les relations sont explicitement décrites, le modèle est lisible et facile à interpréter. Cependant, sa traduction en un modèle relationnel génère plusieurs jointures ce qui rend son interrogation très coûteuse en terme de temps de réponse. Par exemple, pour identifier les nœuds associés à un volume, on doit effectuer une série de jointures *Feature*, *Feature\_Volume*, *Face\_Volume*, *Edge\_Face*, *Node\_Edge*. Le modèle MBB-Rep (Modified Binary B-Rep), décrit dans la figure (Fig3.23), est proposé par [ELLUL, *et al*, 2008] pour optimiser ces jointures en créant des relations directes entre les éléments (*Features*) et les primitives topologiques ce qui évite de faire des jointures multiples.



**Fig3.23.** MBB-Rep (Modified Binary B-Rep) : Les tables qui apparaissent en rouge sont celles nécessaires pour la détermination des relations des 9-intersections [ELLUL, *et al*, 2008]

### 3.4. LA TOPOLOGIE DANS LES OUTILS SPATIAUX

La plupart des outils spatiaux (SIG, SGBDS, etc.) disposent de fonctionnalités topologiques permettant ainsi à leurs utilisateurs d'interroger qualitativement les données spatiales. [LACHANCE, *et al*, 2006] définissent une structure topologique spécifique qui s'inscrit dans la modélisation par tétraèdres. Cette modélisation s'applique sur des objets naturels, notamment des objets géologiques. Le modèle topologique appelé *GeoTEN* (Fig 3.24) ci-dessous se base sur la terminologie spécifiée par le schéma spatial ISO 19107. Ce modèle se compose d'objets topologiques précédés par le symbole « TP » (gris foncé), les objets topologiques complexes ayant une dimension variable 1, 2 ou 3 (blanc) et les primitives topologiques (gris claire).

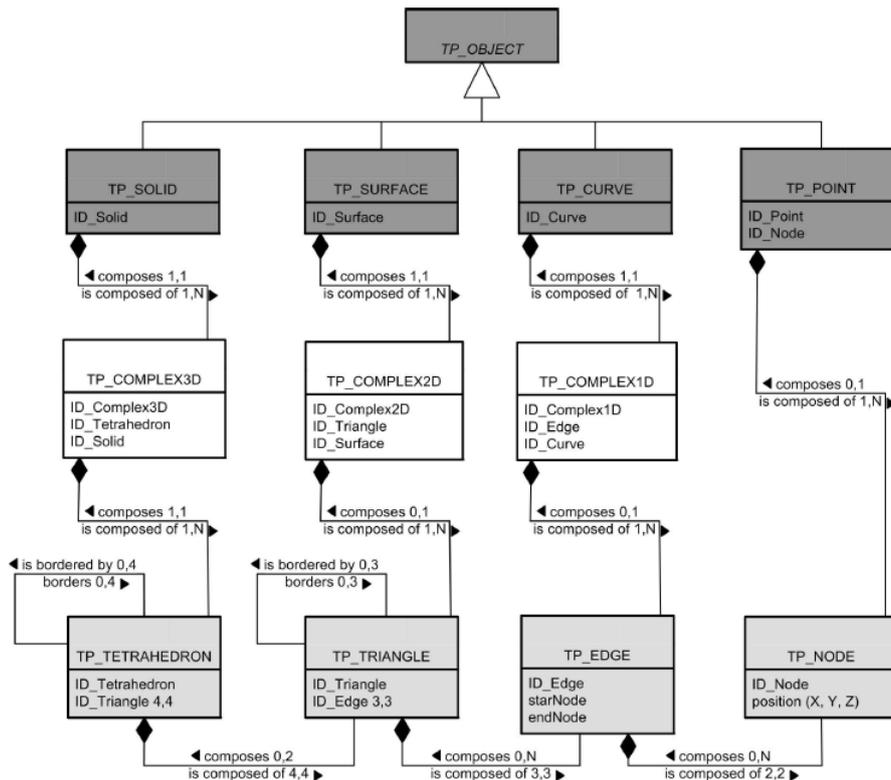


Fig3.24. Diagramme UML GeoTen [LACHANCE, *et al*, 2006]

Le but du modèle consistait à mettre en place un langage de requêtes topologique appelé *TQuery* (Topological Query) compatible avec la plate-forme de modélisation *GOCAD* (Geological Object Computer Aided Design) et respectant la structure *GeoTEN*. Uniquement deux opérateurs topologiques sont développés et implémentés dans le langage *TQuery* et liés à la structure *GeoTEN*. Il s'agit des relations « *TP\_Surface* "touch" *TP\_Solid* » et « *TP\_Point* "contained in" *TP\_Solid* ». Par exemple, la figure (Fig3.25) décrit la matrice correspondant à la relation « *TP\_Point* A "contained in" *TP\_Solid* B ».

$$R(A,B) = \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ - & - & - \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \text{ ou } R(A,B) = \begin{pmatrix} \emptyset & \neg\emptyset & \emptyset \\ - & - & - \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

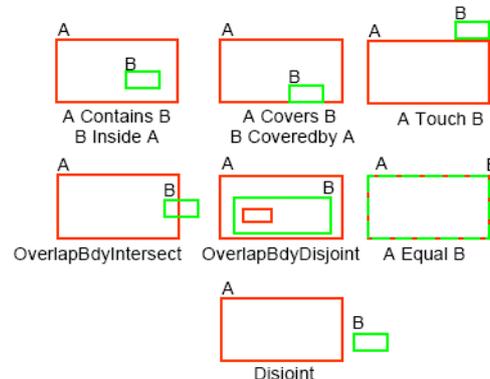
Fig3.25. Matrice 9I correspondant à l'opérateur "contained in" de *TQuery* [LACHANCE, *et al*, 2006]

Pour identifier chaque primitive topologique *TP\_Node* correspondant à ces relations, *TQuery* analyse individuellement chaque *TP\_Node* dans l'objet correspondant et identifie celles respectant la matrice

en question. Des mécanismes spécifiques sont utilisés par *TQuery* pour identifier les matrices de relations topologiques telles que la comparaison des coordonnées des nœuds, la comparaison des triangles communs, vérification des voisins des triangles, etc.

Plusieurs systèmes se basaient sur des modèles topologiques pour décrire les relations spatiales entre différents objets spatiaux. Les premiers systèmes tels qu'ARC/INFO [MOREHOUSE, 1985], TIGRIS [HERRING, 1987] et TIGER/DB [BROOME, 1990] utilisaient des fichiers binaires et des bibliothèques géographiques organisées en un ensemble de cartes. Ces systèmes ont été remplacés par des systèmes fondés sur des bases de données relationnelles, objets et objets-relationnelles.

Depuis sa version 7.1.6 de 1994 jusqu'à sa version actuelle 11g de 2009, ORACLE ne cesse d'intégrer et améliorer les performances de stockage ou d'interrogation de données spatiales. ORACLE spatial 11g définit un modèle topologique comme alternative du modèle géométrique (données vecteurs stockées dans des objets *SDO\_GEOMETRY*). Un modèle topologique décrit les relations entre les objets spatiaux. La géométrie d'un objet peut être dérivée d'un modèle de données topologique et inversement. En effet, stocker un modèle et en interpréter l'autre modèle constitue une approche efficace. Cependant, cela peut alourdir d'une façon considérable les requêtes géométriques et/ou topologiques [SCHÖN, *et al*, 2008]. ORACLE spatial définit des modèles géométriques et topologiques intégrés à travers le type *SDO\_TOPO\_GEOMETRY* pour le stockage d'éléments géométriques partagés. Les éléments topologiques sont stockés généralement sous forme de *nœuds*, *arcs* et *faces*. Un nœud peut être non connecté à d'autres nœuds. Un arc est une géométrie de type *line-string* connectée à deux ou plusieurs nœuds. Une face est une surface polygonale entourée par un ensemble fermé d'arcs. Au jour d'aujourd'hui, ORACLE ne propose pas une topologie 3D pour déterminer les relations spatiales entre des objets volumiques 3D.



**Fig3.26.** Relations topologiques traitées par ORACLE

Au niveau des outils spatiaux, la topologie constituait une fonctionnalité non négligeable. ArcGIS de ESRI propose pour ce faire un traitement particulier de la topologie. Le but majeur de l'utilisation de la topologie est d'assurer une meilleure qualité des données dans une Geo-database (base de données géo-spatiale) et de permettre ainsi des analyses spatiales avancées sur des lots de données intégrées. La figure (Fig3.27) montre les différentes primitives topologiques utilisées par ArcGIS, les relations entre elles ainsi que la manière de leur stockage dans la base de données.

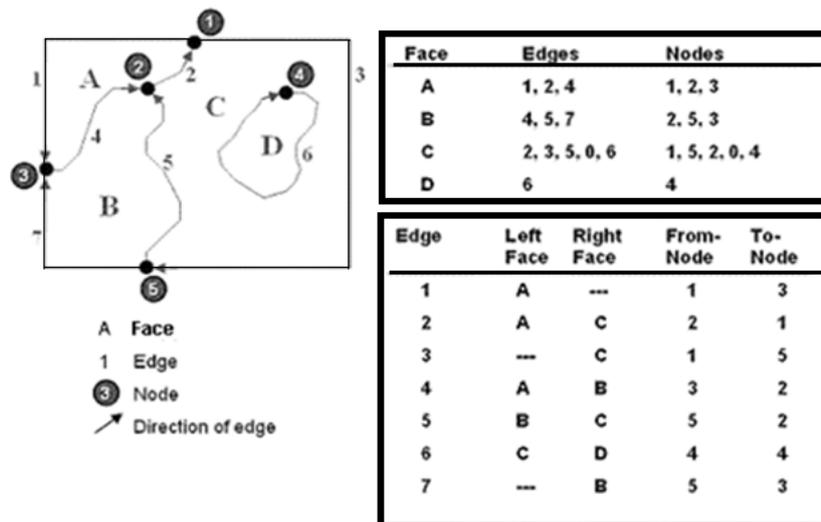


Fig3.27. Eléments topologique d'ArcGIS et leurs relations

ArcGIS permet de créer et valider des topologies en se basant sur une ou plusieurs Geo-databases. Il faut tout d'abord choisir tous les éléments impliqués dans la topologie, définir leurs types (point, ligne ou polygone), ajouter des données thématiques pour distinguer les différents éléments (par exemple, un numéro de parcelle pour chaque polygone), définir quelques rôles topologiques (par exemple, les polygones correspondant aux parcelles ne doivent pas se chevaucher), etc.

PostGIS propose à son tour un ensemble de fonctionnalités pour faire des requêtes topologiques 2D sur des données spatiales (pouvant avoir des coordonnées 3D). En effet, il offre des fonctions pour calculer les distances (*ST\_DISTANCE*), la connectivité (*ST\_TOUCHES* et *ST\_DISJOINT*), les intersections (*ST\_INTERSECTS*), etc. Il propose toute une bibliothèque d'opérateurs spatiaux permettant de traiter ce type de données. Jusqu'à présent, PostGIS n'offre pas de fonctionnalités pour gérer des géométries 3D (pour le stockage des solides, par exemple) ni des topologies 3D (telle que l'intersection de deux solides).

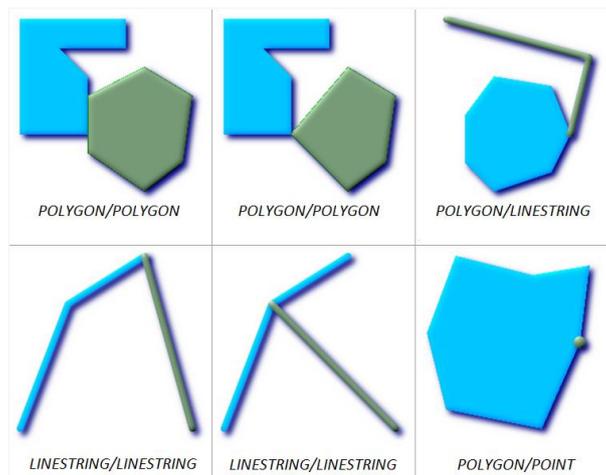


Fig3.28. Exemple d'une relation "ST\_TOUCHES" de PostGIS

Dans ses travaux de recherche [KHUAN, *et al*, 2008.a] propose des polyèdres comme un nouveau type de données 3D implémentés sous PostGIS ainsi que les fonctionnalités d'analyse correspondantes.

**CONCLUSION**

Naturellement, voire techniquement, nous ne pouvons percevoir qu'une partie du monde réel qui nous entoure. L'informatisation des données spatiales constitue un moyen incontournable qui a beaucoup facilité, à travers des modèles virtuels, la visualisation ainsi que l'analyse spatiale des objets et des phénomènes spatiaux.

La modélisation des données spatiales, décrite dans ce chapitre, reste une problématique de recherche assez considérable dont l'objectif majeur consiste à optimiser le stockage et la manipulation des données géo-référencées. En effet, les organismes internationaux, en particulier l'ISO et l'OGC ainsi que les communautés des chercheurs autour du monde travaillent en parallèle, et parfois en collaboration, pour résoudre les problèmes fondamentaux en matière de modélisation spatiales. Vu la diversité des contextes spatiaux, un modèle spatial ne peut pas être universel. Généralement, un modèle spatial est créé pour répondre à des besoins particuliers. Un modèle spatial doit tenir compte de plusieurs aspects tels que le type de données spatiales, leur volume, le contexte dans lequel est créé et utilisé ce modèle spatial, etc.

L'utilisation et l'implémentation de données spatiales sont effectuées à travers un ensemble d'outils informatiques dédiés. Parmi ces outils, les Systèmes d'Information Géographique (SIG) sont des demandeurs potentiels de ce type de données. Il s'agit de plateformes informatiques constituant l'interface entre les utilisateurs finaux et les sources de données spatiales. Ces plateformes offrent une gestion de telles données en termes de stockage, manipulation, visualisation, analyse, etc. Les SIG feront l'objet du chapitre suivant qui s'intéresse principalement aux concepts et fondements relatifs aux SIG.



**CHAPITRE.4 SYSTEME D'INFORMATION GEOGRAPHIQUE (SIG)**

Résumé.....	76
Introduction.....	76
<b>1. Définitions et terminologies.....</b>	<b>76</b>
1.1. SIG (Système d'Information Géographique).....	76
1.2. SIGM (Système d'Information Géographique Multimédia).....	79
<b>2. Base de données spatiale (BDS).....</b>	<b>79</b>
<b>3. SGBD spatiaux.....</b>	<b>82</b>
<b>4. SIG et l'open source.....</b>	<b>84</b>
<b>5. SIG et les outils commerciaux.....</b>	<b>86</b>
<b>6. SIG et Internet.....</b>	<b>88</b>
Conclusion.....	92

« On peut acheter un logiciel de SIG mais pas un SIG »

(IGN Magazine, Février/Janvier 2005)

## **RESUME**

*L'informatisation des données spatiales a apporté une valeur ajoutée considérable aussi bien pour les communautés d'utilisateurs experts que pour des utilisateurs novices. Les SIG constituent des outils incontournables pour l'automatisation, à partir de diverses sources, de l'organisation, de la gestion, de la représentation et de l'analyse des informations géographiquement localisées. Les SIG ne sont plus des applications isolées au sein d'une organisation. Ils sont intégrés dans le processus de développement pour répondre aux différents besoins des utilisateurs. Mais, la mise en place d'un SIG reste une tâche non aisée. Plusieurs considérations doivent être prises en compte pour la création de tout SIG telles que la définition des besoins, le choix des technologies à utiliser, la conception, le développement, etc.*

## **INTRODUCTION**

On remarque que l'utilisation des données relatives, de près ou de loin, à une localisation dans l'espace s'élargisse, de plus en plus, pour des domaines d'applications multiples et pour des fins assez diversifiées. Par exemple, la donnée spatiale est maintenant employée par des entreprises de transport pour la planification et l'optimisation des parcours, par des autorités régionales pour la gestion territoriales, par des services nationaux ou internationaux pour la protection des zones urbaines et du patrimoine naturel contre la pollution, les incendies, etc. En effet, cette prolifération d'utilisation des données spatiales a créé la nécessité de développer des systèmes "robustes" capables de mettre en œuvre différentes applications sur des données géo-référencées. En situant ce chapitre par rapport aux deux chapitres précédents, les SIG impliquent des données spatiales avec toutes les problématiques associées (modélisation géométrique et topologique, stockage, optimisation, etc.). Ces données spatiales peuvent impliquer à leur tour toutes les techniques relatives à la 3D (modélisation, rendu, animation, etc.). Dans ce cadre, ce chapitre introduit les SIG en tant qu'outils exploitant les données spatiales pour diverses fonctionnalités et en utilisant différentes technologies que ce soit pour le stockage ou la manipulation de ces données pour des fins d'analyse. Ce chapitre présente alors les principaux concepts que nous allons aborder tout au long des travaux de cette thèse qui consistent à implanter un SIG 3D sur Internet pour la gestion des modèles urbains, en particulier des bâtiments.

En effet, la première partie présente quelques définitions relatives au domaine des SIG. La deuxième partie présente les Bases de Données Spatiales en termes de méthodes de modélisation. De la troisième jusqu'à la cinquième partie, les SIG sont expliqués respectivement par rapport à l'Open Source, les outils commerciaux et Internet. La sixième partie décrit les SGBD spatiaux les plus répandus et les plus utilisés au niveau international.

## **1. DEFINITIONS ET TERMINOLOGIES**

Dans cette partie, une brève définition du SIG est établie en le situant par rapport aux concepts relatifs.

### **1.1. SIG (SYSTEME D'INFORMATION GEOGRAPHIQUE)**

Dans la littérature, il existe une multitude de définitions de Système d'Information Géographique (SIG), dit aussi Systèmes d'Information Spatiale (SIS) ou encore Systèmes d'information à Référence Spatiale (SIRS). Mais toutes les définitions partagent la notion de données géo-référencées comme données de base pour tout SIG. Au niveau le plus simple, le SIG est vu comme un environnement fournissant des informations sur les objets spatiaux. La principale fonction est d'offrir à l'utilisateur

une interface graphique permettant la localisation des objets à décrire ou l'identification de ceux répondant aux caractéristiques souhaitées. Un SIG peut être défini comme un « *environnement conçu pour l'analyse et la modélisation de la distribution spatiale de phénomènes. Il se compose d'une base de données géographique (BDG), d'une boîte à outils contenant des procédures d'analyse, de gestion, de saisie et de représentation, ainsi qu'une interface utilisateur* » [COLLET, 1992]. La définition du SIG dépend du contexte dans lequel il est étudié. Il peut être vu comme un système informatique, comme outil de stockage de données géo-référencées, comme outil d'affichage de données spatiales, comme un outil d'aide à la décision, etc.

- **SIG, Système informatique**

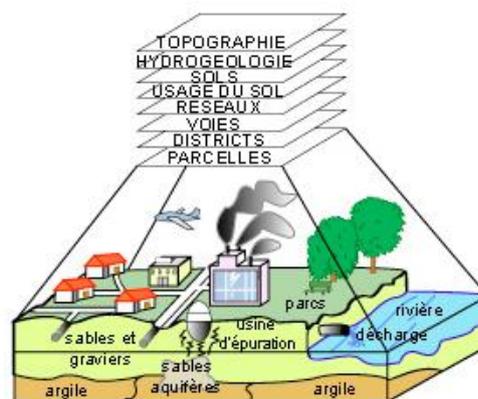
La liaison entre les SIG et les sciences de l'informatique est fondamentale, voire indispensable. En effet, la mise en place des SIG est basée sur le regroupement de données dans un système informatique. Dans ce contexte, un SIG peut être alors caractérisé comme « *système informatique de matériels, de logiciels, et de processus conçus pour permettre la collecte, la gestion, la manipulation, l'analyse, la modélisation et l'affichage de données à référence spatiale afin de résoudre des problèmes complexes d'aménagement et de gestion* » (définition officielle du SIG établie en 1988 par le FICCDC-comité fédéral de coordination inter-agences pour la cartographie numérique). En se référant à cette définition, nous pouvons dégager la définition suivante d'un SIG : « *c'est l'ensemble de données (spatiales et non spatiales) et de traitements rassemblés dans une application informatique permettant d'afficher des objets et phénomènes spatiaux pour des raisons d'analyse* ». Le système informatique est utilisé ainsi pour exécuter toutes les fonctionnalités du SIG (stockage, visualisation, analyse, etc.). Un SIG doit pouvoir permettre plusieurs fonctions dites les "5A" : Abstraction, Acquisition, Archivage, Analyse et Affichage de données à caractère spatial [DENEGRE, *et al*, 1997].

- **SIG, outil de stockage de données spatiales**

Les données spatiales sont généralement volumineuses et complexes. Un SIG doit pouvoir alors, d'une part, les stocker, et d'autre part, optimiser leur interrogation. Un SIG peut être défini alors comme « *un Système de Gestion de Base de Données pour la saisie, le stockage, l'extraction, l'interrogation, l'analyse et l'affichage de données localisées* » [LAURINI, 1993].

- **SIG, outil de visualisation de données spatiales**

Les données spatiales, dans leur forme de base de données, sont généralement illisibles et difficiles à interpréter. Pour ce faire, la transformation des données provenant des Bases de Données Spatiales (BDS) en des modèles spatiaux (bidimensionnels ou tridimensionnels) rendus à l'écran facilitera énormément la compréhension de leur contexte spatial ainsi que leur interprétation et leur analyse. Un SIG est représenté généralement sous la forme d'un ensemble de couches superposées. Chaque couche présente une thématique particulière concernant une zone géographique donnée.



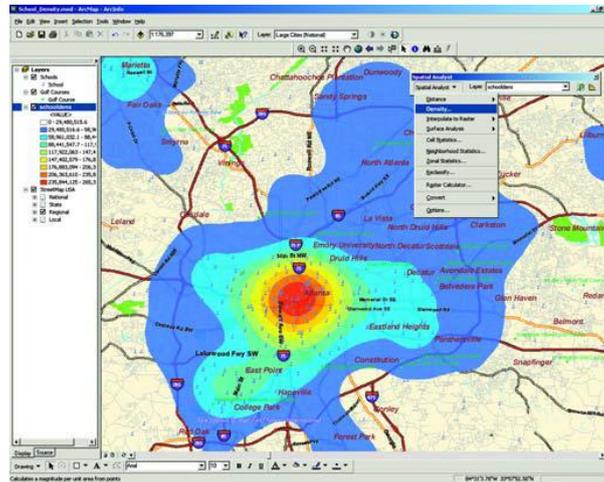
**Fig4.1.** Exemple de superposition de couches spatiales dans un SIG 3D [DENEGRE, *et al*, 1996]

La superposition des couches dans un SIG facilite l'analyse de données. En effet, un utilisateur donné a la possibilité de travailler proprement sur chaque couche à part, et après il peut visualiser l'ensemble. Un SIG doit pouvoir offrir une visualisation vectorielle et/ou raster de telles couches. Cette notion de couche est très importante dans les concepts fondamentaux des SIG. D'une part, elle permet de

regrouper des éléments de même nature dans une structure unique pour pouvoir ensuite la manipuler indépendamment des autres données. Et d'autre part, elle permet de structurer les données de manière à optimiser la gestion de celles-ci. Selon les SIG, une couche aura une signification différente, mais elle composera toujours un élément de base du système.

- **SIG, outil d'analyse de données spatiales**

L'une des caractéristiques fondamentales, voire la raison d'être, d'un SIG est sa capacité à fournir des outils d'analyse de données spatiales sous forme de traitements quantitatifs et/ou qualitatifs.



**Fig4.2.** Exemple d'une analyse spatiale : Recherche de voisinage et de zones de densité dans un milieu urbain [ESRIFRANCE, 2010]

L'analyse peut porter sur la distribution d'objets spatiaux dans une zone géographique donnée, les différentes thématiques abordées par le SIG, les propriétés géométriques et topologiques des objets spatiaux, etc. La figure (Fig4.2) est une illustration de la représentation et de multiples analyses effectuées dans un SIG à partir d'un jeu de données métier.

- **SIG, outil d'aide à la décision**

L'analyse des données spatiales dans un SIG émet généralement un ensemble d'interprétations sur la base desquelles se reposeront des décisions concernant un phénomène spatial ou non. Dans ce contexte, un SIG peut être défini comme « *un ensemble de données repérées dans l'espace, structuré de façon à pouvoir en extraire commodément des synthèses utiles à la décision* » [DIDIER, 1993]. Cette définition qualifie le SIG d'un outil incontournable d'aide à la décision. En effet en s'inspirant de [DENEGRE, *et al*, 1996] cité dans [BRUSLE, 2008], un SIG doit pouvoir répondre à des questions du genre :

- **Où ?** (Où se trouve tel objet ou tel phénomène spatial ?)

Vu qu'il gère des données géo-localisées, le SIG est l'outil le mieux placé pour répondre à telle question. Cette question porte sur la distribution ou la répartition des objets et phénomènes spatiaux dans une zone géographique donnée.

- **Quoi ?** (Que trouve-t-on à tel endroit ?)

Cette question porte sur tous les objets ou phénomènes spatiaux se trouvant dans un endroit donné.

- **Comment ?** (Comment tels objets ou tels phénomènes sont répartis dans l'espace et quel type de relation existe-t-elle entre eux?)

C'est une question fondamentale dans l'analyse spatiale qui consiste à trouver des réponses concernant une problématique spatiale donnée qui porte sur la répartition des objets spatiaux dans l'espace et les relations entre eux.

- **Quand ?** (Quand est-ce-que tel phénomène spatial aura lieu ?)

Cette question porte sur l'analyse temporelle des objets ou phénomènes spatiaux. Elle consiste à faire des prévisions sur les changements qui auraient lieu concernant tels objets et phénomènes spatiaux.

- **Si ?** (Que se passerait-il si tel phénomène spatial aurait lieu et quelles conséquences se produiseraient sur tels objets spatiaux?)

Cette dernière question complète la question précédente. Elle permet de dégager des hypothèses concernant un scénario spatial et ses conséquences sur des objets spatiaux (simulation).

## 1.2. SIGM (SYSTEME D'INFORMATION GEOGRAPHIQUE MULTIMEDIA)

L'invasion des technologies numériques dans tous les secteurs de l'information constitue l'une des grandes innovations du siècle. Le mélange des diverses techniques à savoir les télécommunications, l'audiovisuel et la micro-informatique a rendu possible la cohabitation du texte, du graphisme, de l'image, du son et de la vidéo [MAREDJ, *et al*, 2001]. La notion de SIGM (Système d'Information Géographique Multimédia) est une notion récente. L'objectif des SIGM est de permettre aux utilisateurs des SIG de tirer profit des avancées des outils multimédia (images, son, vidéo, Web, etc.). Un SIG multimédia ne permet pas uniquement la collecte, l'analyse et le stockage de données dans des formats traditionnels tels que les textes, les images et les graphes, mais aussi de l'audio, des animations et des vidéos [SOOMRO, *et al*, 1999].

## 2. BASE DE DONNEES SPATIALE (BDS)

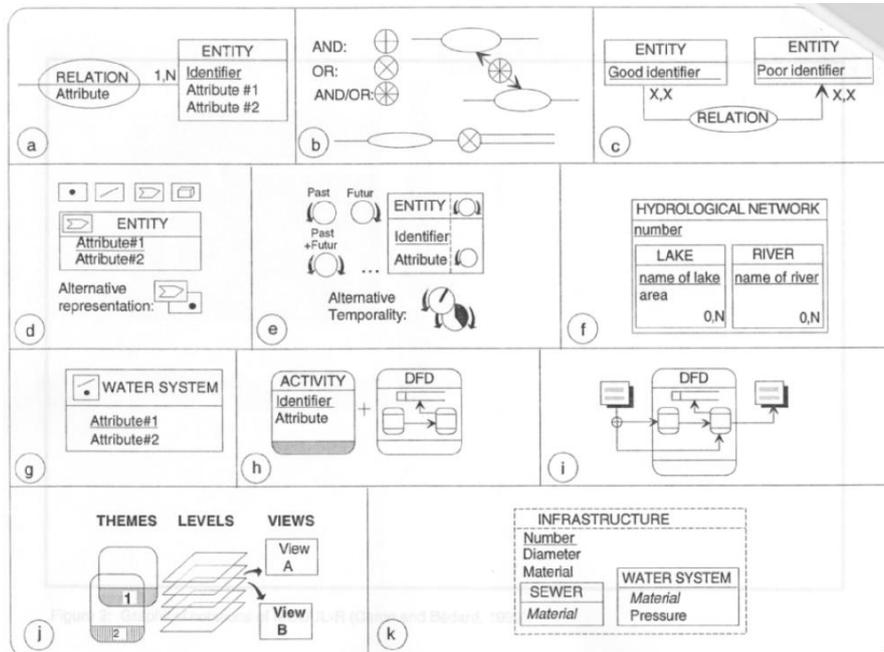
Les applications spatiales sont l'un des domaines où l'utilisation des techniques des Bases de Données (BD) connaît depuis quelques années un formidable essor. En termes de BD, on peut dire que la réalité géographique se compose d'une part des données géographiques (spatiales) et, d'autre part, des données alphanumériques reliées ou non aux données géographiques. Les données spatiales décrivent l'emplacement (la position absolue dans l'espace et/ou la position relative, par rapport à d'autres objets et/ou dans un système de référence local), la géométrie (les caractéristiques géométriques) et la topologie (les relations entre les objets : voisinage, superposition, etc.) de l'information à modéliser [SERVIGNE, *et al*, 2006]. Une BDS peut être alors qualifiée d'une BD, généralement volumineuse, contenant des données spatiales et non spatiale. Les données géographiques sont généralement caractérisées par des données attributaires ou thématiques (noms, désignations, descriptions, etc.) et également par des informations sur leur forme et leur localisation dans l'espace (coordonnées, géométries, topologies, etc.).

Les trois niveaux de modélisation (conceptuelle, logique et physique) sont nécessaires pour la mise en place de tout SI(G). Plusieurs solutions sont alors proposées tant au niveau SIG qu'au niveau formalismes de conception des BDS, etc. Nous classifions ces solutions en trois groupes : i) Les solutions qui ont fait recours aux BD classiques (relationnelles, objets, etc.). ii) Des solutions qui ont ajouté des extensions spatiales aux BD classiques. iii) Des solutions qui ont proposé des nouveaux modèles. Des travaux comme [BEDARD, *et al*, 1989], [BEDARD, *et al*, 1992], [HUGH, 1996], [HADZILACOS, *et al*, 1997], etc. traitent des problèmes liés à la modélisation des BDS tels que la représentation géométrique des objets spatiaux, les relations entre les géométries et les thématiques associées, les relations spatiales ou topologiques, etc.

Pendant la fin des années 1980 et surtout pendant les années 1990, plusieurs facteurs tels que les limites des systèmes à base de fichiers pour traiter des données de plus en plus importants, la généralisation du SQL (Structured Query Language) en tant que langage d'interrogation de BD, etc. ont conduit à une utilisation croissante des SGBDR (Systèmes de Gestion de Bases de Données Relationnelles). De ce fait, plusieurs solutions ont été proposées pour gérer des données spatialement référencées dans tel type de SGBD.

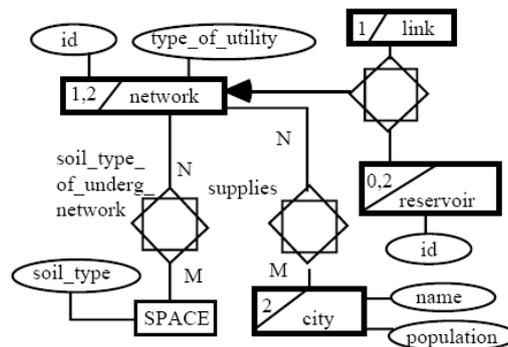
[BEDARD, *et al*, 1989] considèrent comme un bon modèle de données, tout modèle incluant tous les types d'entités à propos desquelles on veut avoir des informations, tous les attributs nécessaires pour la description des caractéristiques désirées des entités sélectionnées, et toutes les relations nécessaires entre les différents types d'entité. MODUL-R, proposé par [BEDARD, *et al*, 1992] est un formalisme de modélisation de données spatiales. C'est une extension du modèle traditionnel ER utilisant la technologie CASE (Computer-Assisted System Engineering). L'objectif majeur du modèle consiste à faciliter le lien entre la phase de conception de données spatiales et la phase de programmation.

MODUL-R présente toute une bibliothèque de symboles (*Fig4.3*) pour la modélisation de données spatiale. Cette méthode propose alors une manière pour la représentation des entités spatiales ainsi que les relations entre elles comme par exemple les relations entre entités sont représentées par des ellipses, les relations AND, OR et AND/OR sont représentées par des cercles contenant des symboles différents, etc.



**Fig4.3.** Les notions graphiques de l'approche MODUL-R [BEDARD, *et al*, 1992]

[HADZILACOS, *et al*, 1997] suggèrent une extension du modèle ER pour la modélisation conceptuelle des données géographiques. Leur modèle appelé Geo-ER est composé d'un ensemble d'entités spéciales, les relations entre elles ainsi que quelques nouveaux concepts. Outre les données géographiques, Geo-ER traite des données thématiques dues à la localisation des objets dans l'espace et non pas à leur nature. Cette méthode utilise une représentation particulière des entités spatiales (thématiques) ainsi que les relations entre elles. Par exemple dans une ville (city), peut correspondre 1 ou N entités de type réseau (network). La figure (*Fig4.4*) montre un exemple d'utilisation du formalisme Geo-ER.

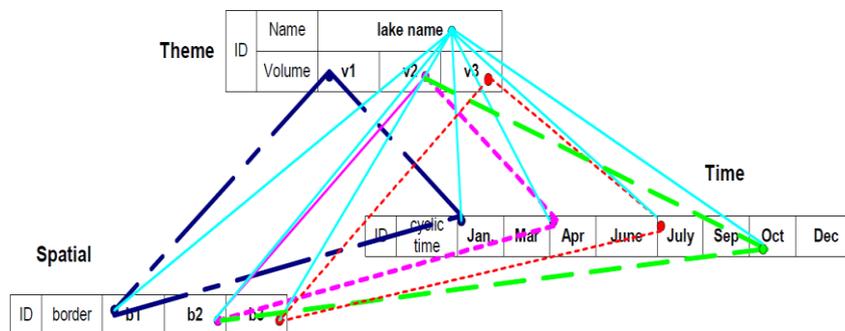


**Fig4.4.** Exemple d'utilisation du modèle Geo-ER [HADZILACOS, *et al*, 1997]

Alors que [BEDARD, *et al*. 1992], [HUGH, 1996] et [HADZILACOS, *et al*. 1997] exploitent des modèles ER, [WORBOYS, *et al*. 1990], [SACCHI, *et al*. 1994], [PANTAZIS, *et al*. 1996], [BONAN, *et al*. 2002], manient plutôt des modèles Orientés-Objets (OB) pour la modélisation des BDS.

[PANTAZIS, *et al.* 1996] ont proposé une méthode OB pour la modélisation des données géographiques appelée CONGOO (CONception Géographique Orientée Objet). Son objectif s'inscrit dans le cadre de la modélisation conceptuelle des données géographiques. Il s'agit de données ayant une représentation spatialement référencée, ainsi que les données attributaires associées. La méthode utilise les primitives géométrique *point*, *ligne*, *polygone*, *surface*, ainsi que leur combinaison. Elle utilise également un ensemble d'attributs, dits graphiques, comme la *couleur*, *forme*, etc. qui déterminent l'apparence des objets à modéliser.

[BONAN, *et al.*, 2002] proposent une modélisation spatio-temporelle OB des BDS. L'objectif du modèle est de modéliser une BDS efficace contenant des informations correctes et la structure logique est compréhensible par les utilisateurs et facile à maintenir et étendre. Cette méthode distingue trois types d'entité : entité thématique, spatiale et temporelle ainsi que trois types de relations entre ces entités : un-un, un-plusieurs et plusieurs-plusieurs. La figure (Fig4.5) montre un petit exemple de cette méthode montrant l'évolution de la bordure d'un lac au fil du temps (selon le mois).



**Fig4.5.** Intégration de données thématiques, spatiales et temporelles [BONAN, *et al.*, 2002]

Il existe plusieurs autres modèles qui ont été proposés pour modéliser les BDS tels que les modèles EER (Extended Entity-Relation) [WANG, *et al.*, 2001], STER (Spatial Temporal Entity-Relationship) [TRYFONA, *et al.*, 1999], STSQL (Spatial Temporal SQL) [BOHLEN, *et al.*, 1998], ERT (Entity-Relationship with Time) [RENOLEN, 2000], etc.

Plusieurs techniques sont utilisées pour accélérer l'interrogation de BDS tels que les indexes spatiaux. Les indexes les plus employés sont GiST (Generalized Search Tree) et R-Tree, B-Tree et Hash. D'après [SENSAGENT, 2010], les indexes spatiaux sont utilisés par des BDS (qui stockent des données relatives au positionnement des objets dans l'espace) pour optimiser les requêtes spatiales. Les indexes utilisés par les bases de données non spatiales ne peuvent pas interroger efficacement des entités telles que la mesure de la distance entre deux points ou tester si un point appartient à une zone d'intérêt, etc.

[PINET, *et al.*, 2003] expriment un aperçu sur les méthodes de conception et formalismes pour la géomatique (voir annexe page 222).

Le contenu d'une BDS peut porter sur des données cadastrales, des données cartographiques, des données géologiques, des données pour l'étude de l'occupation du sol, etc. En France, l'IGN (Institut Géographique National) est le premier fournisseur de données géographiques. Il fournit des BD raster (ou image) telle que la "BD SCAN1000", des BD vectorielles telles que la "BD CARTO" et même des BD à l'échelle européenne telles que le "EGM" (EuroGlobalMap) et le "ERM" (EuroRegionalMap), etc. Le RGE (Référentiel à Grande Echelle), produit de l'IGN, est un projet qui a commencé en 2000 et il a été achevé en 2008. Il consistait à produire des données cadastrales, des données vectorielles topographiques et données orthophotographiques couvrant tout le territoire français. A l'échelle internationale, il existe plusieurs BD géographiques pour répondre aux différents besoins telles que la BD "CORINE Land Cover" en tant que base de données européenne d'occupation biophysique des

sols, "SPADE" (Soil Profile Analytical Database of Europe), "SSURGO" (Soil Survey Geographic Database), etc.

### 3. SGBD SPATIAUX

Comme le stockage de données spatiales est une fonctionnalité primordiale dans un SIG, un SGBD spatial (SGBDS) constitue une composante basique dans tout SIG. Le choix d'un SGBDS se repose sur plusieurs critères comme le coût d'achat (gratuit ou payant), le coût de maintenance, les systèmes d'exploitation sur lequel il peut fonctionner, la performance d'interrogation, les fonctionnalités, les volumes de données, etc.

Les SGBDS devancent le simple stockage de données. En effet, ils incluent des fonctionnalités spatiales pour le traitement de tel type de données telles que le stockage de données géographiques, les fonctionnalités géométriques, topologiques, etc. Les SGBDS, peuvent être alors qualifiés de SGBD classiques sur lesquels sont greffées des couches de fonctionnalités spatiales. Il s'agit de modules complémentaires consistant à gérer des données spatiales.

Néanmoins, le choix d'un SGBDS est restreint. Parmi les SGBDS commerciaux se distinguent "Oracle spatial" et "Oracle locator" de la société "Oracle", "SQL server" de la société "Microsoft" et des SGBDS gratuits tels que "PostgreSQL" avec sa cartouche spatiale "PostGIS", "MySQL" avec son extension spatiale "MyGIS", "SQLite" avec sa couche spatiale "Spatialite", etc. Qu'il soit gratuit ou payant, un SGBDS doit pouvoir respecter les normes internationales notamment les spécifications de l'ISO, de l'OGC, etc.

⇒ Oracle est un SGBD relationnel (SGBDR) imaginé à partir des années 1970 par Oracle Corporation. Depuis la version 1 apparu en 1978 jusqu'à la version 11g Release 2 en 2009, Oracle a connu plusieurs époques et surtout des progrès considérables. En 1983, la version 3 d'Oracle apparaissait et supportait les transactions grâce aux fonctionnalités *commit* et *rollback*. En 1992, la version 7 d'Oracle a été publiée et supportait les contraintes d'intégrité, les procédures stockées et les déclencheurs. En 2003, la version 10g (le g signifie *grid*) a été publiée et elle supportait le *grid computing*. Et enfin, en septembre 2009, Oracle lança sa version 11g Release 2.

En tant que leader au niveau mondial, le SGBD Oracle a consacré une attention non négligeable aux données spatiales. Oracle lance son produit Oracle spatial pour la gestion et l'interrogation des données spatialement référencées. Oracle Spatial, une option d'Oracle Database 11g Enterprise Edition, intègre la prise en charge complète 3D et services Web pour gérer toutes les données géospatiales, notamment vectorielles et matricielles, topologie et modèles de réseau [ORACLE, 2009]. Les types 3D supportés par Oracle Spatial sont les *points*, les *lignes*, les *polygones* et les *solides* et aussi les agrégations tels que les *multi-points*, les *multi-lignes* et les *multi-surfaces*. Il intègre les spécifications de GML 3.1.1 et ISO 19107.

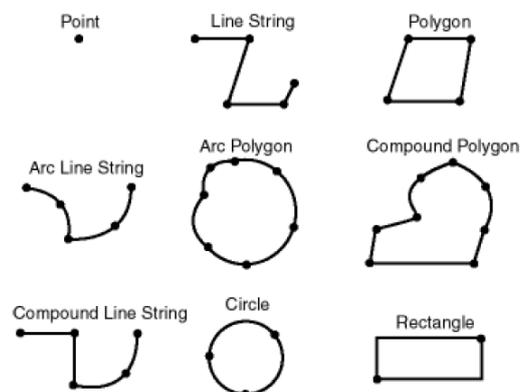
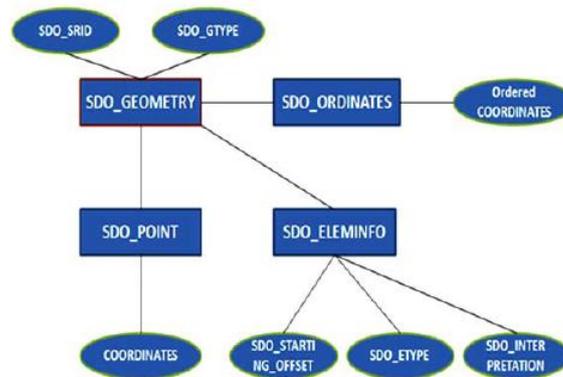


Fig4.6. Type de données spatiales d'Oracle

Il supporte les solides simples tels que les cubes ou les pyramides. Oracle spatial propose aussi un ensemble de fonctionnalités pour traiter les données spatiales, telles que *SDO\_Distance*, *SDO\_Geom.Anyinteract*, *SDO\_area*, *SDO\_volume*, *SDO\_CS.Transform*.



**Fig4.7.** SDO\_Geometry d'Oracle spatial

L'exemple ci-dessous explique une requête d'Oracle spatial qui retourne l'intersection topologique entre deux géométries.

```

SELECT SDO_GEOM.SDO_INTERSECTION (table1.shape, table2.shape, 0.005)
FROM table1, table2
WHERE table1.name LIKE '%building%'
  
```

⇒ PostgreSQL est un SGBD objet-relationnel (SGBDRO) libre. Il est créé en 1985 par MICHAEL STONEBRAKER comme un raccourci de la base de données "Ingres" et appelé au début Postgres. En 1995, Postgres intégrait SQL et se renommait en PostgreSQL. PostgreSQL supporte plusieurs fonctionnalités comme les transactions, les requêtes complexes, les déclencheurs, etc. Pour le qualifier en tant que SGBD spatial, PostgreSQL est couplé avec la couche spatiale PostGIS. PostGIS est un ensemble de fonctionnalités pouvant s'exécuter sur PostgreSQL pour traiter des objets spatiaux. PostGIS ne permet pas encore, avec sa version 1.5, de traiter des objets 3D tels que les solides. La 3D dans PostGIS se limite à des coordonnées 3D et quelques fonctionnalités 3D telles que *ST\_Box3D* (construire le cube englobant d'un ensemble de géométries), *ST\_Collect* (construire une collection de géométries), *ST\_Length3D* (renvoie la longueur 3D d'une polyligne), etc. [KHUAN, *et al*, 2008.a] proposent un type de données 3D appelé *Polyhedron* pour stocker des volumes dans PostgreSQL/PostGIS.

⇒ SQL Server est un SGBD développé et commercialisé par Microsoft à partir de 1994. SQL Server utilise des requêtes T-SQL (Transact-SQL). C'est une fonctionnalité de SQL Server qui utilise les procédures stockées et les déclencheurs. C'est un SGBDR qui garantit les contraintes d'intégrité des données qui y sont stockées. Comme les autres SGBD (Oracle et PostgreSQL/PostGIS) mentionnés ci-dessus, SQL Server permet, à son tour, le stockage de données géo-référencées compatibles avec les spécifications de l'OGC, l'utilisation d'index spatiaux, l'interrogation des données spatiales, etc.

⇒ Spatialite est une couche spatiale greffée sur SQLite (concaténation des deux mots SQL et light signifiant SQL léger). SQLite est un moteur de BD relationnelles, implémentant une grande partie du standard SQL tel que les contraintes, les clés primaires et étrangères, les déclencheurs, les Vues, les Index, etc. Sa particularité est sa capacité à être intégré directement dans les programmes (au contraire des autres SGBD utilisant l'architecture client serveur). Toutes ces caractéristiques rendent SQLite comme un moteur de BD très utilisé dans le monde. Spatialite est une bibliothèque spatiale implémentant les spécifications de l'OGC. En effet, les types de géométries utilisées sont *Point*, *Line*,

*Polygon, GeometryCollection, MultiPoint, MultiLineString, MultiPolygon.* Il implémente les SFS (Simple Features Interface Standard) pour SQL de l'OGC telles que *AsText(), GeomFromText(), Area(), PointN(),* etc., ainsi que les opérations topologiques telles que *Overlaps(), Touches(), Contains(),* etc. et les métadonnées spatiales proposées par l'OGC telles que *spatial\_ref\_sys, geometry\_columns,* etc.

Le tableau (*Tab4.1*) ci-dessous interprété de [BALDE, 2007] présente une brève comparaison entre les trois SGBDS les plus utilisés : Oracle, PostgreSQL/PostGIS et MySQL/MyGIS.

	Oracle	PostgreSQL/PostGIS	MySQL/MyGIS
<b>Dimensions</b>	2D (X, Y), 3D (X, Y, Z), 3D (X, Y, M), 4D (X, Y, Z, M) : M désigne la dimension temps	2D (X, Y), 3D (X, Y, Z), 3D (X, Y, M), 4D (X, Y, Z, M) : M désigne la dimension temps	2D (X, Y)
<b>Primitives géométriques</b>	Point, Linestring, Polygon, GeometryCollection, MultiPoint, MultiLineString, MultiPolygon, Rectangle, ArcLinestring, ArcPolygon, CompoundLinestring, CompoundPolygon, Circle	Point, Linestring, Polygon, GeometryCollection, MultiPoint, MultiLineString, MultiPolygon, CircularString, CompoundCurve, CurvePolygon, MultiCurve	Point, Linestring, Polygon, GeometryCollection, MultiPoint, MultiLineString, MultiPolygon
<b>Temps d'exécution</b>	<ul style="list-style-type: none"> <li>- Buffer, ConvexHull très performants</li> <li>- Jointures entre tables très longues (plusieurs dizaine de minutes)</li> <li>- Union de gros polygones très longue (plusieurs heures)</li> <li>- Ordre des opérandes important</li> </ul>	<ul style="list-style-type: none"> <li>- Jointures entre tables relativement lentes</li> <li>- Buffer, ConvexHull prenant beaucoup de temps</li> <li>- Opérateurs performants</li> </ul>	<ul style="list-style-type: none"> <li>- Requête généralement les plus rapides des trois systèmes</li> <li>- Jointures entre tables très rapides</li> <li>- Temps prometteurs</li> </ul>

**Tab4.1** Comparaison entre les SGBD spatiaux Oracle, PostgreSQL/PostGIS et mySQL

#### 4. SIG ET L'OPEN SOURCE

L'Open Source dans les SIG a débuté dans les années 1980 avec la création du SIG libre GRASS comme premier outils multi-plate-formes permettant de traiter des données vectorielles et raster. L'Open Source est très présent dans le monde du SIG que ce soit au niveau de l'outil SIG lui-même (des SIG libres tels que GRASS) ou au niveau des technologies utilisées par le SIG tels que les SGBD(S), les visionneurs de données spatiales, les serveurs spatiaux, des services Web, etc. Il existe une multitude de SIG libres mis à disposition par des groupes de projet à travers l'OGC, la plupart sont des SIG 2D. Le tableau (*Tab4.2*) présente un récapitulatif des caractéristiques techniques des SIG libres les plus connus (tableau interprété de [FREEGIS, 2010]).

SIG	Applications	Système d'exploitation	Fonctions	Standards (Web services)
GRASS	BD, SIG local	MacOS X, Windows, GNU/Linux, etc.	Géo-traitement, numérisation, animations, GPS, etc.	WMS, WPS
QGIS	SIG Web, SIG local	MacOS X, Windows, GNU/Linux, etc.	Géo-visionneur, GPS, Conversion de données, etc.	OGC WMS Client, OGC WMS Server, OGC WFS Server
DEGREE	SIG Web, SIG local	Windows, GNU/Linux, etc.	Géo-visionneur, Géo-traitement, 3D, etc.	OGC WMS Client, OGC WMS Server, OGC WFS Server, OGC WFS Client
GeoServer	SIG Web	MacOS X, Windows, GNU/Linux, etc.	Génération de cartes, édition et partage de données utilisées pour la génération des cartes	OGC WMS Client, OGC WMS Server, OGC WFS Server, OGC WFS Client
UMN MapServer	SIG Web	MacOS X, Windows, GNU/Linux, etc.	Traitement d'images	OGC WMS Client, OGC WMS Server, OGC WFS Server, OGC WFS Client

Tab4.2 Caractéristiques techniques de quelques SIG libres

Concernant les applications spatiales libres, il existe de nombreux travaux sur le stockage, la visualisation et le traitement de données spatiales. GeoTools [GEOTOOLS, 2010] est un ensemble de bibliothèques java libres fournissant des méthodes conformes aux normes pour la manipulation des données géo-spatiales, comme par exemple l'implémentation des SIG. GeoTools implémente les spécifications de l'OGC.

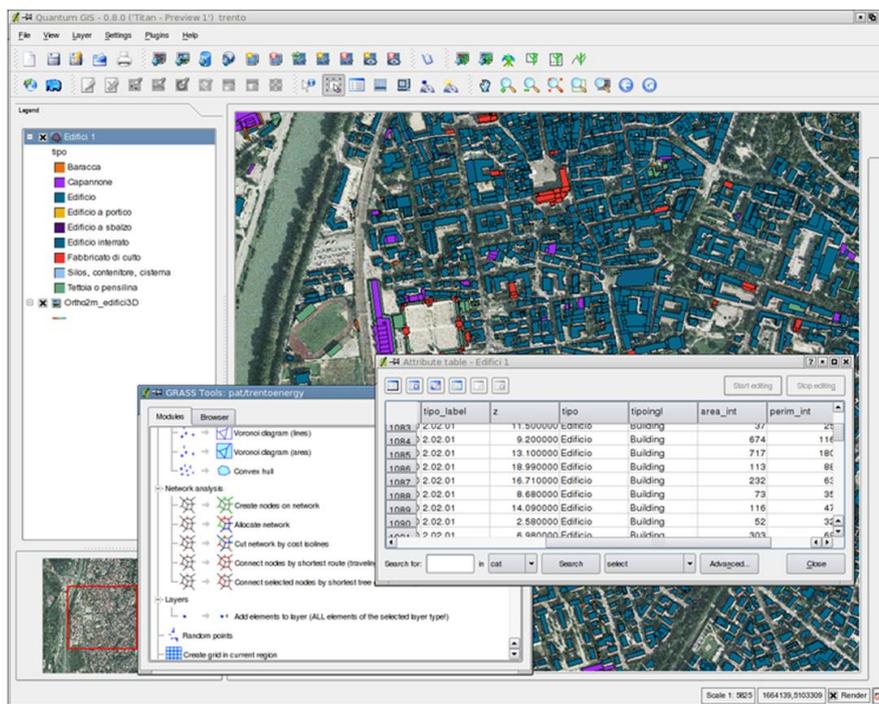


Fig4.8. Exemple d'une application SIG réalisée avec QGIS

Il existe beaucoup d'autres solutions libres, SIG et non SIG, locales et en ligne, permettant de fournir des services spatiaux intéressants telles que g3DGMV (3D Graphical Map Viewer) [G3DGMV, 2010] visionneur des MNE (Modèle Numérique d'Élévation) et GLN (Graphes Linéaires Numériques), [GEOSERVER, 2010] un serveur libre permettant de partager et modifier des données spatiales, [GMT, 2010] une collection de soixante outils pour la manipulation de données géographiques et cartésiennes, [GVSIG, 2010] un outils pour la manipulation et la visualisation de données SIG (GML, SHP) ou CAD

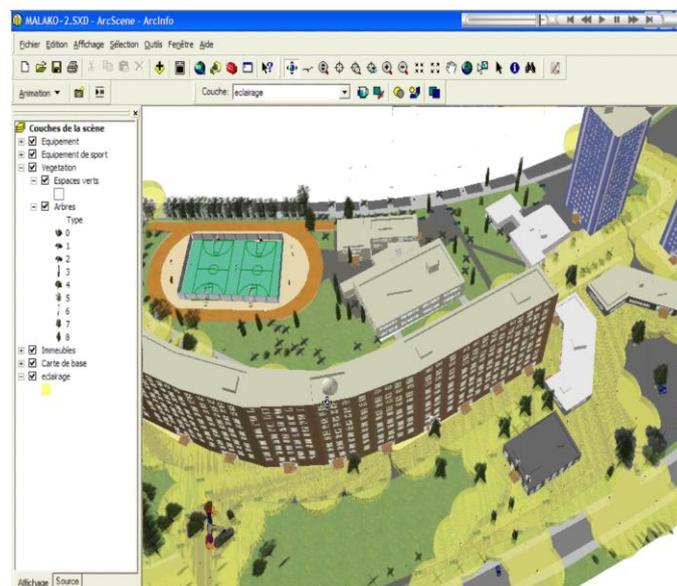
(DWG, DXF, etc.) ainsi que des BD (PostGIS, Oracle, etc.), [JUMP, 2010] application pour l'accès aux services WMS (Web Mapping Service) et des fichiers GML selon les spécifications de l'OGC, [MAPFISH, 2010] application pour le développement des outils de Web mapping, [MAPSERVER, 2010] un outil pour la publication de cartes sur Internet ainsi que des services Web (WFS, WMS, WCS), etc.

Une liste plus exhaustive des outils spatiaux ainsi que des solutions SIG libres est expliqué dans [OPENSOURCEGIS, 2008], [FREEGIS, 2010], etc.

## 5. SIG ET LES OUTILS COMMERCIAUX

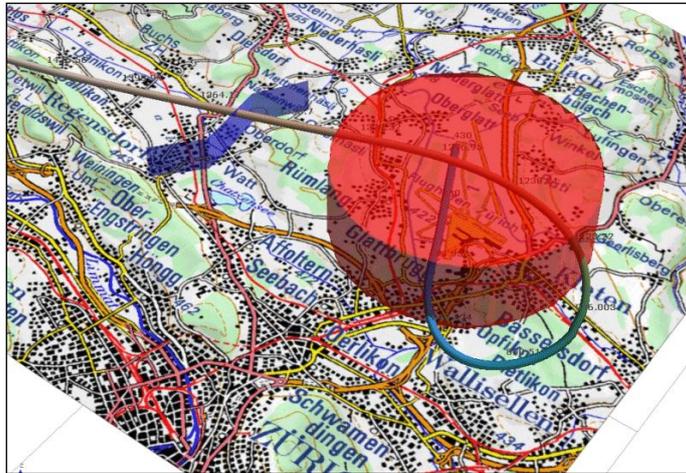
Vu leur importance en tant qu'outils d'aide à la décision, les SIG se sont rapidement commercialisés pour proposer des services généraux ou des services personnalisés selon les besoins des clients. Les premiers SIG commerciaux, créés par les sociétés "ESRI", "CARIS" et "M&S Computing", sont apparus au début des années 1980. ESRI s'est rapidement internationalisé, on évoque ESRI Etats-Unis, ESRI France, ESRI Canada qui sur des bases logicielles communes permettent de s'adapter aux spécificités de chaque pays, etc. Au fil du temps, d'autres logiciels SIG sont apparus sur le marché et constituaient des concurrents potentiels aux sociétés historiques précédentes. On peut ainsi citer "MapInfo" actuellement filiale du groupe "Pitney Bowes", "CityGIS" de la société "CityDisc/TIM Software", "Carto WinSTAR" et "Elyx" de la société "STAR-APIC", et récemment des produits SIG d' "Autodesk" tels que "Autodesk GIS software", "AutoCAD Map 3D" ou encore "MapGuide", etc.

⇒ Les produits SIG d'ESRI et de Pitney Bowes (MapInfo) sont les plus répandus au niveau mondial. ESRI propose une gamme de produits ArcGIS, autour du logiciel ArcMap, pour des utilisations diverses. On distingue les SIG bureautiques, SIG serveurs, SIG nomades, SIG pour les développeurs, SIG spécialisés [ESRIFRANCE, 2010]. Un SIG bureautique, dit Desktop, identifie, dans les données, des relations et des tendances qui sont moins visibles dans les bases de données, les feuilles de calcul ou les statistiques. Il offre la possibilité d'afficher des données sur une carte, de gérer et d'intégrer des données, de réaliser des analyses, de modéliser et d'automatiser des traitements, et d'afficher les résultats sur des cartes. Les différentes versions des produits ArcGIS sont ArcInfo, ArcEditor, ArcView (classées d'après le nombre décroissant de fonctionnalités), ArcReader (visionneur), ArcGIS Explorer (visionneur internet), ArcGIS Engine, etc. Les solutions SIG serveur sont capables de transférer des cartes, des modèles et des outils à d'autres utilisateurs de la même organisation, d'une manière qui s'intègre à leur environnement de travail. Le SIG nomade permet aux équipes de terrain de capturer, stocker, mettre à jour, manipuler, analyser et afficher des informations géographiques.



**Fig4.9.** Exemple d'un modèle 3D et une analyse de la luminosité avec l'extension 3D Analyst et rendu dans ArcScene [ESRIFRANCE, 2010]

⇒ MapInfo, créé pendant les années 1980, est une solution SIG permettant l'édition des cartes au format numérique. "Pitney Bowes MapInfo" [PBMAPINFO, 2010] propose une gamme de produits et services dédiés aux problématiques des Systèmes d'Information Géographique et du géomarketing décisionnel.



**Fig4.10.** Exemple d'une carte MapInfo : Aéroport de Zürich avec routes de vol et objets 3D extrudés

⇒ Carto WinSTAR [STAR-APIC, 2010] a été conçu pour les utilisateurs experts exigeant des fonctionnalités avancées d'acquisition, de structuration, d'intégration et de maintenance de données vectorielles et matricielles combinées, afin d'élaborer des modèles de données continues complexes et de gérer des applications aux transactions longues cohérentes. Carto WinSTAR est un ensemble d'outils d'administration SIG (droits d'accès, organisation, indexation, etc.), outils de traitement d'images, de géo-référencement, des fonctions SIG, analyses spatiales, études thématiques et statistiques, etc.

⇒ Autodesk a lancé dernièrement son produit "Autodesk GIS software" fondé autour du produit AutoCAD. Ce produit consiste à utiliser des données CAO (Conception Assistée par Ordinateur) dans des applications SIG. Il est capable de traiter des données vectorielles 2D et 3D et données raster. L'application supporte aussi des topologies CAO et SIG qui inclut des *nœuds*, des *arcs* et des *polygones*.

Le tableau (Tab4.3) ci-dessous présente une comparaison ainsi qu'une classification des principales solutions SIG (commerciales et libres) [MASTERSIG, 2005]. Les SIG commerciaux mentionnés dans cette partie ne présentent qu'un petit échantillon des produits existant sur le marché. Ces SIG sont complétés par des outils spatiaux répondant aux différents besoins pour le traitement de données spatiales.

Nom	Type	Mode	Système	Diffusion	Statut	Auteur
ARCVIEW	Suite logicielle	vecteur/raster	Windows	Commerciale	Propriétaire	ESRI
ARCGIS	Suite logicielle	vecteur/raster	Windows	Commerciale	Propriétaire	ESRI
GEO-CONCEPT	Suite logicielle	vecteur	Windows	Commerciale	Propriétaire	Géoconcept SA
MAPINFO	Suite logicielle	vecteur	Windows	Commerciale	Propriétaire	Axiom France
GEOMEDIA	Suite logicielle	vecteur/raster	Windows	Commerciale	Propriétaire	Intergraph SA
STAR	Suite logicielle	vecteur	Windows	Commerciale	Propriétaire	STAR Informatic
MANIFOLD	Suite logicielle	vecteur/raster	Windows	Commerciale	Propriétaire	Manifold
SAVANE	Suite logicielle	vecteur/raster	Windows	Scientifique	Propriétaire	IRD/M. Souris
SPRING	Logiciel	vecteur/raster	Windows	Gratuite	Propriétaire	Brazil's National Institute for Space Research
JUMP	Logiciel	vecteur/raster	Windows	Gratuite	Libre	Vivid Solution
GRASS	Logiciel	vecteur/raster	Linux/Windows	Gratuit	Libre	GRASS Development Team (depuis 1999)
IDRISI	Logiciel	vecteur/raster	Windows	Commerciale	Propriétaire	Clark Lab
SAGA	Logiciel	vecteur/raster	Linux/Windows	Gratuite	Libre	SAGA User Group e.V
TATUKGIS	Outil de développement		Windows	Commerciale	Propriétaire	TatukGIS Compagny
QGIS	Logiciel	vecteur/raster	Linux/Windows	Gratuit	Libre	QGIS team
LANDSERF	Logiciel	vecteur/raster	Linux/Windows/Mac/Unix	Gratuit	-	Department of Information Science, City University, London
EPI	Logiciel	raster	Windows	Gratuit	-	INRA Montpellier et ATEN
FME	Logiciel	-	Windows	Commerciale	-	Geo Task

**Tab4.3** Comparaison des principales solutions SIG

## 6. SIG ET INTERNET

A l'ère de TIC (Technologies de l'Information et de la Communication), l'information est de plus en plus disponible pour un nombre croissant d'utilisateurs. Internet a révolutionné les outils de communication et est devenu un moyen essentiel pour l'échange et le partage de différents types de données autour du monde. Réservé à ses débuts, pour des applications locales et pour des utilisateurs particuliers, les données spatiales se sont maintenant généralisées à des géo-applications complexes, locales et en ligne, et à un nombre étendu d'utilisateurs. En effet, un ensemble de concepts combinant à la fois le SIG et Internet ont été inventé tels que WebGIS (SIG Web), Internet-Based GIS (SIG sur Internet), GIS Web mapping (cartographie SIG sur le Web), etc. De ce fait, plusieurs modèles 2D et 3D sont maintenant accessibles sur Internet offrant divers services spatiaux tels que le calcul d'itinéraires, l'optimisation de parcours, la recherche d'adresses, la simulation, les villes 3D, etc.

Vu les progrès technologiques d'Internet en termes de transfert et de diffusion de données, standardisation, Internet haut débit, entre autres, il est désormais possible de déployer sur Internet des applications spatiales avancées y compris les applications SIG. On distingue différents types de données spatiales sur Internet telles que les données cartographiques, image satellite, données de

réalité virtuelle, données 3D, etc. Il est très considérable de distinguer les SIG Web et les sites Web fournissant des services spatiaux sur Internet. Un SIG Web est le déploiement de toutes les fonctionnalités d'un SIG sur Internet y compris le stockage, l'affichage et l'analyse de données. Alors qu'un site internet fournissant des services spatiaux est un ensemble d'outils permettant d'utiliser et manipuler en ligne des données géo-référencées tels que "Google Maps" [GOOGLEMAPS, 2010], "via Michelin" [VIAMICHELIN, 2010], "Pages Jaunes" [PAGESJAUNES, 2010], "National Geographic" [NATIONALGEOGRAPHICS, 2010], "Géo-portail français" [IGN, 2010], "MapGuide" [MAPGUIDE, 2010], etc.

Pendant la dernière décennie, le concept de SIG Web est devenu omniprésent dans le WWW (World Wide Web). Mais, vue la complexité des données spatiales, la majorité des applications en ligne fournit principalement des modèles 2D. Selon les objectifs du SIG Web, on distingue deux types de services : i) Services gouvernementaux ou organisationnels tels que la gestion du patrimoine, l'étude de l'occupation des sols, l'étude et le contrôle des ressources et risques naturels, etc. ii) Services commerciaux ou privés tels que l'étude du marché et de la concurrence, le géomarketing, l'étude de la dispersion de la clientèle, etc. "DGO4" [WALLONIE, 2010] est un SIG Web 2D qui consiste en un assemblage de couches et fournissant une variété de services s'articulant autour de l'aménagement et l'urbanisme, logement, énergie et patrimoine. Un ensemble de données thématiques et statistiques sont associées aux données spatiales proposées par DGO4. "Pasco Internet GIS" [ESRI, 2009] est une application basée sur la technologie ESRI. Il soumet plusieurs opérations SIG telles que la recherche d'éléments par la sélection d'une couche et la spécification de la valeur d'un attribut, afficher la table attributaire pour les éléments spécifiés et calculer des valeurs, créer des graphiques et des diagrammes pour les objets sélectionnés, etc. [WEBGIS.NET, 2010] est un SIG Web 2D américain pour les Etats de Virginia et North Carolina. Il suggère des fonctionnalités SIG sur des cartes 2D représentant les villes de chaque Etat en termes de bâtiments, parcelles, zones d'inondation, images aériennes, etc. ainsi qu'un ensemble de fonctionnalités SIG telles que l'interrogation des cartes pour chercher des objets, la mesure des distances, la sélection et l'identification des objets, la création des couches spatiales selon différentes thématiques, etc. Main Street GIS [MAINSTREET, 2009] est une solution SIG Web qui consiste en un ensemble de couches de cartes et un visionneur 2D. Il propose un ensemble d'outils pour la mise en échelle, la navigation et l'interrogation de données spatiales, impression et export de données, etc.

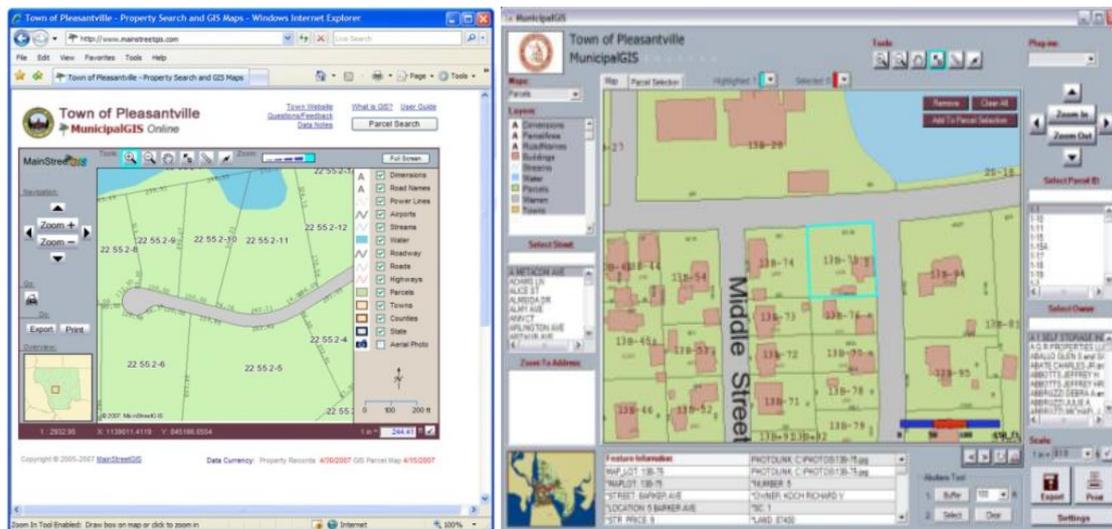
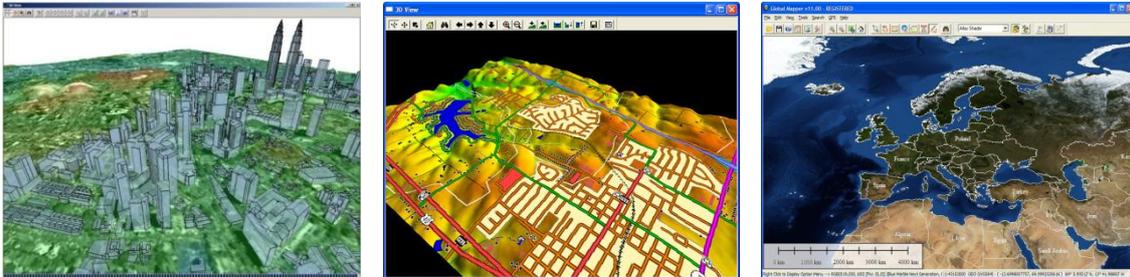


Fig4.11. Le SIG Web pour les municipalités [MAINSTREET, 2009]

Sur Internet, la liste des SIG Web 2D proposant une variété de services est très élargie. Cependant, les SIG Web 3D ne sont pas encore très répandus. Les SIG Web 3D utilisent différentes techniques pour modéliser et afficher les objets spatiaux telles que VRML (Virtual Reality Markup Language), X3D (eXtensible 3D), Web services, etc. "Apia Antica Project" [APIA, 2009] est un SIG Web pour le parc archéologique de Rome. Il fournit des modèles 3D pour des bâtiments historiques projetés sur un MNT (Modèle Numérique de Terrain). "Mantova 3D terrain and buildings" [ABACOGROUP, 2009] est un SIG Web offrant la possibilité d'afficher des bâtiments 3D texturés. Il propose quelques

fonctionnalités de mesure de distance, de surface et de volume. L'internaute peut aussi choisir une thématique pour afficher des objets spatiaux (bâtiments, monuments, etc.). "O3DG (Open3DGIS)" [OPEN3DGIS, 2009] est projet open source, créé par OpenGEO, permettant d'afficher sur Internet des objets 3D provenant des BD géographiques (PostgreSQL/PostGIS). L'idée du projet consiste à intégrer de manière simple des SIG 2D avec une visualisation et une interactivité 3D sur Internet. "Global mapper" [GLOBALMAPPER, 2010] est un ensemble d'outils SIG Web pour l'affichage et le traitement de données 3D. La figure (Fig4.12) présente quelques captures d'écran de l'outil "Global mapper".



**Fig4.12.** Quelques applications de l'outil "Global mapper" [GLOBALMAPPER, 2010]

- (a) Un MNT (Modèle Numérique du Terrain) et des bâtiments 3D
- (b) Un MNT avec une représentation vectorielle des rues
- (c) Une image rendue à partir d'un WMS (Web Map Service)

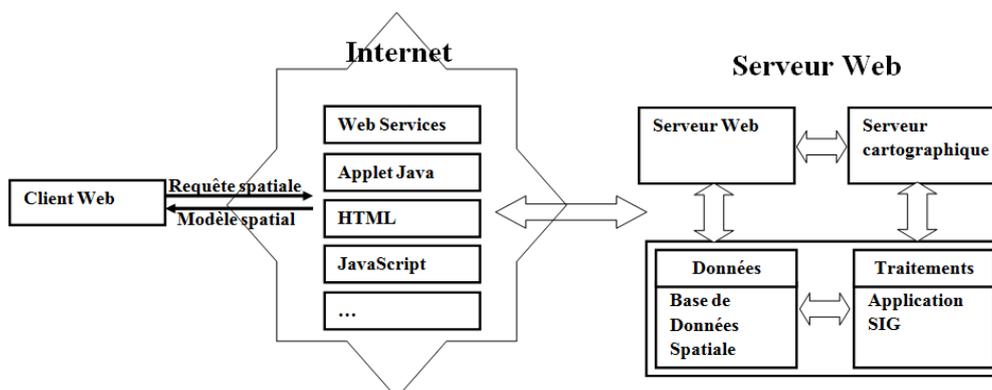
Cette partie a présenté un aperçu de quelques solutions SIG Web et non SIG Web, 2D et 3D, qui existent dans le WWW et proposant une variété de services spatiaux.

A l'issue de ce petit aperçu des SIG Web existants, nous proposons une comparaison entre tels SIG Web tel que c'est décrit dans le tableau (Tab4.4) suivant.

SIG Web	Dimensions des données	Types de données	Formats de données	Technologies Internet utilisées
DGO4	2D	Vecteur	Données ESRI : - <i>ShapeFile</i> - Geodatabase	- Origines de données : Cartes Navtech. - SGBD : Oracle - Langages : JavaScript, XHTML, etc.
Pasco Internet GIS	2D	Vecteur	<i>ShapeFile</i> , SDE data, fichiers BMP, fichiers TIFF	- Serveur cartographique : MapServer. - Visionneur : Applet Java
WebGIS.Net	2D	Vecteur + images aériennes	Données ESRI ( <i>ShapeFile</i> , Geodatabase)	- SGBDS : MS SQL Server. - Visionneur : ArcCAD, ArcInfo, ArcView. - Outil de développement : MapObjects, PHP, CSS and XHTML
Apia Antica Project	2D/3D	Vecteur	VRML (3D)	- Visionneur : Plugin osg4Web sur Firefox. - Bibliothèques : OpenSceneGraph, Virtual Terrain, etc.
Mantova 3D terrain and buildings	2D/3D	Vecteur/ raster	Arc/Info ASCII Grid (.grd, .asc), Arc/Info Binary Grid (.adf), GeoTiff (.tif), USGS ASCII DEM (.dem), Surfer Grid (.grd)	- Visionneur : Applet Java - Serveur : Oracle Spatial
O3DG	3D	Vecteur	Données provenant d'une BD PostgreSQL/PostGIS	- SGBD : PostgreSQL/PostGIS - Outil de développement : Map Server
Global- Mapper	2D/3D	Vecteur/ raster	GeoPDF, DLG, DRG, DOQ, DEM, DGN, DTED, <i>DWG</i> , <i>DXF</i> , <i>ShapeFile</i> , etc.	- Origine de données : Terra-Server USA, WMS, etc. - Visionneur : Global Mapper Viewer

**Tab4.4** Un petit comparatif entre des SIG Web

Nous nous inspirons de [HELALI, 2001] pour définir une architecture typique simplifiée d'un SIG Web dans la figure (Fig4.13).



**Fig4.13.** Architecture simplifiée d'un SIG Web

Cette architecture révèle trois composantes principales d'un SIG Web : le serveur web, le réseau Internet et le client Web. Le serveur Web accède aux données spatiales ainsi qu'à l'application SIG

Web directement ou à travers un serveur cartographique tel que MapServer, ArcGIS server, etc. Pour rendre un SIG Web accessible aux clients Web, plusieurs techniques sont utilisées telles que les Web services, les applets Java, les applications JavaScript, etc. De l'autre côté, l'utilisateur exprime ses besoins à travers une requête spatiale. La réponse du système peut être matérialisée par un nouveau modèle spatial ou la mise à jour d'un modèle spatial déjà affiché à l'écran de l'utilisateur.

### CONCLUSION

La puissance d'un SIG émane essentiellement de sa capacité à faire cohabiter plusieurs types de données (spatiales et non spatiale) dans un contexte spatial et d'en tirer des interprétations, d'élaborer des conclusions, de dégager des relations entre les entités spatiales, etc. Les SIG ont pu résoudre des problématiques liées à la répartition géographique des objets du monde réel. Un SIG est capable de répondre aux questions relatives aux localisations, thématiques et même l'évolution temporelle des objets et des phénomènes spatiaux. Un SIG peut alors révéler des nouvelles informations pertinentes pouvant conduire à des prises de décisions élaborées et appropriées.

Les fonctionnalités du SIG ainsi que les besoins des utilisateurs doivent être bien définies et délimitées. Pour un usage étendu, un SIG doit être conforme, en termes de données spatiales, aux normes nationales et internationales.

Comme le SIG est un domaine multidisciplinaire (faisant appel à des disciplines très variées tant au niveau des données qu'au niveau des traitements spatiaux), ce chapitre a essayé de situer le SIG par rapport aux concepts et outils utilisés dans le contexte spatial. Quelques définitions sont données pour expliquer quelques disciplines utilisées dans les SIG. Après, le SIG est situé par rapport à l'open source et les outils commerciaux pour donner un aperçu sur les outils existants. Et comme les méthodes de conception des SIG sont très rares, nous citons une seule méthode appelée MECOSIG (Méthode de COnception de SIG) a été proposée par [PANTAZIS, *et al*, 1996]. MECOSIG est principalement une méthode de conception de SIG informatisés. Elle constitue un ensemble de principes, de démarches et d'outils théoriques (notamment des formalismes de modélisation) pour la mise en place des SIG.

Ce chapitre achève la section "*Etat de l'art*". Les prochains chapitres du présent mémoire décriront la mise en place d'un SIG accessible via Internet illustré par des exemples commentés de développement.

### SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

#### INTRODUCTION GENERALE DE LA SECTION II

Dans le chapitre 1 "*Définition des besoins*", nous avons abordé les problématiques liées aux données spatiales que nous utilisons dans le cadre de ce SIG. En effet, nous avons tout d'abord fixés plusieurs hypothèses relatives aussi bien aux données qu'aux traitements spatiaux. En particulier, nous supposons que les données sources sont géométriquement correctes et que notre prototype de SIG ne disposera pas dans sa première version des fonctionnalités de correction des géométries invalides. Les données traitées sont vectorielles. L'idée initiale consistait à intégrer ces données vers une base de données spatiale (BDS) stockée dans le Système de Gestion de Bases de Données Spatial (SGBDS) PostgreSQL (on utilisera des outils d'intégration libres tels que *shp2pgsql*), pour ensuite les utiliser en l'état dans le SIG. Les premiers tests ont montré quelques limites au niveau de l'affichage et de l'analyse (manque de données topologiques et surtout de données attributaires dans les fichiers initiaux). Nous avons ainsi opté pour une phase de modélisation puis d'intégration de données spatiales provenant de ces fichiers sources. Cela fera l'objet de cette section qui s'intitule "*Modélisation et intégration des données spatiales dans le SIG*".

L'étape de modélisation n'aura pas pour objectif le changement mais plutôt l'optimisation de leur structure, d'une part, et leur enrichissement par d'autres types de données, d'autre part.

Les objectifs principaux de la modélisation seront alors :

- Utiliser des primitives géométriques compatibles avec celles utilisées par les fichiers sources pour la représentation vectorielle des données spatiales.
- Représenter la géométrie d'une manière facilitant son interrogation et son rendu.
- Comme les fichiers sources que nous utilisons ne permettent pas jusqu'à présent le stockage des solides 3D (dont nous avons besoin pour la représentation des bâtiments), la modélisation doit permettre le stockage et l'interrogation de tels solides.
- Proposer une structure facilitant l'enrichissement des données géométriques par des données thématiques pour les capacités du SIG en termes d'interrogation et d'analyse.

L'étape d'intégration consistera alors à transformer les fichiers sources vers un format de base de données plus facile à interroger. En effet, nous nous basons sur des outils d'intégrations libres de ces fichiers, en particulier l'outil "*shp2pgsql*". Notre outil SIG doit proposer une méthode simple pour associer des données attributaires aux données géométriques obtenues. Une structure particulière de base de données va être alors mise en place pour le stockage de données spatiales et attributaires utiles pour notre prototype de SIG.

Cette section est composée de deux chapitres. Le premier intitulé "*Modélisation des données spatiales pour le SIG 3D*" reprend nos travaux en termes de modélisation de données spatiales effectuée à travers les trois niveaux de modélisation : conceptuel, logique et physique. Le deuxième qui s'intitule "*Intégration & création des jeux de données spatiales*" décrit la méthode d'intégration des données spatiales.



CHAPITRE.5 MODELISATION DES DONNEES SPATIALES POUR LE SIG 3D

Résumé.....96

Introduction .....96

**1. Modélisation conceptuelle de données**.....96

1.1. Modélisation géométrique.....97

1.1.1. Liste de critères .....97

1.1.2. Représentation des objets 3D .....98

1.1.2.2. Modélisation surfacique .....99

1.1.2.3. Modélisation volumique.....101

1.1.2.4. Notre choix : Modélisation hybride.....102

1.2. Modélisation topologique.....105

1.2.1. Deux types de topologies .....105

1.2.2. Choix topologiques.....105

1.2.3. Modélisation de la topologie .....106

1.3. Problématique de modélisation attributaire.....108

**2. Modélisation logique** .....108

2.1. Différents types de modélisation logique .....108

2.2. Comparaison et choix du modèle logique .....108

2.3. Représentation du modèle logique .....110

**3. Modélisation physique** .....112

3.1. Choix du SGBD .....112

3.2. Base de données spatiale multi-niveaux.....113

3.3. Représentation du modèle physique.....114

3.4. Optimisation.....115

Conclusion.....116

*« Isolé, le modéliste a peu de chances de succès: sa réalisation risque de s'écraser au premier vol. Au sein d'un groupe qui le conseille et le guide, la réussite est beaucoup plus assurée »*

(SCIENCE ET VIE, juin 1981, n° 135, p.137)

### RESUME

*La représentation 3D des objets spatiaux dans un SIG a beaucoup évolué. La modélisation spatiale peut porter sur la géométrie, la topologie ainsi que les propriétés attributaires d'objets spatiaux. Alors que la modélisation 2D a atteint un niveau de maturité très avancé, la modélisation 3D de données spatiales reste toujours une des problématiques fondamentales des SIG. C'est dans ce cadre que ce chapitre propose une modélisation géométrique et topologique des objets spatiaux à travers les trois niveaux de modélisation de la base de données spatiale (BDS): conceptuel, logique et physique.*

### INTRODUCTION

La modélisation de données est une phase préliminaire et obligatoire avant la mise en place de tout SI(G). Il s'agit d'une représentation abstraite des objets à modéliser ainsi que des relations sémantiques entre de tels objets. La modélisation se base, usuellement, sur des formalismes définis au sein d'une approche ou d'une méthode de modélisation. Malgré l'existence de plusieurs méthodes consacrées à la modélisation des données géographiques ainsi que des Bases de Données Spatiales (BDS) telles que la méthode MODUL-R [BEDARD, *et al*, 1992], CONGOO (CONception Géographique Orientée Objet) [PANTAZIS, *et al*, 1996], Geo-ER [HADZILACOS, *et al*, 1997], il n'existe pas de méthodes normalisées pour la modélisation des SIG. Un concepteur de données spatiales pour des BDS ou SIG a donc plusieurs solutions. En effet, il peut adopter une méthode parmi celles proposées par les communautés des chercheurs dans le domaine spatial parmi celles proposées par [BEDARD, *et al*, 1992], [PANTAZIS, *et al*, 1996], [HADZILACOS, *et al*, 1997], [TRYFONA, *et al*, 1999], [BOHLEN, *et al*, 1998], [RENOLEN, 2000], [WANG, *et al*, 2001]. Il peut également adapter une méthode entité-association (EA) telle que MERISE ou une méthode orientée-objet telle que le langage UML (Unified Modeling Language) ou le couplage des deux méthodes pour modéliser des données spatiales.

Dans ce contexte, ce chapitre présente nos travaux en termes de modélisation de données spatiales 3D. Pour modéliser la BDS, les trois phases essentielles de conception sont respectées. En effet, la première partie présente la modélisation conceptuelle tant pour les données géométriques que pour les données topologiques. La deuxième partie constitue une traduction du modèle conceptuel défini dans la première section en un modèle logique. La troisième partie présente le modèle physique qui a pour but la traduction du modèle logique en une BDS selon les spécifications définies par le Système de Gestion de Base de Données (SGBD) PostgreSQL avec sa couche spatiale PostGIS.

### 1. MODELISATION CONCEPTUELLE DE DONNEES

Pourquoi la modélisation conceptuelle ? Qu'est-ce-que l'on va modéliser ? Quel sera le résultat de la modélisation ? Ce sont des questions typiques que l'on peut se poser avant de modéliser tout SI. La modélisation conceptuelle est un processus capable de formaliser, graphiquement et/ou textuellement, les données qui seront utilisées par le SI sous la forme d'un ensemble de concepts reliés par des relations logiques ou sémantiques. Il s'agit donc d'une représentation de données, facilement compréhensible et permettant de décrire le contenu du SI.

Dans le contexte du SIG, la modélisation conceptuelle porte essentiellement sur les données géométriques et topologiques. La modélisation repose principalement sur des exemples de bâtiments,

mais elle pourrait être étendue à la modélisation de tout objet spatial pouvant être représenté par des polygones frontières.

Malgré l'existence d'une multitude d'approches proposant différents formalismes relationnels et objets pour la modélisation géographique, nous allons suggérer une modélisation simplifiée de ces données. Dans la littérature, toutes les définitions de la modélisation conceptuelle incitent le concepteur à éviter toute considération technologique lors de la modélisation conceptuelle. Ce n'est pas le cas dans ce travail. En effet, le choix d'un SGBDS (spatial) que nous allons utiliser pour le stockage de données du SIG est relativement restreint. Plusieurs critères, expliqués dans la section « modélisation physique », ont motivé le choix du SGBD objet-relationnel PostgreSQL avec sa couche spatiale PostGIS pour le stockage de la BDS. Plusieurs considérations conceptuelles (les primitives géométriques et leur représentation dans le SGBDS, le stockage des solides, etc.) seront influencées par ce choix.

### 1.1. MODELISATION GEOMETRIQUE

La géométrie des objets spatiaux doit être bien décrite aussi bien pour le stockage de données spatiales que pour la représentation et l'interrogation de tels objets. Elle consiste à décrire les formes des objets réels ou imaginaires à travers un ou plusieurs modèles virtuels. Un objet spatial peut être simple ou complexe, bidimensionnel ou tridimensionnel. Dans un modèle virtuel et pour plus de réalisme, la modélisation géométrique est généralement complétée par des informations sur les apparences physiques (couleur, lumière, textures, matériaux, etc.), des informations temporelles (évolution des objets dans le temps), des informations attributaires, etc. D'un point de vue purement conceptuel, la modélisation géométrique des objets spatiaux, en particulier des objets urbains, différencie deux types de modélisation : la modélisation *a priori* et la modélisation *a posteriori*. La modélisation *a priori* consiste à construire un modèle virtuel pour un objet urbain n'ayant pas encore une existence physique, une maquette d'un immeuble pour un projet de construction par exemple. Une fois mis en place, l'objet urbain réel aura une apparence géométrique fidèle aux propriétés définies par le modèle virtuel. Alors que la modélisation *a posteriori* consiste à mettre en place des modèles virtuels pour des objets ayant déjà une existence physique et pouvant avoir des propriétés géométriques très différentes. La modélisation géométrique, dans ce cas, doit pouvoir représenter les propriétés géométriques communes de tous les objets urbains. La modélisation *a posteriori* s'avère être beaucoup plus compliquée que celle *a priori*. La modélisation que nous allons décrire dans ce chapitre est la modélisation *a posteriori*.

#### 1.1.1. LISTE DE CRITERES

Avant de choisir une méthode de modélisation géométrique, il est essentiel de définir un ensemble de critères pour motiver nos choix, d'une part, et pour expliquer les caractéristiques les plus importantes des nos données, d'autre part. Les critères de choix peuvent porter sur la dimension des données à modéliser, la méthode de modélisation géométrique, la méthode de modélisation conceptuelle, etc. [BERNARD, *et al*, 2000] définissent un ensemble de critères de choix d'un modèle géométrique dont nous retenons les critères que nous jugeons les plus importants :

- L'étendue du domaine : le domaine d'une représentation doit être suffisamment large pour permettre à un ensemble utile d'objets physiques d'être représenté.
- Compacité : une représentation compacte permet d'optimiser l'espace mémoire nécessaire pour décrire l'objet.
- Efficacité : une représentation efficace est une représentation qui permet l'utilisation d'algorithmes efficaces pour calculer les propriétés physiques des objets et générer les images de synthèse.

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

- Potentiel topologique : ce critère évaluera la capacité d'une représentation à permettre d'effectuer des opérations topologiques sur des objets 3D décrits sous cette forme.
- Potentiel multi-échelle : une représentation répond à ce critère si elle permet facilement de gérer plusieurs types de représentations (niveaux de détails) des objets 3D décrits sous cette forme.

Nous complétons cette liste par quelques critères qui portent sur :

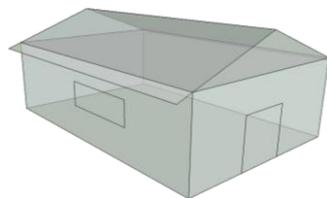
- Caractéristiques des objets : la modélisation porte-elle sur des objets naturels, artificiels ou une combinaison des deux ?
- Dimensions : La dimension des objets à modéliser peut être critique pour la méthode choisie. En effet, les besoins en termes de stockage et de traitements d'objets géométriques peuvent varier en fonction des dimensions de tels objets (par exemple, une même opération topologique retourne des résultats différents sur des objets 2D et objets 3D).
- Standards : positionner les primitives géométriques et topologiques utilisées par la méthode par rapport aux standards (surtout spécifications de l'OGC et l'ISO).
- SGBD spatiaux : est-ce que la méthode fournit des primitives géométriques et topologiques pouvant être stockées dans le SGBDS qui sera utilisé pour le stockage des objets spatiaux ?
- Extensibilité : une méthode est extensible si elle peut être modifiée pour répondre à des besoins particuliers.
- Simplification : la méthode peut-elle être simplifiée pour l'optimisation de stockage des objets géométriques ainsi que leur interrogation ?
- Intégration : la méthode peut-elle être intégrée avec une autre méthode pour la modélisation des objets géométriques.

Le concepteur d'une BDS peut se fixer des choix avant d'entamer le processus de conception selon les formalismes définis par telle ou telle méthode de modélisation 3D. Les critères définis dans cette section peuvent être élargis par d'autres critères selon les besoins attendus du modèle 3D.

### 1.1.2. REPRESENTATION DES OBJETS 3D

Avant de choisir une méthode de modélisation géométrique des objets spatiaux qui seront stockés dans la BDS, nous décrivons la représentation géométrique des objets tridimensionnels dans le SIG 3D développé dans le cadre de cette thèse. En particulier, un bâtiment peut être modélisé selon deux types de représentation complémentaires:

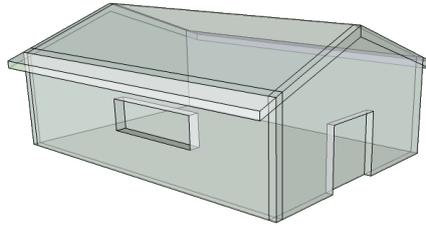
- Collection de surfaces : toutes les façades, toits et éléments de détails (portes, fenêtres, etc.) d'un bâtiment sont représentés par une collection de surfaces pouvant partager des parties communes (illustration dans *Fig5.1*). Outre les bâtiments, cette représentation peut être utile pour d'autres types d'objets tridimensionnels.



*Fig5.1.* Un bâtiment représenté en tant qu'une collection de surfaces

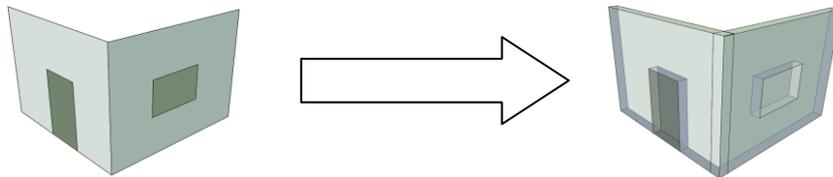
- Collection de solides : en réalité, les murs d'un bâtiment ont une épaisseur non négligeable. Telle épaisseur est facilement remarquable à travers les ouvertures (portes et fenêtres) qui

rentrent à travers les murs (illustration dans *Fig5.2*). Une telle représentation peut apporter une valeur ajoutée en matière de réalisme du modèle.



*Fig5.2.* Un bâtiment représenté en tant qu'une collection de solides

Le couplage de ces deux représentations permettra une représentation simplifiée des bâtiments qui peut modéliser le même bâtiment dans les différents niveaux de détail. Un mur d'un bâtiment peut donc avoir deux représentations différentes:



*Fig5.3.* Deux représentations des murs d'un bâtiment selon le niveau de détail

Pour passer d'une représentation à une autre, il faut extruder les surfaces correspondant aux murs. Nous avons développé un algorithme pour automatiser un tel changement de modèle. Selon les différents types de représentations décrites ci-dessus, un bâtiment peut être vu comme un ensemble de surfaces ou un ensemble de solides pour représenter les façades, toits et éléments de détails. Nous avons étudié alors quelques méthodes de modélisation surfaciques, volumiques ainsi que la possibilité de leur couplage dans une méthode hybride.

### 1.1.2.2. MODELISATION SURFACIQUE

Parmi les méthodes de modélisation surfaciques, nous avons étudié B-Rep et NURBS. Leur comparaison selon les critères de choix défini ci-dessus révèle le tableau (*Tab5.1*).

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

Méthodes surfaciques Critères de choix	B-Rep (Boundary Representation)	NURBS (Non-Uniform Rational Basis Splines)
L'étendue du domaine	(+) Flexible pour des objets plats, (-) Plus de surfaces pour des objets courbés.	(+) Flexible pour la représentation d'objets plats et courbés grâce aux points de contrôle et aux courbes B-Splines.
Compacité	(-) Stockage relativement important des points, lignes, surfaces et volumes, etc.	(-) Stockage très important des points de contrôle, poids, fonctions B-splinaires, surfaces, volumes, etc.
Efficacité	Algorithmes d'Euler, algorithmes conversion B-Rep-CSG, etc.	Algorithme NURBS, algorithme de Boor, algorithme de Casteljau, etc.
Potentiel topologique	Distingue la géométrie de la topologie, Formule d'Euler pour valider les topologies.	Utilisation des points de contrôle et des surfaces subdivisées pour des calculs topologiques.
Potentiel multi-échelle	(-) Plus de surfaces pour plus de détails.	(-) Plus de subdivision de surfaces pour plus de détails.
Caractéristiques des objets	(+) objets artificiel, (-) objets naturels.	(+) pour objets naturels et artificiels
Dimensions	2D et 3D	2D et 3D
Standards	(+) Les primitives géométriques de B-Rep peuvent être représentées selon les spécifications OGC et ISO.	-
SGBD spatiaux	(+) La plupart des SGBDS peuvent stocker des points, lignes et surfaces. (-) Il faut ajouter des modules pour le stockage des volumes	(-) Il faut ajouter des modules au SGBDS pour le stockage des courbes et surfaces NURBS
Simplification	(+) Plus intuitive, Peut être simplifiée (stockage des points, surfaces et volumes et interprétation des lignes)	(-) Moins intuitive, utilisation B-Splines
Intégration	(+) B-Rep peut être intégrée avec CSG	-

**Tab5.1** Comparaison des méthodes de modélisation surfacique B-Rep et NURBS

Etant donné que l'objectif de cette thèse est la mise en place d'un SIG 3D et vu que les objets à représenter sont principalement des objets artificiels, en particulier des bâtiments contenant généralement des murs plats, la plupart de tels objets peuvent être décrits en termes de surfaces composant leurs enveloppes. A l'instar des critères de choix vus dans le tableau (Tab4.1) ci-dessus, nous avons étudié les deux méthodes de modélisation surfacique 3D : B-Rep et NURBS. Depuis qu'elle utilise des primitives géométriques dont l'association conduit à former des volumes relativement simples, la modélisation B-Rep paraît plus intuitive que la modélisation par des surfaces NURBS représentées mathématiquement. Mais comme indiqué ci-dessus, le choix du SGBD PostgreSQL/PostGIS a également orienté le choix de la méthode de modélisation conceptuelle géométrique. En effet, PostGIS implémente les spécifications de l'OGC en termes de primitives géométriques utilisées telles que *POINT*, *LINESTRING*, *POLYGON*, *MULTILINESTRING*, *MULTIPOLYGON*, etc. Ces géométries peuvent être représentées par B-Rep. En utilisant les NURBS, il est possible d'atteindre un niveau de réalisme assez élevé pour la représentation d'objets complexes. Mais, leur implémentation nécessite la mise en place d'une structure particulière dans un SGBDS. Malgré le fait que B-Rep souffre de quelques inconvénients relatifs surtout à la précision (nécessité d'un maximum de surfaces pour une meilleure précision), cela peut être moins pénalisant une fois l'approche optimisée. L'avantage majeur de B-Rep est sa capacité à décrire à la fois les propriétés géométriques et topologiques d'un objet 3D. L'orientation des surfaces B-Rep peut apporter une valeur ajoutée en termes d'algorithmes appliqué à des objets 3D. Nous pensons ici aux algorithmes de calcul de volumes, pour tester si un point est à l'intérieur ou à l'extérieur d'un volume, etc. Les

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

nombreux avantages de B-Rep ont motivé son adoption pour la modélisation des objets spatiaux de notre SIG.

### 1.1.2.3. MODELISATION VOLUMIQUE

Parmi les méthodes de modélisation volumiques, nous avons étudié CSG, SPR et Octree. Leur comparaison selon les critères de choix défini ci-dessus permet d'établir le tableau (*Tab5.2*) suivant.

Méthodes volumiques Critères de choix	CSG (Constructive Solid geometry)	SPR (Spatial Partitioning Representation)	Octree
L'étendue du domaine	(+) Objets 3D précis et complexes	(-) Approximation des objets 3D à l'aide des voxels	(+) Subdivision récursive de l'espace pour représenter des objets 3D
Compacité	(+) Stockage des primitives solides (arbres, géométrie, localisation, orientation, etc.). (-) Les arbres peuvent devenir très rapidement complexes	(-) Stockage des voxels (qui peut être rapidement important pour des objets complexes)	(-) Stockage des arbres Octrees, position et tailles des octants, etc.
Efficacité	Algorithmes de conversion CSG-B-Rep, SCS (Sequenced Convex Subtraction), etc.	-	Algorithme d'Octree
Potentiel topologique	(-) CSG ne considère pas les relations topologiques	-	(+) Interprétation des relations d'adjacence à partir des arbres Octree
Potentiel multi-échelle	(+) Représentation 3D précise multi-échelle	(-) Représentation 3D approximative multi-échelle	(+) Représentation 3D précise multi-échelle
Caractéristiques des objets	(+) artificiels, (-) naturels	(+) artificiels, (-) naturels	(+) artificiels, (-) naturels
Dimensions	3D	3D	3D
Standards	-	-	-
SGBD spatiaux	(-) La plupart des SGBDS ne permettent pas le stockage de solides CSG	(-) Il faut des structures spécifiques pour le stockage des voxels	(-) Il faut des structures pour le stockage des arbres Octree.
Extensibilité	-	-	-
Simplification	(+) Simplification de l'arbre CSG en des sous-arbres	-	(+) Algorithmes de simplification Octree
Intégration	(+) CSG peut être intégrée avec B-Rep	-	(+) Octree peut être intégré avec CSG. Octree est une variante de SPR

**Tab5.2** Comparaison des méthodes de modélisation volumique CSG, SPR et Octree

⇒ La modélisation surfacique B-Rep est capable, même approximativement, de représenter des objets tridimensionnels. Mais, il est important dans un modèle 3D d'atteindre un certain niveau de réalisme. En particulier, les façades des bâtiments peuvent être représentées par des solides (illustration décrite dans *Fig5.2*). Pour ce faire, nous avons étudié les méthodes de modélisation volumique : CSG, SPR et Octree pour la modélisation de telles façades. CSG se base sur des primitives solides telles que les

cubes, les sphères, les prismes, les cônes, etc. associées à l'aide d'un arbre d'opérations booléennes : intersection, union et différence. Elle permet une représentation précise des objets complexes. Cependant, cette méthode nécessite le stockage des informations sur les primitives solides (géométries, positions, orientations) ainsi que l'arbre d'opérations booléennes, pouvant devenir rapidement volumineux pour des scènes complexes. Mais, l'association de cette méthode avec une autre méthode de modélisation d'objets tridimensionnels peut fournir une valeur ajoutée en termes de modélisation.

⇒ SPR utilise des voxels adjacents et ne s'intersectant pas pour la modélisation d'objets 3D. SPR est avantageuse du point de vue simplicité. Elle permet une distinction facile de l'intérieur et l'extérieur d'un solide, un calcul approximatif du volume du solide, etc. Cependant, on peut atteindre facilement un nombre important de solides primitifs élémentaires pour la représentation d'un objet tridimensionnel. L'apparence visuelle des solides modélisés par cette méthode n'est pas très réaliste. Ces inconvénients rendent cette méthode inappropriée pour la modélisation des objets spatiaux de notre SIG, les bâtiments en particulier.

⇒ Les Octrees sont utilisés pour la modélisation des objets tridimensionnels à travers une subdivision récursive de l'espace (à travers des voxels). Telle répartition permet une flexibilité de placement et de construction de volumes en utilisant un parcours simple. Toutefois, les arbres utilisés pour la distribution des objets ne sont pas forcément équilibrés ce qui pourra alourdir le parcours pour des scènes complexes. Cette méthode nécessite également un espace de stockage important pour des scènes complexes.

⇒ A l'instar de ces critères de choix, CSG pourrait être une approche très appropriée pour la modélisation des façades d'un bâtiment à travers des primitives solides simples sur lesquelles on peut appliquer quelques opérations booléennes pour ajouter des détails tels que les portes, les fenêtres, etc. (les ouvertures peuvent être obtenues en appliquant une opération booléenne de "*différence*" entre la géométrie correspondant à une façade et la géométrie correspondant à une porte).

### 1.1.2.4. NOTRE CHOIX : MODELISATION HYBRIDE

A l'issue de la représentation géométrique des objets spatiaux ainsi que des critères de choix (décrits ci-dessus), nous avons étudié quelques méthodes de modélisation géométriques surfaciques et volumiques (parmi celle décrites dans le premier chapitre de ce mémoire) pour en tirer la ou les méthodes les plus appropriées pour la modélisation des objets 3D du SIG. B-Rep utilisant des primitives pouvant être représentées selon des spécifications de l'OGC, est capable d'approximer tout objets 3D du monde réel. PostGIS peut stocker les primitives géométriques, à l'exception des solides, de la méthode B-Rep car il se base à son tour sur des spécifications OGC. Les bâtiments en particulier, peuvent être représentés par ce type de modélisation. Néanmoins, avec plus de détails sur tels bâtiments, le volume de données B-Rep peut être assez important. Le couplage de B-Rep avec CSG peut être intéressant en termes de représentation et optimisation de stockage de tels bâtiments. Les avantages de couplages des deux approches dans le contexte de bâtiments peuvent porter sur :

- Précision : B-Rep indique des informations sur toutes les surfaces du solide. Cela peut être très important pour la gestion de tel solide. CSG permet la création de volumes précis avec des primitives solides.
- Niveaux de détails : Le couplage de B-Rep et CSG permet une représentation à travers un ensemble de niveaux de détails (CSG pour ajouter plus de détail à B-Rep).
- Simplification : Vu que les surfaces B-Rep que nous utilisons sont orientées, une surface pourra être représentée uniquement par ses sommets et les arêtes seront interprétées par l'association des sommets deux par deux. Les solides CSG représentant les façades sont

obtenus par extrusion des surfaces B-Rep, donc les informations concernant la position ainsi que l'orientation des volumes CSG ne seront pas stockées. Les opérations booléennes nécessaires pour l'assemblage des primitives solides ne sont pas également stockées car telles primitives sont obtenues par extrusion des surfaces B-Rep qui sont déjà bien positionnées et associées pour la construction des objets 3D. Les opérations booléennes se limitent sur les opérations de différence entre solides pour construire des ouvertures sur les façades des bâtiments.

- Stockage : Pour les surfaces B-Rep, uniquement les sommets seront stockés. Pour les volumes CSG, une seule information sera stockée, il s'agit de l'épaisseur du solide (correspondant à un mur du bâtiment). Un solide est interprété à partir de la surface B-Rep et l'épaisseur correspondante.
- Passage d'une présentation à l'autre : Un algorithme basé sur l'extrusion est décrit dans le chapitre 6 "*Représentation des données spatiales dans le SIG 3D*" pour expliquer le passage d'une représentation à l'autre.

La modélisation hybrides peut apporter une valeur ajoutée en termes de :

- Un utilisateur peut tester "visuellement" ses données à travers les différents niveaux de détail en utilisant le SIG comme visionneur de données. Des fonctionnalités de vérification de validité géométriques peuvent être ajoutées.
- Le passage d'une représentation à l'autre se fera d'une manière automatique et transparente à l'utilisateur ce qui facilitera ses tâches.

Le schéma conceptuel du modèle géométrique est expliqué dans le diagramme de classes suivant (Fig5.4). Par rapport au modèles géométriques existants et surtout par rapport à l'ISO, le modèle conceptuel que nous proposons respecte la même structure avec une légère modification. En effet, les primitives géométriques utilisées sont principalement les *points*, *lignes*, *surfaces* et *solides* ainsi que l'association de telles primitives pour avoir des géométries complexes telles que les *muti\_surfaces*, les *multi\_solides*, etc. Vu que les fichiers sources mis à notre disposition (*Shapefile*, *DWG*, etc.) représentent les surfaces à travers un ensemble ordonnés de sommets (sens trigonométrique ou sens inverse), nous simplifions le modèle en éliminant les liens et les lignes et en ajoutons un lien entre les surfaces et les points. Une surface sera définie par des sommets ordonnés permettant d'interpréter les lignes de contour en les associant deux par deux. Nous ajoutons aussi un lien entre les solides et les sommets qui les composent. Un solide sera représenté à la fois par les surfaces et les sommets qui le composent.

Les avantages de cette simplification apportent une valeur ajoutée en matières de :

- Lignes : L'entité ligne peut être utilisée par exemple pour la représentation d'objets spatiaux de type réseau tels les réseaux des rues, d'électricité, etc.
- Surfaces : Le rendu des surfaces devient plus rapide car on va éliminer les jointures "surface-ligne" et "ligne-point".
- Solides : La jointure "solide-surface" est utile pour la représentation des enveloppes des solides tels que les bâtiments. Alors que la relation "solide-point" sera utile pour le calcul des relations topologiques entre les solides. Cela est repris en détail dans le chapitre 8 "*Interrogation des données spatiales*".

Les liens qui seront éliminés du modèle sont représentés en rouge et les liens à ajouter sont représentés en vert dans la figure (Fig5.4) du modèle conceptuel.

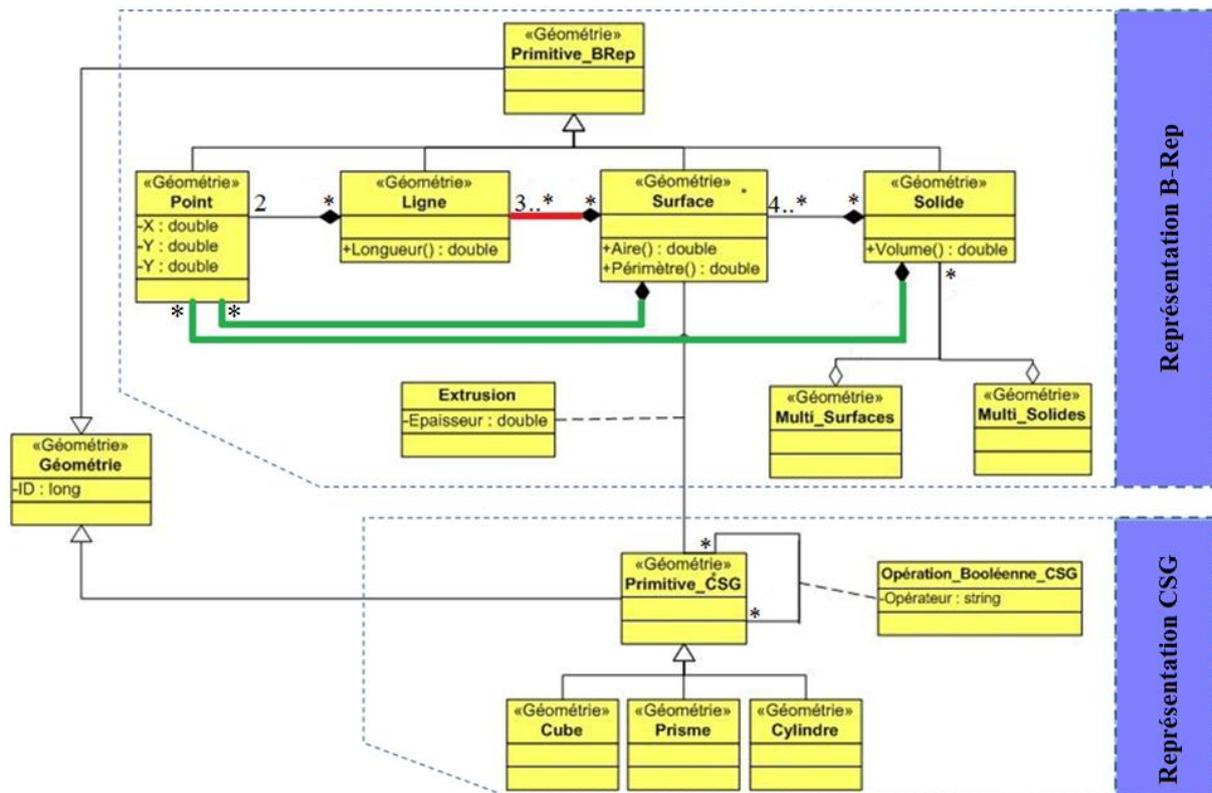


Fig5.4. Modèle conceptuel géométrique

Les primitives CSG (cube, prisme et cylindre) sont obtenues par extrusion des surfaces B-Rep selon une épaisseur donnée. Que ce soit des primitive B-Rep ou des primitives CSG sont utilisées, le modèle conceptuel doit pouvoir représenter tout objet tridimensionnel. L'algorithme de passage d'une représentation B-Rep vers une représentation CSG est expliqué dans le chapitre 6 « *Présentation des données spatiales dans la SIG 3D* ».

La modélisation conceptuelle décrite dans cette section ne se limite pas que sur des objets urbains tels que les bâtiments. En effet, tout objet 3D pouvant être représenté en termes des surfaces composant son enveloppe est capable d'être représenté selon une modélisation B-Rep. Donc, l'approche B-Rep est utilisée pour la modélisation de tout objet tridimensionnel, alors que le couplage de B-Rep avec CSG se limite, dans notre cas, aux bâtiments pour l'ajout de détails sur les façades et les toits.

Le tableau (Tab5.3) suivant compare de B-Rep, B-Rep modifiée et B-Rep+CSG en terme de nombre de primitives géométriques utilisées pour la modélisation des bâtiments des figures (Fig5.1 et Fig5.2).

	B-Rep	B-Rep modifiée	B-Rep + CSG
Points	36	36	36
Lignes	36	0	0
Surfaces	9	9	9
Solides	1	1	1
Epaisseur	0	0	7
<b>Total</b>	82	46	53

Tab5.3 Comparaison du stockage de données pour les bâtiments des figures (Fig5.1 et Fig5.2)

En comparant B-Rep et B-Rep modifiée utilisées pour la modélisation surfacique du bâtiment de la figure (Fig5.1) en termes de nombre de primitives utilisées, on remarque que l'omission des relations "surface-ligne" et la création d'une relation directe "surface-point" a beaucoup optimisé le nombre de

primitives stockées. Concernant l'interprétation d'une représentation B-Rep + CSG du même bâtiment, nous ajoutons une information sur l'épaisseur pour chaque surface représentant un mur. Le solide correspondant au bâtiment sera stocké, alors que les solides correspondant aux murs du bâtiment seront interprétés automatiquement à partir des surfaces et les valeurs d'épaisseur correspondantes.

## 1.2. MODELISATION TOPOLOGIQUE

Un SIG doit permettre des fonctionnalités topologiques pour l'étude des relations de connectivité entre les objets spatiaux. La topologie doit être formalisée à son tour à l'aide d'un modèle conceptuel.

### 1.2.1. DEUX TYPES DE TOPOLOGIES

Nous différencions deux types de topologie :

- Topologie simple : une topologie simple exprime les relations de connectivité entre des primitives topologiques simples pour former des solides. En effet, on distingue quatre types de primitives topologiques : *nœud*, *arête*, *face* et *volume*. Une *arête* est définie par deux *nœuds* extrémités, une *surface* est définie par au minimum trois *arêtes* formant son contour. Un *volume* est défini par au minimum quatre *faces* composant l'enveloppe extérieure. Dans la littérature, il existe plusieurs modèles topologiques implémentant cette structure tels que [WEI, *et al*, 1998], [ISO, 1999], [ZLATANOVA, 2000], [HOEL, *et al*, 2003], [LACHANCE, *et al*, 2006], [ELLUL, *et al*, 2008], etc.

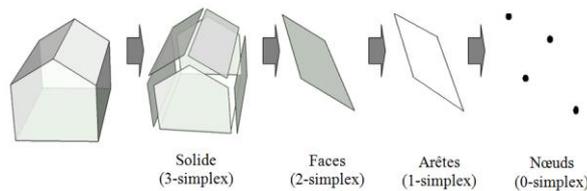


Fig5.5. Les différentes primitives topologiques simples

- Topologie complexe : Une topologie complexe exprime des relations de connectivité entre des objets 3D simples ou complexes (vérifier par exemple si deux bâtiments se touchent ou non). Ce type de topologie a fait, à son tour, l'objet de plusieurs travaux de recherche tels que [ZLATANOVA, 2000], [HOEL, *et al*, 2003], [LACHANCE, *et al*, 2006], [ELLUL, *et al*, 2008], etc.

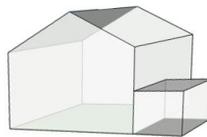


Fig5.6. Exemple de topologie entre deux solides qui se touchent

### 1.2.2. CHOIX TOPOLOGIQUES

Vu l'importance de la topologie dans les SIG, d'une part, et vu que le calcul topologique constitue une fonctionnalité parmi celles proposées par notre prototype de SIG, d'autre part, nous avons décidé d'intégrer la topologie avec la géométrie lors du processus de modélisation conceptuelle.

Mais avant de faire l'intégration et pour des raisons d'optimisation, nous avons fixé des objectifs portant sur :

- Topologie structurelle : Dans un SIG, deux types principaux de topologie peuvent être distingués : topologie structurelle (décrivant la structure d'un objet) et topologie de réseaux

(décrivant un réseau, les rues par exemple). Dans cette thèse, nous ne nous intéresserons qu'à la topologie structurelle.

- Cohérence entre géométrie et topologie : la topologie ne doit pas être contradictoire avec la géométrie. En effet, les primitives géométriques doivent correspondre à des primitives topologiques valides. Les primitives géométriques B-Rep, utilisées dans le modèle conceptuel géométrique, peuvent correspondre aux primitives topologiques mentionnées ci-dessus (voir *Tab3.1* de correspondance géométrie-topologie mentionné dans le chapitre 2 "*Données spatiales, concepts & modélisation*").
- Portée de la topologie : La topologie doit pouvoir calculer les relations topologiques suivantes : relations d'adjacence, de composition et d'intersection entre des objets 3D.
- Stockage : Pour alléger le stockage de données, nous avons choisi que la topologie soit interprétée à partir de la géométrie. Le stockage de la topologie ne porte que sur la topologie simple. Alors que la topologie complexe servant de calculer des relations de connectivité entre des objets 3D ne sera pas stockée par ce qu'elle sera calculée sur la demande de l'utilisateur. Cela pourra optimiser le stockage de géométrie-topologie dans le SGBD.
- Algorithmes topologiques : Les calculs topologiques sont exprimés à l'aide d'un certain nombre d'algorithmes. Les fonctions de calcul de relations topologiques 2D proposées par PostGIS seront complétées par quelques algorithmes pour alimenter le SIG par des fonctionnalités topologiques portant sur des objets spatiaux 3D tels que les bâtiments.

### 1.2.3. MODELISATION DE LA TOPOLOGIE

Compte tenu du modèle géométrique ainsi que des choix topologiques mentionné ci-dessus, le modèle topologique est exprimé à partir du modèle géométrique. En effet, le modèle géométrique intégré au modèle topologique aura la représentation décrite dans le diagramme de classe UML ci-dessous (*Fig5.7*). Comme l'un des objectifs de l'utilisation de la topologie est d'assurer sa cohérence avec la géométrie, pour chaque primitive géométrique correspond une primitive topologique : Point-Nœud, Ligne-Arête, Surface-Face, Solide-Volume. La même primitive topologique peut correspondre à une ou plusieurs primitives géométriques et une même représentation topologique peut se rapporter à des objets ayant des représentations géométriques différentes. Pour qu'il soit cohérent avec le modèle géométrique, le modèle topologique est légèrement modifié par rapport aux modèles topologiques existant. En effet, comme la primitive *ligne* est éliminée du modèle géométrique, la primitive *arête* est éliminée à son tour du modèle topologique. Une relation est créée entre les primitives topologiques *face* et *nœud* en indiquant le numéro du *nœud* par rapport aux autres *nœuds* de la *face*.

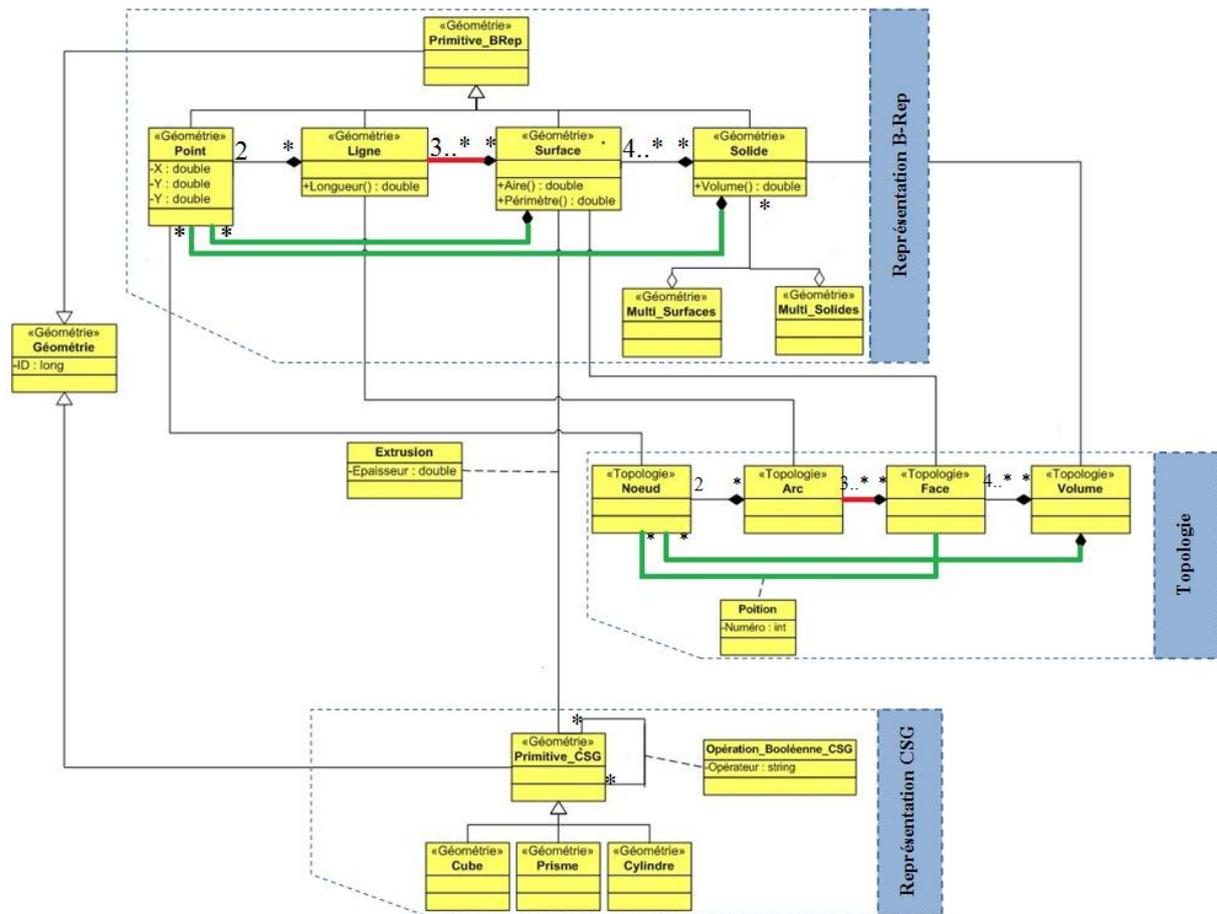


Fig5.7. Modèle conceptuel géométrique topologique

La figure (Fig5.7) traduit notre perception de la modélisation géométrique et topologique des objets spatiaux. Cependant le modèle conceptuel retenu est celui de la figure (Fig5.4) car la topologie sera interprétée de la géométrie (cela est expliqué en détail dans la partie 2.2 intitulée "Modélisation logique" de ce chapitre). La figure (Fig5.7) propose les relations possibles entre les primitives géométriques et topologiques dans un modèle conceptuel après la modification de la structure B-Rep (omission de la relation "surface-solide" et l'ajout des relations "surface-point" et "solide-point"). Certes la topologie structurelle est très importante pour assurer la cohérence du modèle géométrique, mais elle doit être complétée par d'autres fonctionnalités pour traiter des relations topologiques entre des objets 3D complexes. Dans la littérature, la topologie entre des objets 3D est largement étudiée en employant différentes techniques telles que la matrice des 9-intersections [ELLUL, *et al*, 2008], [CODEHAUS, 2009], l'étude des relations entre solides [ZLATANOVA, 2000], [KHUAN, *et al*, 2008.a], etc. Les principales relations topologiques pouvant exister entre deux objets 3D A et B sont : Disjoint (A, B), Rencontre (A, B), Touche (A, B), Contient (A, B) ou Contenu\_dans (B, A), Couvre (A, B) ou Couvert\_par (B, A), Chevauche (A, B) et Egal (A, B).

Comme PostGIS (jusqu'à sa version 1.5) ne permet pas de calculer des relations topologiques entre des objets 3D, il a été décidé de mettre en place une structure algorithmique capable de traiter ce type de relations. Nous avons utilisé les fonctions proposées par PostGIS et nous les avons complétées afin d'alimenter le SIG par des fonctionnalités topologiques 3D. Ces fonctionnalités sont décrites en détail dans le chapitre 7 «Interrogation des données spatiales».

### 1.3. PROBLEMATIQUE DE MODELISATION ATTRIBUTAIRE

Pour décrire des objets spatiaux, on peut associer des données attributaires aux données géométriques. Alors qu'une même modélisation géométrique et topologique peut être employée pour la représentation d'objets 3D issus de contextes assez diversifiés, la modélisation de données attributaires dans un SIG présente une problématique du fait qu'elle change selon le contexte à étudier. En effet, la construction de modèle géométrique et topologique d'un objet 3D donné peut être automatisée, alors que les données sémantiques correspondantes sont généralement rajoutées manuellement. Des structures de tables peuvent être définies *a priori* pour contenir des données géométriques et topologiques. Cependant, les structures des tables attributaires associées aux tables géométriques et topologiques ne peuvent pas être définies *a priori*. Comme la structure attributaire dépend de l'application, nous ne pouvons pas définir un modèle conceptuel attributaire capable de représenter tout type de données attributaires. En effet, les données thématiques associées aux données spatiales dépendent essentiellement du contexte dans lequel sont créés et utilisés les modèles spatiaux correspondants.

Pour ce faire, une structure générique doit être mise en place pour que le SIG permette de prendre en considération ce type de données. Lors de l'intégration de données dans la BDS, une ou plusieurs tables attributaires peuvent être intégrées pour décrire les objets spatiaux. Nous avons implémenté une structure générique décrite dans la troisième partie permettant d'intégrer des données attributaires quelque soit le contexte employé. Les données attributaires ne seront pas représentées par un modèle conceptuel car elles sont créées à la volée par les utilisateurs de notre prototype de SIG.

## 2. MODELISATION LOGIQUE

La modélisation logique, deuxième phase de conception du SI(G), consiste à décrire la structure de données utilisée sans faire référence au type de SGBD utilisé. En effet, si le passage du modèle conceptuel au modèle logique est une étape, le choix d'un type de modélisation logique en est une autre.

### 2.1. DIFFERENTS TYPES DE MODELISATION LOGIQUE

Il existe trois types de modélisation logique : relationnelle, objet et objet-relationnelle.

Bien que le modèle relationnel se base sur des principes relativement simples, il permet néanmoins de modéliser aussi bien des données simples que complexes. Il représente la réalité à travers des structures tabulaires, dites aussi matrices. Il offre trois types d'opérations : les opérations ensemblistes telles que l'union, la différence, le produit cartésien, etc., les opérations relationnelles telles que la projection, la restriction, etc. et les opérations dérivées telles que intersection, division.

Le modèle objet modélise les objets du monde réel selon deux concepts fondamentaux : données et comportements nécessaires pour décrire tels objets. Les classes décrivent les structures et les objets constituent des instances de telles classes.

La modélisation objet-relationnelle constitue un couplage des deux modèles : relationnels et objets dans un modèle dit "*relationnel étendu*". L'objectif majeur de ce type de modélisation est de profiter des avantages des deux modèles et d'éviter au mieux leurs désavantages. PostgreSQL/PostGIS et Oracle Spatial, par exemple, implémentent des structures spatiales objets-relationnelles.

### 2.2. COMPARAISON ET CHOIX DU MODELE LOGIQUE

Le choix d'un modèle logique pour la BDS se repose sur plusieurs critères qui portent essentiellement sur :

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

- Données spatiales : Il s'agit de tester la capacité du modèle pour la représentation de données spatiales. En effet, une entité spatiale est représentée à travers un ensemble de tables interliées, ou à travers des objets représentant des données et des traitements.
- Requêtes spatiales : Il s'agit d'indiquer les requêtes spatiales par rapport au langage d'interrogation.
- SGBD spatiaux : Bien qu'au niveau logique on doive éviter toute considération technologique relative au SGBD, nous considérons que le choix d'un SGBD particulier peut être critique pour le choix d'une modélisation logique. Il est très important de vérifier si le type de modèle logique est supporté ou non par le SGBD proposé.
- Avantages et inconvénients du modèle : Il faut connaître les avantages et les inconvénients d'une modélisation logique particulière pour juger la performance du modèle et éviter au mieux ses inconvénients.

Modèle logique	Relationnel	Objet	Objet-relationnel
<b>Critère de choix</b>			
Données spatiale	- Représentation des objets spatiaux à travers des tables reliées - Séparation entre données et traitements	Représentation des objets spatiaux à travers des tables décrivant leurs propriétés ainsi que les traitements associés.	Reprendre le modèle relationnel en ajoutant quelques notions qui comblent les plus grosses lacunes du modèle relationnel
Requêtes spatiales	(+) Requêtes s'appuyant sur le standard SQL, disponible sur la plupart des SGBD.	(-) OQL (Object Query Language) disponible sur quelques SGBD, structures complexes, l'hierarchie de généralisation / Spécialisation, les méthodes, etc.	(+) SQL3 intègre quelques notions objets.
SGBD spatiaux	Oracle, PostgreSQL, MySQL, SQLite, Ingres, Sybase, Access, Foxpro, Paradox, etc.	O2, GemStone, Orion, Gbase, Ontos, ObjectStore, Objectivity, Vbase, Iris	Oracle, PostgreSQL, Informix, etc.
Avantage et inconvénients	(+) Fondements mathématiques, plus intuitif, plusieurs utilisateurs sont familiarisés avec SQL, grande diffusion dans les SGBD, etc. (-) Non gestion d'objets complexes, Normalisation pouvant augmenter le nombre de jointures, etc.	(+) L'unité entre données et traitements, Structures complexes, héritage, appariement entre modèle et objets modélisés, etc. (-) Interrogation, les traitements sur les objets définis une fois pour toute, etc.	(+) Reprend tous les concepts du relationnel et adoption des concepts objets (-) Manipulation d'objets parfois compliquée

**Tab5.4** Comparaison entre différents types de modèle logiques

La plupart des SGBD relationnels mettaient en évidence l'insuffisance du modèle relationnel pour la modélisation de données complexes. L'apparition du modèle objet-relationnel a résolu plusieurs lacunes du modèle relationnel. Puisque PostgreSQL le permet, nous choisissons d'adopter un modèle relationnel avec quelques extensions objets telles que les classes, l'héritage, les types de données utilisateurs (tableaux, structures, listes, etc.), les fonctions, etc. Il s'agit d'une structure "objet-relationnelle", ou nous préférons plutôt l'appeler structure "relationnelle étendue".

## 2.3. REPRESENTATION DU MODELE LOGIQUE

Le modèle logique de données représenté dans la figure (Fig5.8) est composé d'un ensemble de tables objet-relationnelles (géométriques et attributaires). En effet, chaque table est représentée par un nom, la liste de ses attributs et un ou plusieurs attributs de jointure avec d'autres tables s'il y a lieu. La structure d'une table est représentée dans la figure (Fig5.8) avec une explication de chaque élément de la table.

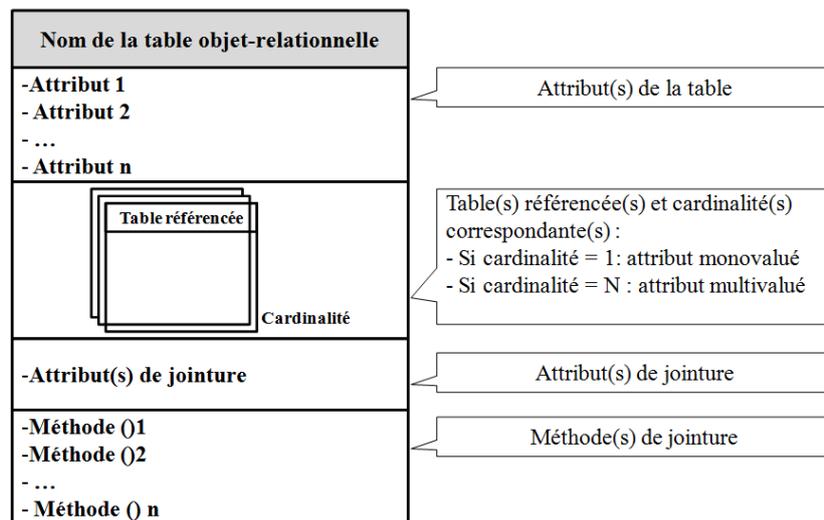


Fig5.8. Schéma illustratif de la représentation d'une table objet-relationnelle

L'objectif du modèle logique est de représenter au mieux le modèle conceptuel géométrique et topologique ainsi que le modèle attributaire. Les primitives géométriques *Point*, *ligne*, *Surface* et *Solide* ainsi que les agrégations *Multi\_surfaces* et *Multi\_solides* seront représentées par les tables *geo\_point*, *geo\_ligne*, *geo\_surface*, *geo\_solide*, *geo\_multi\_surface* et *geo\_multi\_solide*. Les relations entre les tables sont exprimées par des attributs sous forme de tableaux. Par exemple, dans le niveau conceptuel, une *surface* est définie par un ensemble de *points* correspondant à ses sommets. L'implémentation logique est matérialisée par une colonne dans la table *geo\_surface* contenant un tableau de *geo\_point*. La même implémentation logique est effectuée sur les autres relations.

Les primitives topologiques ne seront pas traduites par des relations logiques pour une raison double : les relations topologiques structurelles peuvent être interprétées à partir de la géométrie, d'une part, et d'autre part, nous jugeons que la topologie complexe est plus importante dans un SIG. Tout objet géographique peut être représenté par une géométrie particulière matérialisée par une relation de jointure entre la table correspondant à l'objet géographique et la table géométrique correspondante. Par exemple, en supposant gérer une table associée à un bâtiment, la table géométrique correspondante peut être de type *geo\_solide* ou *geo\_multi\_surfaces* et l'attribut de jointure pourrait être "*id\_bâtiment*" en tant qu'identifiant unique des bâtiments.

Outre la représentation géométrique, un objet géographique peut avoir une ou plusieurs tables attributaires associées. Cela peut qualifier la structure proposée par le modèle logique d'une structure générique car à la structure géométrique peut être associée une structure attributaire quelconque.

La figure (Fig5.9) reprend les différents composants du modèle conceptuel géométrique et les relations de jointure avec les tables attributaires correspondantes.

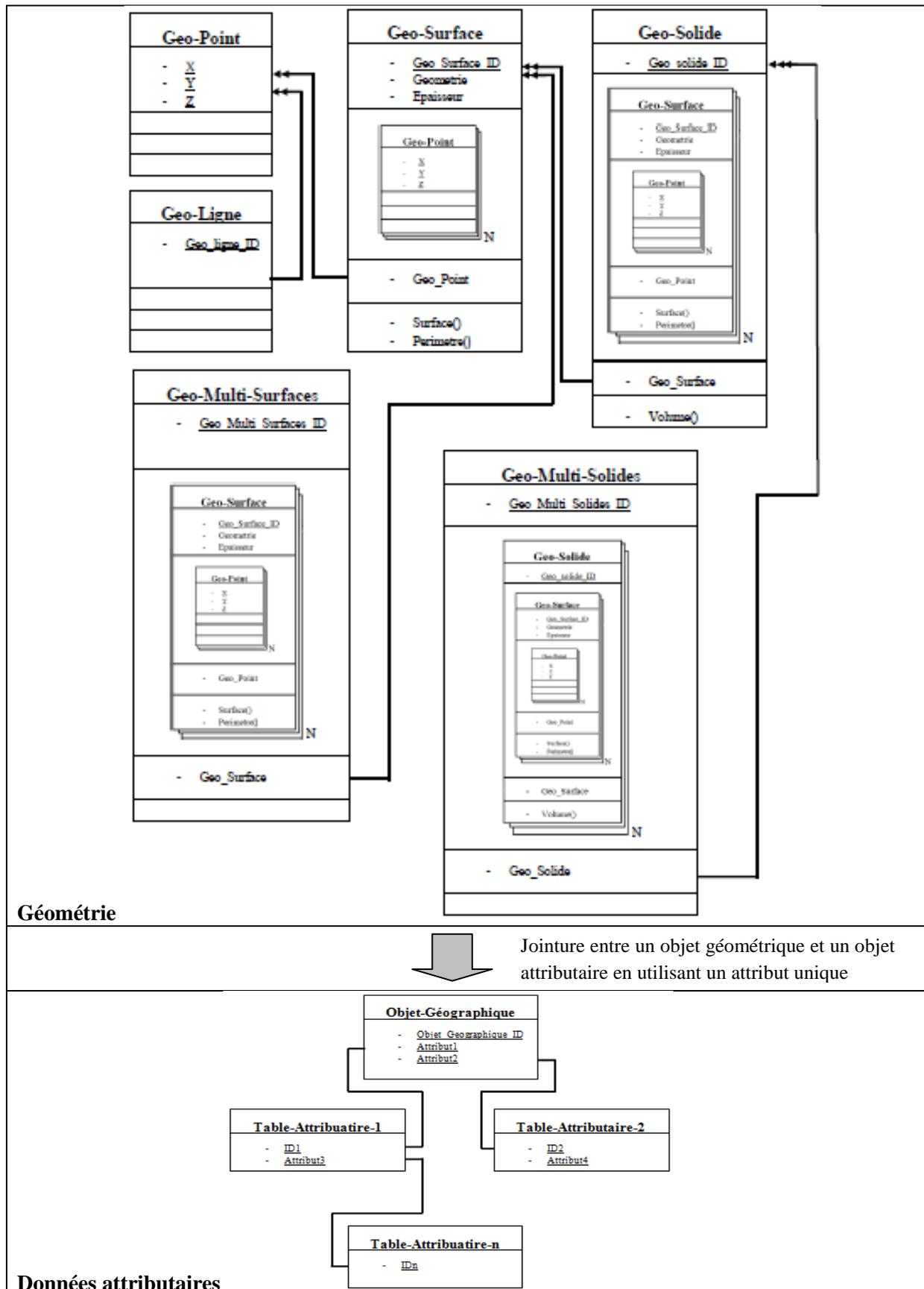


Fig5.9. Modèle logique de données

### 3. MODELISATION PHYSIQUE

Le stockage de données spatiales est une fonctionnalité primordiale pour tout SIG. La modélisation physique, troisième étape de conception du SI(G), consiste à implémenter le modèle logique dans un SGBD(S). Il s'agit d'une traduction du modèle "relationnel étendu" vu dans la partie précédente en un ensemble de tables physiques selon les spécifications d'un SGBD spatial particulier.

#### 3.1. CHOIX DU SGBD

Le nombre de SGBDS qui servira pour le stockage de la BDS est très limité. Parmi ce nombre restreint de SGBDS, le concepteur doit justifier son choix en se basant sur un ensemble de critères pouvant porter sur :

- Standards : Il s'agit de spécifications implémentées par le SGBDS, en particulier les spécifications ISO et OGC.
- Modèle de données : Il s'agit des modèles de données (relationnels, objets ou objet\_relationnels) supportés par le SGBD(S).
- Utilisation : Il s'agit de vérifier l'importance du SGBDS en termes d'utilisation aussi bien par les communautés d'utilisateurs que par les outils spatiaux commerciaux.
- Dimension de données spatiales : Selon les besoins définis à l'étape conceptuelle en termes de dimensions de données spatiales (2D ou 3D voire 4D), il est important de vérifier la capacité du SGBDS en termes de stockage des objets spatiaux, notamment de point de vue géométrie et attributs.
- Charge : Il s'agit d'étudier les limites du SGBD(S) de point de vue volume de données et le cas échéant si cela pourra dégrader les performances d'interrogation.
- Coût : le coût est parfois critique pour le choix du SGBD(S). Selon les ressources mises en œuvre pour la réalisation du SIG, on peut déterminer la nature du SGBD : gratuit ou commercial.

Nous avons comparé trois SGBDS (PostgreSQL/PostGIS, Oracle spatial et MySQL) selon les critères de choix vus ci-dessus pour en choisir le SGBD le plus approprié pour notre SIG. Les résultats de comparaison sont décrits dans le tableau (*Tab5.5*) suivant.

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

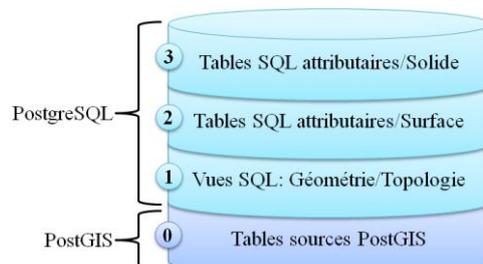
SGBDS	PostgreSQL/PostGIS	Oracle Spatial	MySQL
<b>Critères de choix</b>			
Standards	- Spécifications OGC SFS, - ISO SQL/MM part 3	- Spécifications OGC, - ISO 19107	- Il s'écarte assez sensiblement de la norme SQL, - Spécifications OGC
Modèles de données	Objet-relationnel	Objet-relationnel	Relationnel
Utilisation	- Support de PostgreSQL 8.3 par ESRI dans ArcGIS 9.3, - Utilisé dans des projets SIG open source tels qu'O3DG, - Très utilisé par des utilisateurs indépendants	- Partenariat avec FME (Feature Manipulation Engine), - Collaboration avec ArcGIS Server Enterprise, - Leader des bases de données dans les entreprises	- L'extension spatiale n'est pas assez utilisée. - Le SGBD est très utilisé dans les applications Web
Dimensions	- Coordonnées 3D, - Géométries 2D	- Cordonnées 3D, - Géométries 2D -Solides simples tels que les cubes, pyramides, etc.	- Cordonnées 3D, - Géométries 2D
Charges	- Taille maximum pour une base de données illimitée, - Problème de rapidité pour des grands volumes de données	- Très performant, - Oracle Spatial offre plusieurs fonctionnalités spatiales. - Oracle Locator offre moins de fonctionnalités car c'est gratuit	- Relativement rapides car il n'utilise pas les contraintes d'intégrité référentielle - Relativement rapide par rapport à PostgreSQL
Coût	Gratuit	- Oracle spatial (payant), - Oracle Locator (gratuit)	Gratuit

**Tab5.5** Comparaison de quelques SGBD spatiaux

Sur la base de cette comparaison, Oracle Spatial peut être considéré comme le meilleur SGBD spatial pour plusieurs considérations : performance, fonctionnalités spatiales avancées, stockages 3D, etc. Cependant, nous avons orienté nos choix vers les SGBD libres PostgreSQL et MySQL avec leurs couches spatiales successives PostGIS et l'extension spatiale de MySQL. Notre choix s'est arrêté sur PostgreSQL/PostGIS, celui-ci étant reconnu au niveau mondial comme le premier SGBDS Open Source respectant la norme SQL et les spécifications de l'OGC. De plus son évolution semble être assurée par une communauté active de développeurs.

### 3.2. BASE DE DONNEES SPATIALE MULTI-NIVEAUX

Le modèle physique est matérialisé par l'implémentation du modèle logique dans une BDS. Vu que la BDS contient généralement des données de différents types (géométriques, topologiques et attributaires) et pour des raisons d'optimisation, nous avons défini une structure multi-niveaux pour telle BDS. La figure (Fig5.10) montre l'organisation de la BDS implémentée dans PostgreSQL/PostGIS.



**Fig5.10.** Base de données spatiale multi-niveaux

Pour assurer une certaine cohérence avec le modèle logique, la BDS est organisée selon quatre niveaux.

⇒ Le niveau 0 est composé des tables sources issues du processus d'intégration (expliqué dans le chapitre 5 "Intégration & Création des jeux de données spatiales") des fichiers SIG du type *shapefile* et des fichiers CAO de type *DXF* ou *DWG*, etc. Ce niveau est composé des tables "PostGIS" qui offre plusieurs fonctionnalités d'interrogation géométrique et topologiques des données spatiales.

⇒ Le niveau 1 est matérialisé par un ensemble de vues SQL (géométriques et topologiques) interprétées des tables sources du niveau 0 et respectant la structure définie dans le modèle logique. L'intérêt d'utiliser des vues SQL est expliqué dans la partie 3.4, ci-dessous intitulé "Optimisation".

⇒ Les niveaux 2 et 3 consistent en des tables attributaires relatives aux données géométriques du niveau 1. Le niveau 2 est composé des tables attributaires concernant les surfaces et pouvant contenir des données sur les couleurs, les textures, les matériaux, des descriptions sémantiques, etc., alors que le niveau 3 est composé des tables attributaires concernant des solides ou des groupes de surfaces tels que les bâtiments, les parties des bâtiments, des objets spatiaux 3D, etc. La distinction des différents types de données relatives aux données spatiales peut apporter une valeur ajoutée en termes de compréhension de la base de données et de facilité d'interrogation. De plus, la séparation des niveaux géométrique et attributaire peut accélérer la procédure de construction des modèles 3D pour la visualisation ou l'interrogation.

### 3.3. REPRESENTATION DU MODELE PHYSIQUE

La figure (Fig5.11) montre un schéma illustratif de la BDS.

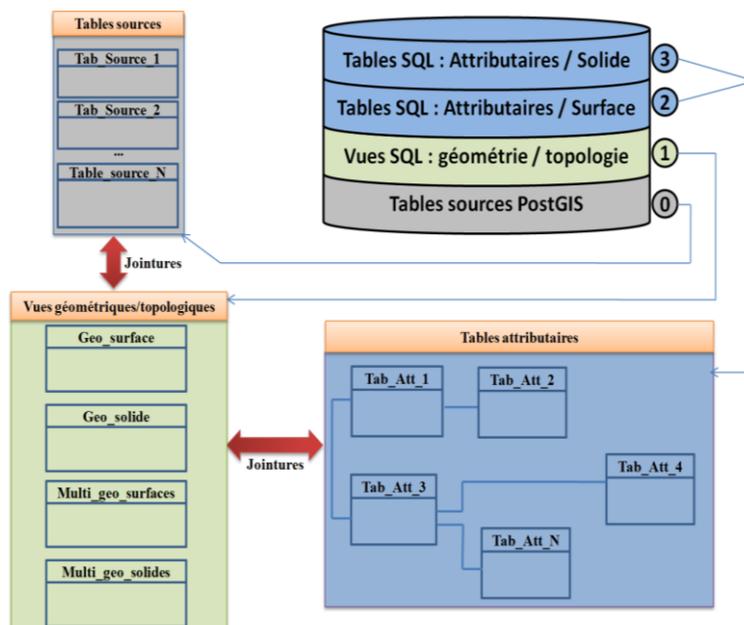


Fig5.11. Schéma illustratif de la BDS

Les tables géométriques/topologiques décrites dans le niveau logique sont traduites en des vues PostgreSQL matérialisées par des requêtes SQL sur les tables sources et ne contenant que des données géométriques ou d'apparence (couleur, textures, etc.). Les points sont les primitives géométriques les plus fréquentes dans une BDS. La création d'une table pour les points peut avoir des conséquences

négligables en termes de temps de réponse car les jointures entre les tables *geo\_point* et *geo\_surface* seront très nombreuses. Pour des raisons d'optimisation, la relation *geo\_point* est omise et remplacée par un tableau de points matérialisés par des coordonnées x, y et z dans la table *geo\_surface* ce qui évitera de mettre en place des jointures très coûteuses en termes de temps de réponse de requêtes SQL. Nous représentons également toutes les relations (1..n) et (n..n) par des tableaux. Par exemple, les vues *geo\_solide* et *geo\_multi\_surfaces* contiennent des colonnes sous forme de tableaux d'identifiants faisant référence aux *geo\_surfaces* composant tels solides et collections de surfaces. Cela permet, d'une part, de limiter les jointures (toutes les jointures vers les *geo\_surfaces* sont stockées dans un seul tableau) donc moins de lignes et d'éviter d'utiliser des tables de jointures, et d'autre part, il permet de stocker des *geo\_surfaces* pouvant être partagées par les différents objets. Le même principe est appliqué sur les *geo\_solides* formant des *geo\_multi\_solides*. Et au niveau attributaire, toutes les tables attributaires sont normalement liées aux tables sources et liées entre elles. Une même requête spatiale peut alors porter sur des tables géométriques et/ou des tables attributaires.

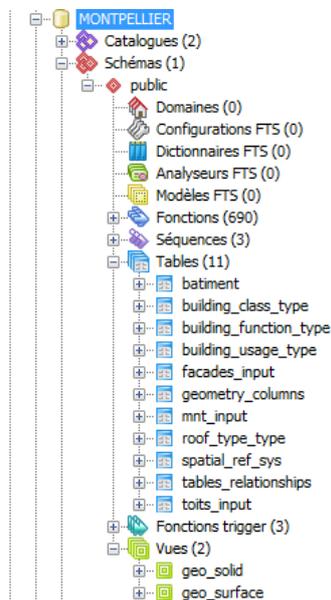


Fig5.12. Exemple d'implémentation d'une BDS dans PostgreSQL/PostGIS

La figure (Fig5.12) reprend un exemple d'implémentation d'une BDS dans le SGBDS PostgreSQL/PostGIS. Elle montre l'organisation des tables géométriques (représentées par des vues SQL), les tables PostGIS et tables attributaires.

### 3.4. OPTIMISATION

L'une des particularités des données spatiales est qu'elles sont généralement volumineuses. Pour ce faire, l'optimisation de la BDS devient très importante voire nécessaire.

Nous adoptons quelques techniques d'optimisation qui portent essentiellement sur :

- Identifiants : l'utilisation des OID (Object Identifier) pour identifier les différents objets dans la BDS permet, d'une part, d'éviter l'utilisation des séquences SQL et donc moins d'espace dans la BDS, et d'autre part, les OID se génèrent automatiquement et le chargement des tables est plus rapide.
- Indexation : Dans une BD, les indexes sont utilisés pour l'optimisation de certaines requêtes. En particulier, les SGBDS utilisent des indexes spatiaux tels que *GiST* (Generalized Search Tree), *B-Tree index*, etc. pour l'indexation des entités spatiales telles que les géométries.

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

Comme nous utilisons PostgreSQL/PostGIS, nous adoptons l'index spatial GiST permettant d'accélérer des requêtes spatiales sur des objets PotsGIS.

- Séparation de la géométrie et l'attributaire : L'architecture multi-niveaux de la BDS permet d'extraire les données géométriques/topologiques dans des tables séparées ce qui offre une rapidité d'affichage des modèles 3D.
- Vues SQL : Les vues SQL sont des tables virtuelles matérialisées par des requêtes sur une ou plusieurs tables et vues interconnectées. Nous utilisons les vues SQL pour extraire des données géométriques/topologiques et donc pour éviter que l'utilisateur ne saisisse des requêtes complexes, pour sécuriser les tables sources car les vues se mettent à jour automatiquement lorsqu'on met à jour les tables sources, etc.
- Mémoire tampon : pour la visualisation des modèles 3D à partir de la BDS, nous chargeons toutes les données nécessaires dans une mémoire tampon dans l'application SIG. Nous observons ainsi une meilleure disponibilité de données ainsi qu'une rapidité d'interrogation et d'affichage, libérer les sources de la BDS qui peuvent être partagées simultanément par plusieurs utilisateurs, etc.

### CONCLUSION

Nous avons proposé à travers ce chapitre une modélisation spatiale simplifiée des objets spatiaux qui seront stockés et traités dans le SIG 3D. Le travail est inspiré des travaux existants en termes de modélisation géométrique et topologique des données spatiales.

Au niveau conceptuel, nous avons appliqué une légère modification de l'approche de modélisation surfacique B-Rep en éliminant les primitives *lignes* du modèle car elles peuvent être interprétées à partir des sommets ordonnés de chaque primitive *surface*. Cette modélisation est capable d'approximer presque tout objet 3D du monde réel. Et pour plus de détails sur les façades des bâtiments, nous avons couplé B-Rep avec CSG pour l'obtention des murs sous forme de solides en extrudant horizontalement les surfaces. Au niveau logique, une modélisation relationnelle-étendue a été adoptée pour optimiser le stockage des primitives géométriques/topologiques. En effet, comme les primitives géométriques, à l'exception des points, sont définies par d'autres primitives géométriques (comme par exemple une surface peut être définie par une liste de points ordonnées, un solide peut être défini par un certain nombre de surfaces, etc.), ce type de modélisation logique peut être très adapté pour l'optimisation de stockage des objets spatiaux. Et au niveau physique, nous avons implémenté le modèle logique dans le SGBD PostgreSQL avec sa couche spatiale PostGIS.

La modélisation de données spatiale consiste alors à modéliser toutes les données utiles qui seront stockées dans la BDS. Mais, pour être exploitées par le SIG, la modélisation de données doit être couplée avec une modélisation des traitements pour définir les fonctionnalités du SIG.

CHAPITRE.6 INTEGRATION & CREATION DES JEUX DE DONNEES SPATIALES

Résumé ..... 118

Introduction ..... 118

**1. Définitions** ..... 118

1.1. Formats de données ..... 119

1.2. Fichiers CAO..... 119

1.2.1. *DXF* (Drawing eXchange Format) ..... 119

1.2.2. *DWG* (DraWinG) ..... 119

1.2.3. *COLLADA* (Collaborative Design Activity) ..... 119

1.3. Fichiers SIG (*SHP*)..... 119

**2. Tests et choix des formats** ..... 120

2.1. *Dxf2postgis* ..... 120

2.2. *Shp2pgsql* ..... 121

2.3. *Collada2pgsql*..... 122

**3. Processus d'intégration** ..... 125

3.2. Intégration des données géométriques/Topologiques..... 126

3.3. Intégration des données attributaires ..... 127

Conclusion..... 128

*«Entre ce que je pense, ce que je veux dire, ce que je crois dire, ce que je dis, ce que vous avez envie d'entendre, ce que vous croyez entendre, ce que vous entendez, ce que vous avez envie de comprendre, ce que vous croyez comprendre, ce que vous comprenez, il y a dix possibilités qu'on ait des difficultés à communiquer, mais essayons quand même»*

(EDMOND WELLS, Encyclopédie du savoir relatif et absolu, 15/11/2000)

### RESUME

*Les sources de données pour un SIG peuvent être multiples et hétérogènes. Pour être exploitées dans le SIG, les données peuvent être interprétées directement par le SIG ou transformées vers un autre format. Ce travail traduit une étude des principaux formats de fichiers de données 3D en termes d'avantages et d'inconvénients et explique le processus d'intégration de tels fichiers dans le SIG.*

*Il est à noter que l'acquisition des données initiales fait partie d'une autre étape non abordée dans cette thèse. Cette acquisition comporte ses propres difficultés et spécificités bien maîtrisées par d'autres branches des sciences de la mesure. Il fallait néanmoins se fixer sur certains formats de données pour pouvoir alimenter le prototype de SIG mis en place.*

### INTRODUCTION

Les données spatiales (et non spatiales) constituent le cœur de tout SIG. Elles peuvent indiquer des informations sur les formes des objets du monde réel (géométrie), les relations spatiales entre tels objets (topologie) et les données sémantiques décrivant tels objets. Le couplage de ces trois types de données dans une même Base de Données Spatiale (BDS) est avantageux en termes d'interrogation des objets du monde réel. Malgré que durant les dernières années, on a vu apparaître de nombreux modèles urbains en 3D avec des niveaux de détails plus ou moins réalistes, les SIG endurent encore de la rareté des données 3D (des solides représentant des objets 3D) ainsi que les topologies correspondantes. Les SIG 2D existants peuvent fournir des données attributaires importantes concernant des objets du monde réel. Cependant, tel type de données (2D) peut être insuffisant pour des modèles 3D nécessitant plus d'informations sur les éléments de détails des objets spatiaux, en particulier les bâtiments.

Les données spatiales peuvent provenir de différentes sources de données telles que des fichiers CAO, des fichiers SIG, etc. qui doivent être intégrées afin d'être exploitées dans un SIG. C'est dans ce cadre que s'inscrit ce chapitre. Dans la première partie, il propose quelques définitions. Dans la deuxième partie, il recense les différents formats de fichiers utilisés dans notre SIG 3D. Et dans la troisième partie, il décrit le processus d'intégration dans la BDS.

### 1. DEFINITIONS

Le concept d'intégration signifie *«action d'incorporer un ou plusieurs éléments étrangers à un ensemble constitué, d'assembler des éléments divers afin d'en constituer un tout organique; passage d'un état diffus à un état constant; résultat de l'action»* [CNRTL, 2010]. En se référant à cette définition, l'intégration de données spatiales ainsi que les données attributaires associées consiste à rassembler, généralement dans une BDS, des données provenant de différentes sources de données, pouvant être homogènes ou hétérogènes. Une multitude de formats de fichiers issus de la communauté des logiciels CAO (Conception Assistée par Ordinateur) ou des logiciels SIG permettent de représenter la géométrie ainsi que les propriétés d'apparence des objets 2D ou 3D (couleurs, textures, matériaux, etc.) correspondantes. L'informatisation de la création et de l'export des modèles virtuels a beaucoup facilité leur échange et leur partage entre les communautés d'utilisateurs.

### 1.1. FORMATS DE DONNEES

Nous avons étudié plusieurs formats de fichiers pour en choisir les formats pouvant être intégrés dans la BDS PostgreSQL/PostGIS. Comme les modèles 3D peuvent être créés par des outils SIG ou CAO, nos choix portaient sur des fichiers provenant de ces deux types d'outils.

Dans les domaines de la topographie et du SIG, les principaux logiciels utilisés sont AutoCAD (AutoDesk) et ArcGIS (ESRI). Lorsque des utilisateurs se servent d'AutoCAD, les fichiers sont le plus souvent enregistrés au format *DWG* ou *DXF* tandis que ceux utilisant ArcGIS exportent les données en fichier *Shapefile* (*SHP*). C'est pourquoi notre étude va se porter principalement sur ces trois différents formats (*DWG*, *DXF*, *SHP*), nous allons donc dans la partie suivante effectuer une description plus précise de ces extensions. En plus de ces trois formats standards, nous avons également étudié le format COLLADA qui peut être considéré comme une source de données intégrable dans PostgreSQL/PostGIS. L'étude est menée sur les plateformes Windows et Linux.

### 1.2. FICHIERS CAO

Les dessins produits sont le plus souvent réalisés en mode vectoriel. En d'autres termes, les logiciels de CAO attribuent des coordonnées (X, Y pour les plans 2D et X, Y, Z pour les modèles 3D).

Il existe de nombreux logiciels de CAO impliquant un nombre important de formats d'enregistrement des données.

#### 1.2.1. DXF (DRAWING EXCHANGE FORMAT)

*DXF*, sigle de "*Drawing eXchange Format*", est un format créé par la société "*Autodesk*" servant à échanger des fichiers entre systèmes CAO n'utilisant pas le même format de fichier natif. *DXF* est un format de fichier utilisé pour le transfert de données du type vecteur. Il s'agit d'un format ASCII, facilement générable, modifiable et importable. Il fonctionne par sections et par couple de lignes (Label, Valeur). Il contient de l'information pour la visualisation des données graphiques et est supporté par presque tous les logiciels graphiques.

#### 1.2.2. DWG (DRAWING)

*DWG* est le format des fichiers de dessins AutoCAD. Ce format est devenu un standard dans l'industrie CAO. *DWG* est l'abréviation de "*DraWinG*" (littéralement dessin voire plan). Le *DWG* est le format binaire natif d'AutoCAD. Le *DXF* en est le format ASCII d'export.

#### 1.2.3. COLLADA (COLLABORATIVE DESIGN ACTIVITY)

Collaborative Design Activity (abrégé en COLLADA, signifiant activité de conception collaborative) a pour but d'établir un format de fichier d'échange pour les applications 3D interactives. COLLADA 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 des fichiers XML, habituellement identifiés par leur extension *.dae* (« digital asset exchange », traduit par « échange numérique d'acquisition »).

### 1.3. FICHIERS SIG (*SHP*)

Le *Shapefile*, ou "fichier de formes" est un format de fichier issu du monde des SIG. Initialement développé par ESRI pour ses logiciels commerciaux (ArcGIS), ce format est désormais devenu un

standard et il est largement intégrable et utilisable par un grand nombre de logiciels tels que MapServer, Grass, Udig, MapGuide OpenSource, QGIS, etc.

### 2. TESTS ET CHOIX DES FORMATS

Dans le monde de l'Open Source, il existe quelques outils pour l'intégration des fichiers vus ci-dessus dans une base de données PostgreSQL/PostGIS. L'outil "*shp2pgsql*" permet l'intégration d'un fichier *Shapefile*, "*dxfg2postgis*" est dédié pour l'intégration d'un fichier *DXF* et "*collada2pgsql*" permet l'intégration des fichiers COLLADA.

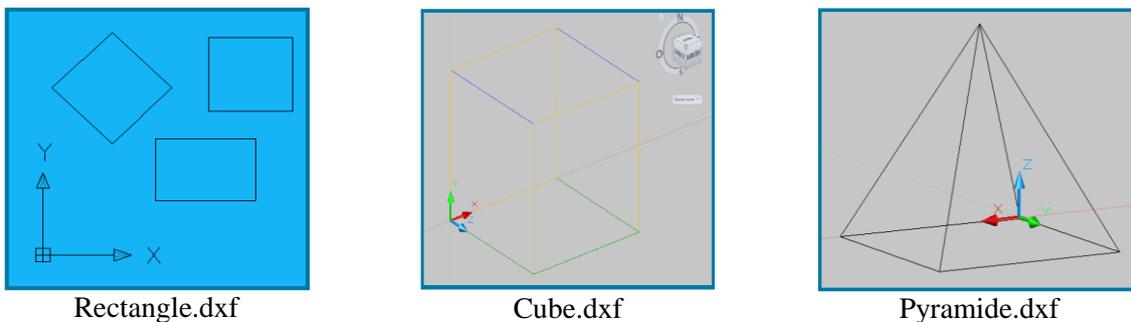
Vu la multitude des formats d'export des fichiers représentant des modèles virtuels, nous avons appliqué un certain nombre de tests sur ces types fichiers pour en choisir les formats répondant au mieux aux critères suivants :

- La capacité d'exporter des solides 3D : Il s'agit de vérifier si l'outil d'intégration est capable ou non d'intégrer des objets 3D en tant que des solides ou en tant que des polygones.
- La dimension des données exportées : Il s'agit de vérifier la capacité de l'outil d'intégration pour exporter des coordonnées 2D ou 3D.
- La simplicité de l'opération d'intégration : Il s'agit de vérifier la convivialité de l'outil en termes de simplicité d'utilisation.

Pour comparer ces outils d'intégration, nous avons créé des modèles 3D relativement simples. Ci-dessous, chaque opération d'intégration est décrite par un ensemble de modèles et un tableau décrivant le résultat d'intégration.

#### 2.1. DXF2POSTGIS

Pour tester "*dxfg2postgis*", nous avons créé trois fichiers *DXF* représentant respectivement un rectangle 2D, un cube 3D et une pyramide 3D tel que c'est décrit dans la figure (*Fig6.1*) suivante.



**Fig6.1.** Modèles 2D et 3D pour tester l'outil "*dxfg2postgis*"

Le tableau (Tab6.1) reprend les résultats de l'intégration des ces fichiers.

		Avant exportation	Après exportation
Rectangle 2D	Nom du fichier	Rectangle.dxf	
	Source	Acad Map 2009	
	Dimension	2D	2D
	Nature	1 polygone	1 polygone
	Système de projection		-1
	Commentaires : dxf enregistré au format dxf R12		
Cube 3D	Nom du fichier	Cube.dxf	
	Source	ACAD map 2009	
	Dimension	3D	2D
	Nature	6 polygones	6 polygones
	Système de projection		-1
	Commentaires : dxf enregistré au format dxf R12		
Pyramide 3D	Nom du fichier	Pyramide.dxf	
	Source	ACAD map 2009	
	Dimension	3D	2D
	Nature	5 polygones	5 polygones
	Système de projection		-1
	Commentaires : dxf enregistré au format dxf R12		

Tab6.1 Résultats d'exportation des fichiers DXF vers PostGIS

## 2.2. SHP2PGSQL

Pour tester "shp2pgsql", nous avons créé quatre fichiers *Shapefile* représentant respectivement des polygones 2D, une pyramide 3D, un cube3D et des polygones 3D tel que c'est décrit dans la figure (Fig6.2).

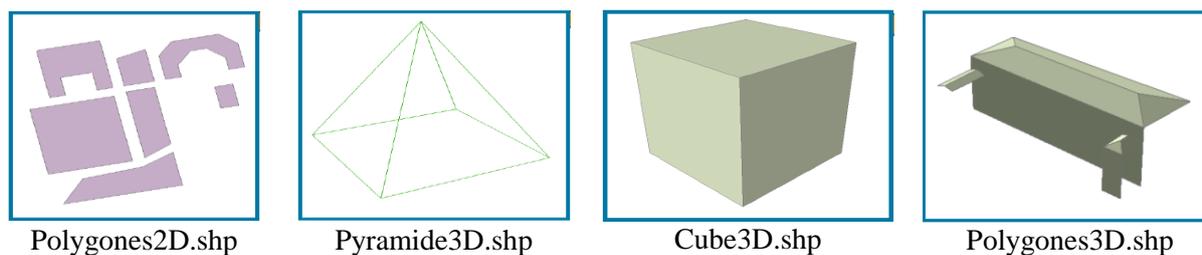


Fig6.2. Modèles 3D pour tester l'outil "shp2pgsql"

Le tableau (Tab 6.2) ci-dessous décrit les résultats de l'intégration des ces fichiers.

		Avant exportation	Après exportation
Polygones2D	Nom du fichier	Polygones2D.shp	
	Source	ArcMap	
	Dimension	2D	2D
	Nature	7 polygones	7 polygones
	Système de projection		-1
	Attributs	Oui	Oui
	Commentaires : - Options utilisées : -S, -c, -D.		
Pyramide3D.shp	Nom du fichier	Pyramide3D.shp	
	Source	ArcScene	
	Dimension	3D	4D
	Nature	5 polygones	5 polygones
	Système de projection		-1
	Attributs	Oui	Oui
	Commentaires : Options utilisées : -S, -c, -D. Les 4D sont : x, y, z + une constante		

Cube3D.shp	Nom du fichier	Cube3D.shp	
	Source	ArcScene	
	Dimension	3D	3D
	Nature	6 polygones	6 polygones
	Système de projection		-1
	Attributs	Oui	Oui
	Commentaires : Options utilisées : -S, -c, -D. Les 4D sont : x, y, z + une constante		
Polygones3D.shp	Nom du fichier	Polygones3D.shp	
	Source	ArcScene	
	Dimension	3D	4D
	Nature	12 polygones	12 polygones
	Système de projection		-1
	Attributs	Oui	Oui
	Commentaires : Options utilisées : -S, -c, -D. Les 4D sont : x, y, z + une constante		

Tab6.2 Résultats d'exportation des fichiers SHP vers PostGIS

### 2.3. COLLADA2PGSQL

Pour tester "collada2pgsql", nous avons créé quatre fichiers COLLADA représentant respectivement un cube 3D, une maison, deux maisons indépendantes et deux maisons mitoyennes telles que décrit dans la figure (Fig6.3).

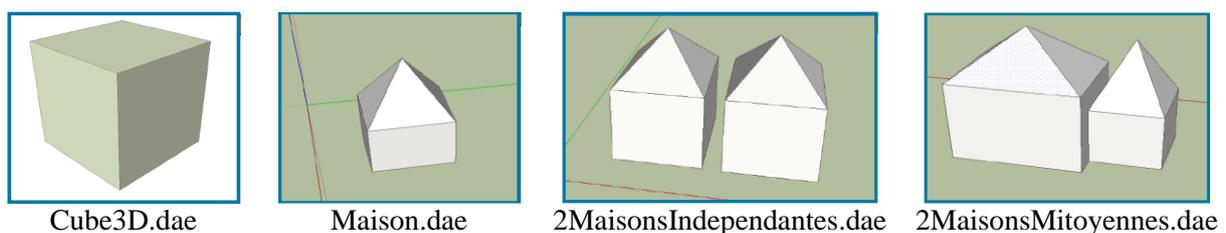


Fig6.3. Modèles 3D pour tester l'outil "collada2pgsql"

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

Le tableau (Tab6.3) ci-dessous montre les résultats de l'intégration des ces fichiers.

		Avant exportation	Après exportation
<b>Cube 3D</b>	Nom du fichier	Cube3D.dae	
	Source	Sketchup pro 7	
	Dimension	3D	3D
	Nature	6 polygones	2 TIN
	Système de projection		-1
	Attributs	Non	Non
	Commentaires :		
<ul style="list-style-type: none"> <li>- options utilisées : -d</li> <li>- les deux cubes ne sont pas mitoyens</li> <li>- le premier TIN est composé de 12 triangles = enveloppe interne du premier cube</li> <li>- le second TIN est composé de 12 triangles = enveloppe externe du premier cube</li> <li>- les enveloppes interne et externe permettent d'appliquer des textures sur les deux faces d'un polygone.</li> </ul>			
<b>Maison 3D</b>	Nom du fichier	Maison.dae	
	Source	Sketchup pro 7	
	Dimension	3D	3D
	Nature	10 polygones	2 TIN
	Système de projection		-1
	Attributs	Non	Non
	Commentaires :		
<ul style="list-style-type: none"> <li>- options utilisées : -d</li> <li>- existence d'un plancher séparateur entre la toiture et le cube</li> <li>- le premier TIN est composé de 12 triangles = enveloppe du cube</li> <li>- le second TIN est composé de 4 triangles = enveloppe de la toiture</li> <li>- il n'y a pas d'unification entre le cube et la toiture</li> </ul>			

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

2 maisons indépendantes	Nom du fichier	2Maisons.dae	
	Source	Sketchup pro 7	
	Dimension	3D	3D
	Nature	20 polygones	4 TIN
	Système de projection		-1
	Attributs	Non	Non
	Commentaires :		
<ul style="list-style-type: none"> <li>- options utilisées : -d</li> <li>- existence d'un plancher séparateur entre la toiture et le cube</li> <li>- le premier TIN est composé de 12 triangles = enveloppe du cube</li> <li>- le second TIN est composé de 4 triangles = enveloppe de la toiture</li> <li>- il n'y a pas d'unification entre le cube et la toiture</li> </ul>			
2 maisons mitoyennes	Nom du fichier	2MaisonsMitoyennes.dae	
	Source	Sketchup pro 7	
	Dimension	3D	3D
	Nature	20 polygones	4 TIN
	Système de projection		-1
	Attributs	Non	Non
	Commentaires :		
<ul style="list-style-type: none"> <li>- options utilisées : -d</li> <li>- les deux maisons sont mitoyennes</li> <li>- existence d'un plancher séparateur entre la toiture et le cube</li> <li>- le premier TIN est composé de 12 triangles = premier cube</li> <li>- le second TIN est composé de 4 triangles = toiture du premier cube</li> <li>- les coordonnées des 12 triangles forment l'enveloppe du cube</li> <li>- le troisième TIN est composé de 12 triangles = second cube</li> <li>- le quatrième TIN est composé de 4 triangles = toiture du second cube</li> <li>- les coordonnées des 6 triangles forment l'enveloppe de la toiture</li> <li>- il n'y a pas d'unification entre les deux maisons.</li> </ul>			

**Tab6.3** Résultats d'exportation des fichiers COLLADA vers PostGIS

A l'issue de ces tests, nous pouvons établir le tableau (*Tab6.4*) ci-dessous pour comparer les différents outils d'intégration en termes d'avantages et d'inconvénients.

Outil d'intégration	Avantages	Inconvénients
Shp2pgsql	<ul style="list-style-type: none"> <li>- Natif à PostGIS</li> <li>- Conserve les coordonnées 3D</li> <li>- Conserve les attributs</li> <li>- Différentes options d'importation</li> </ul>	<ul style="list-style-type: none"> <li>- Ne permet pas de faire ressortir l'appartenance d'un polygone à un solide (Topologie)</li> <li>- Interface sous forme de ligne de commande</li> <li>- Ne gère pas les textures</li> </ul>
Dxf2postgis	<ul style="list-style-type: none"> <li>- Interface graphique</li> <li>- Conserve les attributs</li> </ul>	<ul style="list-style-type: none"> <li>- Ne conserve pas la 3D</li> <li>- Obligation d'enregistrer les <i>DXF</i> en format <i>DXF R12</i></li> <li>- Ne gère pas les textures</li> </ul>
Collada2pgsql	<ul style="list-style-type: none"> <li>- Conserve les coordonnées 3D</li> <li>- Suivant la méthode de construction, il permet de faire ressortir l'appartenance d'un polygone à un solide</li> <li>- Gestion des textures</li> </ul>	<ul style="list-style-type: none"> <li>- Disposer de Linux</li> <li>- Interface sous forme de ligne de commande</li> <li>- Ne conserve pas les attributs</li> <li>- Décomposition des polygones en triangles</li> </ul>

**Tab6.4** Comparaison des outils d'intégration des fichiers *SHP*, *DXF* et *COLLADA*

"*Shp2pgsql*" et "*dx2postgis*" sont incapables, actuellement, d'exporter des solides. En effet, ce type d'outils exporte des objets 3D sous forme de polygones. "*dx2postgis*" ne permet pas, jusqu'à maintenant, d'exporter des coordonnées 3D. "*collada2pgsql*", créée par "CampToCamp", propose une intégration des objets 3D sous formes de solides matérialisés par des TIN. Cependant, l'outil "*collada2pgsql*" est actuellement incapable d'exporter les attributs relatifs aux objets 3D et décompose des objets complexes en des sous objets ce qui nous a poussé à délaisser cet outil.

Nous avons donc choisi de retenir le format de fichier *shapefile* comme source de données intégrables dans la base de données spatiale PostgreSQL/PostGIS. Ce choix est justifié par la conservation des coordonnées 3D et des attributs associés aux données spatiales, grâce à "*shp2pgsql*". De plus, *shapefile* est un format souvent utilisé pour l'échange de données géographiques, alors que le COLLADA est destiné plutôt pour l'échange de données 3D. Pour être intégrés dans le SIG, les formats *DXF* et *DWG* doivent être transformés en un format *shapefile* grâce aux outils ArcGIS ou FME.

### 3. PROCESSUS D'INTEGRATION

L'outil "*shp2pgsql*" permet d'intégrer des fichiers *shapefile* dans une base de données PostgreSQL/PostGIS. Mais pour être intégrées dans la BDS, les tables PostGIS, résultats de ce type d'intégration, doivent subir un autre processus d'intégration composé de trois étapes décrites dans la figure (Fig6.4).

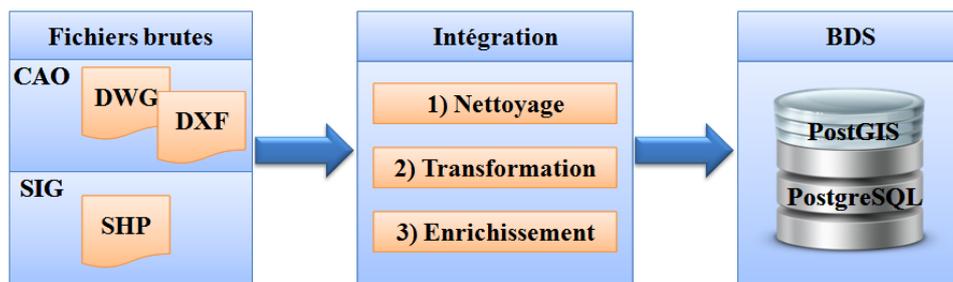


Fig6.4. Processus d'intégration de données spatiale dans le SIG

L'étape de nettoyage de données consiste à éliminer toute donnée non pertinente telle que le bruit, les géométries invalides, etc. Cette étape peut être effectuée manuellement ou automatiquement en utilisant la requête PostGIS :

```
DELETE FROM nom table WHERE isvalid(geometry) = false
```

L'étape de transformation consiste à :

- Transformer les fichiers *shapefile* en des tables PostGIS grâce à l'outil "*shp2pgsql*".
- Interprétation des données géométrique dans des vues SQL.
- Décomposer la BDS en quatre niveaux tels que décrit dans le chapitre précédent.

Cette étape de transformation se fait automatiquement sur demande de l'utilisateur.

L'étape d'enrichissement de données consiste à :

- Construction des solides à partir des polygones pour la gestion des objets 3D tels que les bâtiments.
- Ajouter des données attributaires et les faire correspondre aux objets 2D ou 3D du modèle.

Pendant cette étape, la construction des solides est semi-automatique. En effet, l'usager peut choisir un attribut qui identifie des objets 3D et lancer, à l'aide de l'application, la construction des solides correspondants. Mais, les données attributaires sont généralement créées manuellement.

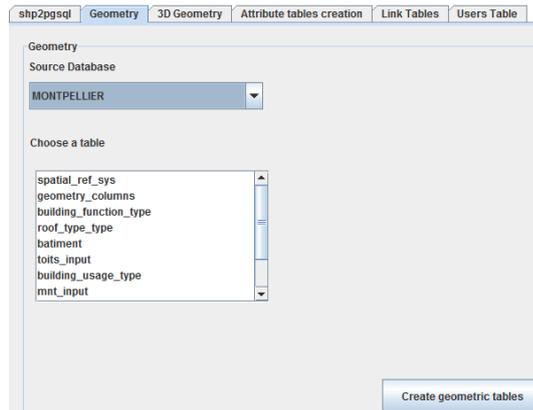


FIG6.5. Une capture écran de l'application permettant l'intégration de données

### 3.2. INTEGRATION DES DONNEES GEOMETRIQUES/TOPOLOGIQUES

L'intégration des données géométriques et topologiques se fait à l'aide d'un certain nombre d'algorithmes développés dans le cadre de cette thèse. Ces algorithmes sont matérialisés par des fonctions que nous avons ajoutées à celles proposées par PostGIS ou des fonctions que nous avons développées dans la cadre de l'application Java.

La géométrie des objets 3D est interprétée à partir des tables PostGIS et plus particulièrement de l'attribut "*the\_geom*" permettant de décrire, à travers un encodage hexadécimal, la géométrie d'un objet donné. PostGIS propose une fonction appelée "*AsEWKT*" permettant de retourner la géométrie d'un objet comme un type de géométrie (point, linestring, polygon, etc.) ainsi que les coordonnées des points le composant. Nous utilisons cette fonction pour décoder le champ "*the\_geom*" et interpréter les coordonnées des sommets des polygones. Tels sommets sont stockés dans un tableau dans chaque ligne correspondant à un polygone de la table géométrique.

Généralement, les fichiers *shapefile* ne contiennent pas des informations sur les solides. L'algorithme permettant de créer des solides à partir des polygones PostGIS est décrit dans la figure (Fig6.6). Cet algorithme est traduit par une fonction PostGIS dont le code correspondant est retracé dans l'annexe (page 225).

```

Paramètres
tables_number : numérique // Le nombre de tables contenant des polygones appartenant aux
mêmes solides
tables_names : tableau [tables number] de chaînes de caractère // Les noms des tables
géométriques
attribute_name: Chaîne de caractères // Les noms des tables géométriques
attribute_value : Chaîne de caractères // La valeur de l'attribut
Variables
Id-surface : numérique // L'identifiant d'une surface
result : chaîne de caractère // Le résultat de la requête
counter: numérique// Un compteur pour vérifier la chaîne de caractère
myquery : Chaîne de caractères // La requête SQL permettant de chercher les identifiants des
surfaces d'un solide
surfaces_cursor: curseur SQL// Le curseur SQL pour exécuter la requête myquery

Début
result :=''
counter :=1
id_surface :=0

//1) Constituer la requête
POUR i allant de 1 tables_number
FAIRE
    SI i=1 ALORS
        myquery := myquery || ' (SELECT oid' || ' FROM ' || tables_names[i] || ' WHERE ' ||
attribute name || ' = ' || attribute value || ' ) '

```

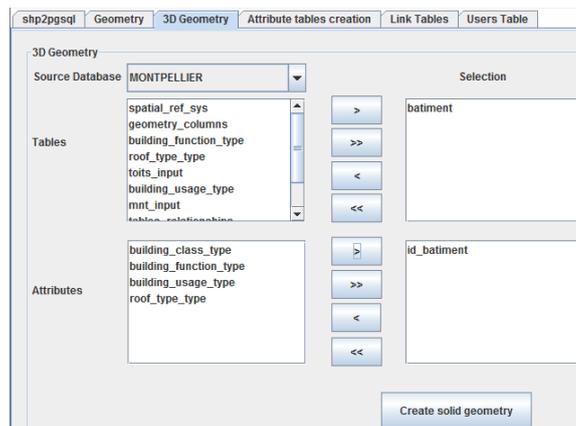
```

SINON
myquery := myquery || ' UNION (SELECT oid' || ' FROM ' || tables_names[i] || ' WHERE
' || attribute_name || ' = ' || attribute_value || ')'
FIN SI
FIN POUR
//2) Exécuter la requête
Exécuter la requête dans le curseur surfaces cursor
Ouvrir le curseur dans id_surface
TANT QUE le curseur retourne de résultat
Faire
IF counter = 1 ALORS
result = result || id surface
SINON
result = result || ',' || id_surface
FIN SI
counter = counter+1;
FIN TANT QUE
Fermer le curseur surfaces cursor
Retourner '{'||result||'}';// Tableau d'identifiants de polygones des chaque solide
Fin

```

**Fig6.6.** Algorithme de création des solides 3D à partir des polygones PostGIS

La figure (Fig6.7) montre un extrait de code de l'application permettant de générer des solides en se basant sur un ou plusieurs attributs indiquant l'appartenance des polygones aux solides.



**Fig6.7.** Une capture d'écran de l'application génératrice des solides

Un solide peut être composé de surfaces provenant des différentes tables (par exemple, un bâtiment peut être composé de surfaces provenant des tables "Façades" et "toits"). L'apport de l'application génératrice des solides est qu'elle peut interpréter automatiquement des solides à partir de plusieurs tables en se basant sur un ou plusieurs attributs communs choisis par l'utilisateur.

### 3.3. INTEGRATION DES DONNEES ATTRIBUTAIRES

Comme c'est indiqué dans le chapitre précédent, les données attributaires présentent une problématique depuis qu'elles n'ont pas *a priori* une structure prédéfinie. En effet, la structure des tables attributaires dans une BDS dépend du contexte du SIG. Ce type de données est généralement créé manuellement car il s'agit des données sémantiques qui ne peuvent pas être générées automatiquement. Par exemple, les données géométriques/topologiques pour un modèle urbain peuvent être générées automatiquement en utilisant différentes techniques telles que les algorithmes de reconstruction 3D à partir des nuages de points, reconstruction des plans 2D à partir des données rasters, etc. Alors que pour attribuer des données sémantiques à des bâtiments tels que le type (résidentiel, commercial, administratif, etc.), on doit saisir manuellement ce type de données dans la BDS pour chaque bâtiment ou en se basant sur des fichiers attributaires déjà existants. La structure multi-niveaux de la BDS permet d'interroger tels type de données à condition qu'elles soient jointes

## SECTION.II MODELISATION & INTEGRATION DES DONNEES SPATIALES DANS LE SIG

---

aux tables PostGIS générées par l'outil "*shp2pgsql*". Pendant la création des jeux de données spatiales, l'application fournit à l'utilisateur des informations sur la structure de la base de données (tables et attributs) ce qui facilitera l'affichage ainsi que l'interrogation des modèles 3D.

Les données attributaires présentent une importance considérable car elles complètent les données géométriques et topologiques en donnant des informations sémantiques sur les objets du monde réel ce qui pourra apporter plus de possibilités en termes d'interrogation de données spatiales. Cependant, nous sommes confrontés à des difficultés pour trouver des données spatiales contenant des données attributaires.

Afin de tester notre outil SIG, nous avons choisi de créer des échantillons de données sur des petits quartiers de quelques villes et sur des bâtiments isolés. Ce modèle comprendra des données géométriques, topologiques et sémantiques. En effet, nous avons choisi de réutiliser des données déjà existantes du laboratoire TRIO/PAGE. Nous avons ainsi obtenu différents types de données provenant de sources variées et collectés à des dates différentes.

Les données géométriques sont issues de la restitution photogrammétriques et de relevé LiDAR. Les données sémantiques pourraient provenir de la BD cadastrale, des données ajoutées manuellement, etc. N'ayant pas accès à toutes les informations nous proposons la constitution de données fictives telles que le nombre d'étages, le nombre d'occupants.

### CONCLUSION

L'analyse des données spatiales constitue la raison d'être des SIG. Elles servent de sources de données pour prendre des décisions concernant un phénomène spatial particulier et aussi pour la visualisation des résultats de l'analyse. Néanmoins, la majorité des fichiers CAO et SIG ne contiennent pas suffisamment d'informations sur les objets 3D, sur les relations spatiales entre tels objets ainsi que des informations attributaires décrivant tels objets. C'est dans ce cadre que ce chapitre a présenté nos travaux en termes d'intégration des différents types de fichiers dans le SIG.

La difficulté majeure que nous avons rencontrée réside dans le contenu des fichiers 3D. Nous avons appliqué quelques modifications sur de tels fichiers en combinant des fichiers provenant de différentes sources de données, corrigé automatiquement ou manuellement les erreurs géométriques et/ou topologiques, interprété des solides 3D à partir des polygones, ajouté des données attributaires, etc.

Les deux chapitres de cette section ont présenté la modélisation ainsi que l'intégration de données spatiales dans la BDS. Ces données vont servir pour la visualisation et l'affichage des résultats de l'analyse des données spatiales. Cela fera l'objectif des chapitres de la section suivante.

**SECTION.III MISE EN PLACE D'UN SIG 3D ACCESSIBLE VIA INTERNET**

**INTRODUCTION GENERALE DE LA SECTION III**

Cette section décrit l'implantation des différents services de notre prototype de SIG. En effet, elle reprend toutes les spécifications techniques et algorithmiques des fonctionnalités développées dans le cadre de cette thèse.

Le but initial consistait à mettre en place toutes ces fonctionnalités du SIG :

- En s'appuyant sur des outils entièrement libres que ce soit pour le stockage des données utiles pour le SIG ou pour les technologies utilisées dans le développement de ce SIG.
- En développant entièrement toutes les fonctionnalités du SIG dans le cadre de cette thèse sans faire recours à des outils SIG libres.

Nos travaux de développement du SIG porte principalement sur deux fonctionnalités : la représentation des données spatiales ainsi que leur interrogation.

Du point de vue représentation de données spatiales, l'objectif consiste à développer un moteur 3D pour :

- La représentation des modèles spatiaux en utilisant un système de couches spatiales.
- Le rendu des objets spatiaux, en particulier des bâtiments, représentés par des primitives géométriques vectorielles (points, lignes, surfaces et solides).
- L'interaction avec les modèles spatiaux rendus à l'écran telles que la rotation, la translation, etc. et leur manipulation telle que la sélection.
- Le rendu des résultats des requêtes utilisateurs d'interrogation d'un modèle spatial de manière facilitant la distinction des objets composant le résultat de cette requête par rapport aux autres objets du modèle.

Un ensemble d'algorithmes va être alors développé pour la mise en place du moteur 3D.

Du point de vue interrogation des données spatiales, nous distinguons deux principales fonctionnalités : L'interrogation des couches spatiales et l'interrogation des objets spatiaux.

L'interrogation des couches spatiales consiste à mettre en place un ensemble de fonctionnalités permettant de combiner deux ou plusieurs couches spatiales en utilisant des opérateurs booléens, en particulier l'union, l'intersection et la différence. L'idée de base consiste à permettre à l'utilisateur de combiner plusieurs couches élémentaires pour obtenir des couches plus sophistiquées.

Quant à l'interrogation des objets spatiaux, nous distinguons deux principaux types : l'interrogation topologique et l'interrogation attributaire. L'interrogation topologique offre à l'utilisateur la possibilité de calculer les relations spatiales entre des objets spatiaux, en particulier des surfaces et des solides. Alors que l'interrogation attributaire consiste à chercher des objets particuliers du modèle spatial répondant à des critères thématiques particuliers sachant qu'une requête attributaire peut également porter sur des propriétés géométriques et topologiques de ces objets.

Tous les services du SIG doivent être accessibles via Internet. Pour ce faire, une architecture client-serveur va être développée pour permettre aux utilisateurs d'accéder à distance aux différents services proposés par notre prototype de SIG.



**CHAPITRE.7 REPRESENTATION DES DONNEES SPATIALES DANS LE SIG 3D**

Résumé.....	132
Introduction .....	132
<b>1. L'organisation des jeux de données .....</b>	<b>132</b>
1.1. Création des couches spatiales .....	133
1.2. Visualisation des couches spatiales .....	136
1.3. Outils IHM (Interaction Homme Machine).....	138
1.3.1. Etape d'analyse .....	139
1.3.2. Etape de conception.....	139
1.4. Les niveaux de détails .....	142
<b>2. Moteur 3D .....</b>	<b>145</b>
<b>3. Comparaison avec les outils commerciaux.....</b>	<b>148</b>
Conclusion.....	149

*« L'imagination ou la visualisation et en particulier l'utilisation de diagrammes joue un rôle crucial dans la recherche scientifique ».*

(RENE DESCARTES, 1637)

### RESUME

*La puissance des SIG réside dans leurs capacités de représentation des données spatiales d'une manière facilitant leur interprétation ainsi que leur analyse. Les outils SIG utilisent généralement un système de couches dans lequel les jeux de données spatiales sont organisés à travers un ensemble de modèles spatiaux, 2D ou 3D, superposés. Chaque couche spatiale contient généralement des objets spatiaux partageant des caractéristiques communes. Ce chapitre explique comment les données spatiales sont organisées et rendues à travers notre outil SIG.*

### INTRODUCTION

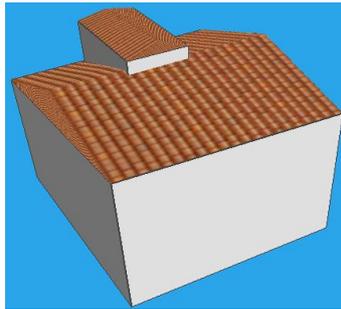
Dès le milieu des années 90, la 3D a commencé à apparaître dans les SIG, sauf qu'à l'époque, son implantation demandait des configurations informatiques relativement lourdes et coûteuses. Avec le développement des outils informatiques en terme de puissance de calcul, possibilités de visualisation 3D, notamment, le terme "SIG 3D" est devenu omniprésent aussi bien au niveau de la recherche scientifique qu'au niveau des outils commerciaux. Ainsi, la plupart des SIG commerciaux utilisent un système de couches pour structurer les données spatiales. Les données spatiales sont alors organisées à travers des sous-ensembles ayant généralement des caractéristiques ou propriétés variées. La superposition des couches spatiales crée un ensemble cohérent de données qui pourra être exploité logiquement. La distinction des différentes couches et objets spatiaux facilitera leur analyse ainsi que les prises de décision concernant un phénomène spatial particulier.

C'est dans ce contexte que s'inscrit ce chapitre. Nous présentons à travers ce chapitre nos travaux pour la représentation des données spatiales ainsi que leur visualisation. Ce chapitre est organisé comme suit : la première partie discute de l'organisation des jeux de données spatiales. Elle explique le concept de couche spatiale et son processus de création, les propriétés de visualisation 3D associée, les outils d'IHM utilisés pour manipuler les différents objets spatiaux ainsi que les différents niveaux de détail des modèles 3D. La deuxième partie positionne nos travaux par rapport aux outils commerciaux majeurs.

### 1. L'ORGANISATION DES JEUX DE DONNEES

Les jeux de données spatiales stockées dans la Base de Données Spatiales (BDS), en particulier les données géométriques et topologiques, sont délicats à interpréter voire illisibles dans leur format d'origine pour les utilisateurs. Par exemple, PostGIS représente les géométries par des chaînes hexadécimales (Fig7.1). Pour interpréter les formes on peut, par exemple, utiliser la fonction "asEWKT".

Le rendu 3D du bâtiment



Le stockage de la géométrie du bâtiment dans la BDS PostgreSQL/PostGIS (attribut *the\_geom*)

attribute_value text	the_geom geometry	solid_surfaces numeric[]
bâtiment = 116	01070000C0060000001060000C00100000001030000C0010000000500000021B072E86C142641022B871627E00141819543886CD74240FFFFFFFFFFFFFFFF21B072E86C142641022B871627E00141381	{42045,42046,42049,42383,

**Fig7.1.** Exemple d'une géométrie (un bâtiment) et sa représentation en PostGIS (*the\_geom*) en tant qu'une collection de surfaces

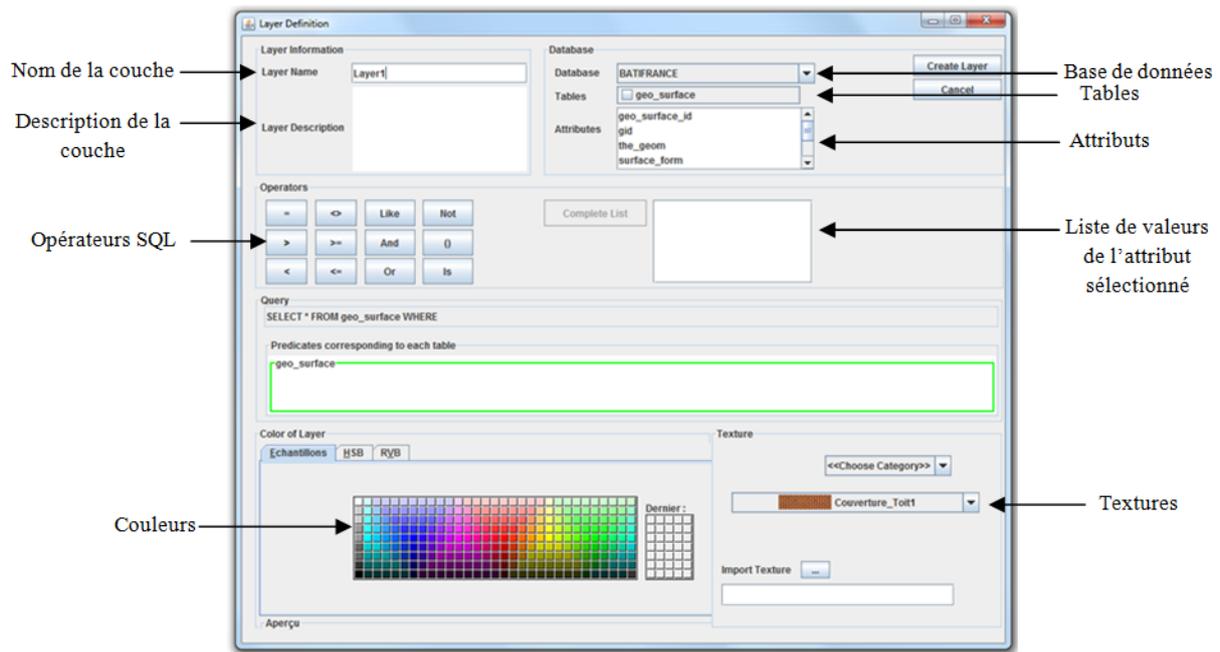
A partir de cet exemple, nous remarquons que la représentation "visuelle" des objets spatiaux couplée à des informations attributaires est plus facile à percevoir que des données purement alphanumériques issues de la base de données. De ce fait, leur traduction sous forme de modèles graphiques 3D facilitera leur compréhension ainsi que leur interprétation. Dans les SIG, les données géo-localisées sont généralement organisées à travers un système de couches spatiales (par exemple une couche pour la végétation, une couche pour les routes, une autre pour les bâtiments, etc.). Dans les concepts mis en place au cours de cette recherche, l'environnement SIG permet la création des couches spatiales sophistiquées en fonction du contenu de la BDS et répondant aux besoins et préférences des utilisateurs.

## 1.2. CREATION DES COUCHES SPATIALES

Parmi les caractéristiques fondamentales des SIG nous retrouvons leur capacité de traduction, de structuration et d'abstraction de la réalité spatiale sous la forme d'un ensemble de couches superposables. Nous nous inspirons des concepts des SIG commerciaux majeurs que nous adaptons pour la représentation de jeux de données spatiales 3D.

Le principe fondamental de notre SIG consiste à permettre à l'utilisateur de créer ses lots de données spatiales d'une façon dynamique. Les choix de l'utilisateur peuvent porter sur le contenu de la base de données (données géométriques et attributaires) ainsi que sur les apparences du modèle 3D (couleurs, textures).

La figure (*Fig7.2*) montre les différentes parties de l'interface graphique permettant de créer une couche spatiale à partir des données stockées dans la BDS.



**Fig7.2.** Interface graphique permettant de créer une couche spatiale

La création d'une couche spatiale se suit le processus ci-dessous :

- Choisir un nom et une description significative de la couche spatiale. Le nom et la description de la couche peuvent être utilisés plus tard pour chercher une couche spatiale stockée dans la base de données, pour partager la même couche entre des utilisateurs différents, etc.
- Ecrire la requête SQL permettant d'interroger la BDS pour en extraire le ou les modèles 3D correspondant aux choix de l'utilisateur.
- Choisir les couleurs et/ou les textures de chaque couche spatiale (propriétés de représentation).

La BDS constitue la source principale des données spatiales. De ce fait, chaque couche spatiale est le résultat d'une requête SQL sur la BDS. Comme indiqué dans la figure (Fig7.2), l'utilisateur choisit une BDS, les tables correspondant aux sources de données à afficher et/ou les attributs sur lesquels l'utilisateur désire appliquer des prédicats pour filtrer le résultat.

Comme les utilisateurs du SIG ne sont pas forcément familiarisés avec un langage de requête comme SQL, la création de ces mêmes requêtes est assistée. En effet, la création des requêtes est simplifiée en permettant à l'utilisateur de choisir directement dans l'interface graphique la ou les BDS, les tables géométriques et attributaires ainsi que la liste d'attributs (composant le prédicat) à utiliser dans la requête pour l'affichage d'un modèle 3D.

Pour expliquer la méthode de création des couches spatiales, nous utilisons une BDS exemple appelée "Bâtiments" dont la structure est expliquée dans la figure (Fig7.3). Pour créer la BDS, nous avons utilisé trois fichiers au format *shapefile* correspondant respectivement à un MNT, aux Façades et aux Toits de Bâtiments.

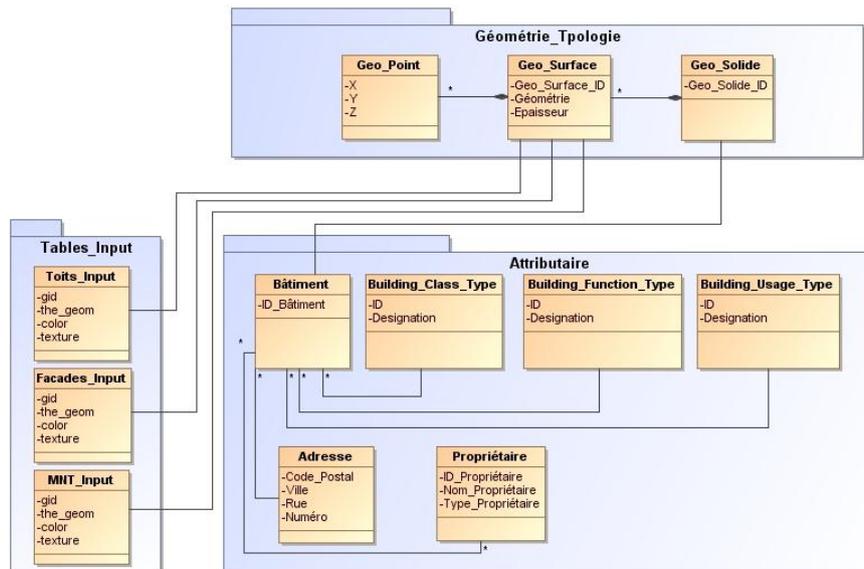


Fig7.3. Structure de la BDS "Bâtiments"

Comme les tables input (tables PostGIS) stockées dans la BDS correspondent à des modèles 3D différents (chaque table est le résultat d'intégration d'un fichier *shapefile*), l'utilisateur a la possibilité d'afficher l'ensemble ou uniquement un sous-ensemble des modèles existant dans la BDS. En effet, le SIG propose une case à cocher pour chaque table dont l'utilisateur veut afficher le modèle correspondant. Par exemple, la figure (Fig7.4) suivante montre la création d'une couche spatiale 3D appelée "Mes\_Bâtiments" incluant les modèles 3D correspondant aux jeux de données Toits et Façades dans une BDS contenant des données sur "Montpellier".

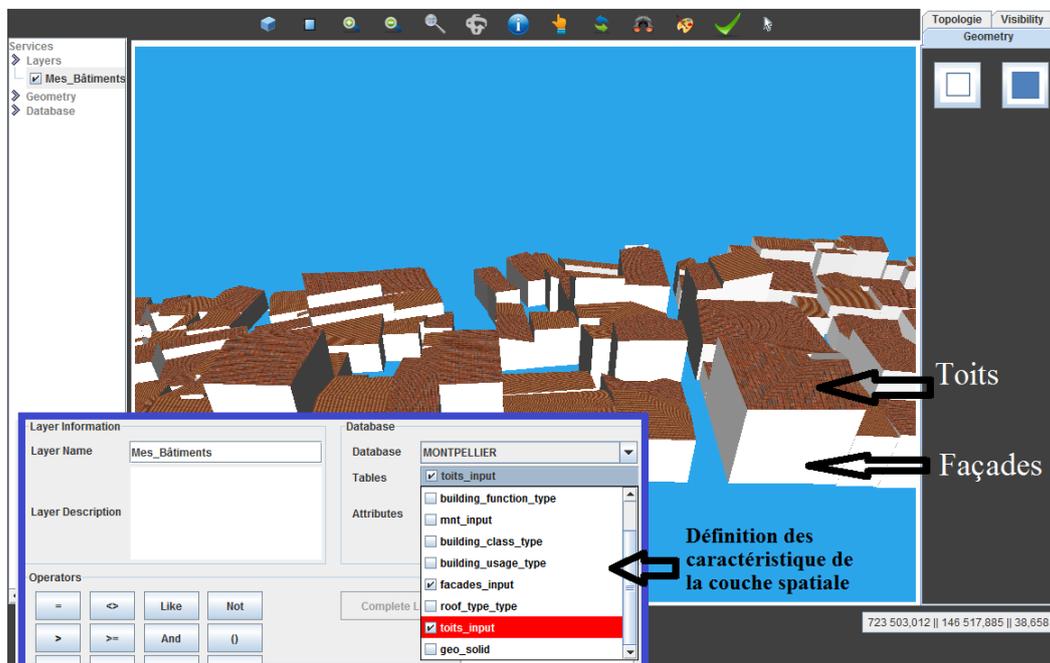


Fig7.4. Exemple d'une requête pour créer une couche spatiale représentant les Façades et les Toits des Bâtiments de la BDS "Bâtiments"

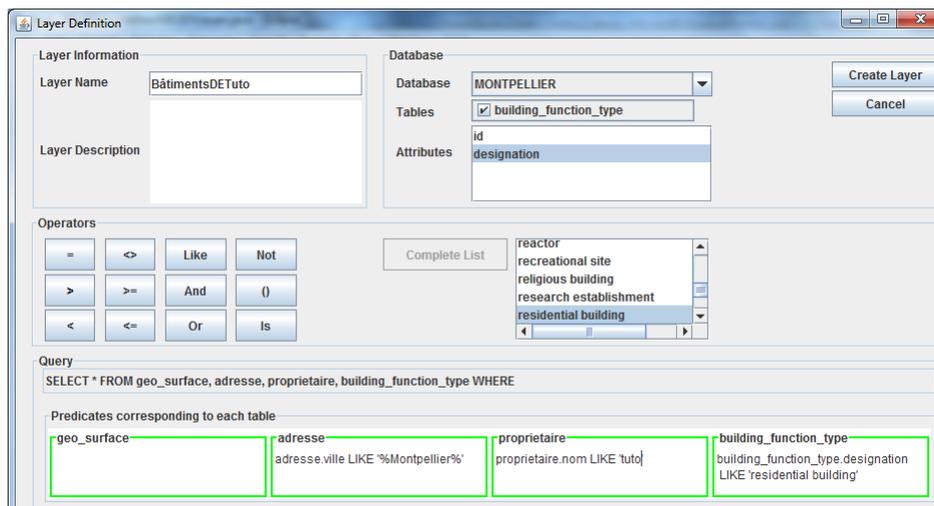
Une requête SQL peut contenir une ou plusieurs conditions dans la clause prédicat. Ces conditions servent de filtre pour le contenu de la table interrogée. Cependant, ces conditions peuvent nécessiter des jointures. Cela peut être une source d'erreur pour les utilisateurs (multitudes des tables à

interroger, utilisateur non familiarisés avec SQL, etc.). Pour éviter au maximum ces erreurs, nous avons mis en place une méthode pour assister l'utilisateur dans l'écriture de ses prédicats.

Le principe de la méthode consiste à créer automatiquement une zone texte pour chaque table, géométrique ou attributaire, sélectionnée par l'utilisateur, afin qu'il y inscrive, indépendamment, la ou les conditions correspondantes. A l'instar de ces conditions, plusieurs jointures entre les tables doivent être indiquées afin que la requête soit valide. Ces jointures sont interprétées automatiquement grâce à un algorithme que nous avons écrit et que nous appelons "*algorithme des jointures automatiques*". L'algorithme permet de détecter toutes les relations de jointures entre les tables de la BDS et y interpréter les jointures adéquates correspondant aux tables sélectionnées par l'utilisateur. Cet algorithme est expliqué en détail dans le chapitre 7 "*Interrogation des données spatiales*".

L'algorithme facilite la tâche de l'utilisateur lors de l'écriture de la requête SQL permettant la création d'une couche spatiale. L'utilisateur ne se contente que de choisir les tables sur lesquelles il veut faire un filtre ainsi que les conditions correspondant à chaque table. L'écriture des jointures entre les tables se fait automatiquement et d'une manière transparente à l'utilisateur.

Par exemple, pour afficher les bâtiments de type "*résidentiels*" situés dans la ville de "*Montpellier*" et appartenant au propriétaire "*tuto*", la requête correspondante est expliquée dans la figure (Fig7.5).



**Fig7.5.** Exemple d'une requête contenant un prédicat sur une table de la BDS

Avec cette méthode, le SIG assiste l'utilisateur lors de la création d'une ou de plusieurs couches spatiales, à travers une ou plusieurs requêtes, tout en se basant sur le contenu d'une BDS ainsi que sur les préférences de l'utilisateur. La validation d'une requête se fait d'une manière automatique et transparente à l'utilisateur. Avec cette technique, l'usager a la possibilité de n'afficher que les données dont il a besoin en filtrant les lots de données originaux.

Le contenu de la BDS est significatif pour la création des couches spatiales. En effet, plus la BDS contient des données attributaires et géométriques, plus la création des couches spatiales pourra être sophistiquée.

### 1.3. VISUALISATION DES COUCHES SPATIALES

Dès que l'utilisateur a fixé ses choix et préférences pour la création d'une couche spatiale, le SIG propose une interface graphique pour l'affichage du modèle virtuel 3D correspondant. La visualisation consiste à transformer les données de leur format (alphanumérique) de base de données vers un format

vectorel. Le modèle 3D obtenu doit donner une impression de relief. Le processus de visualisation est expliqué dans la figure (Fig7.6).

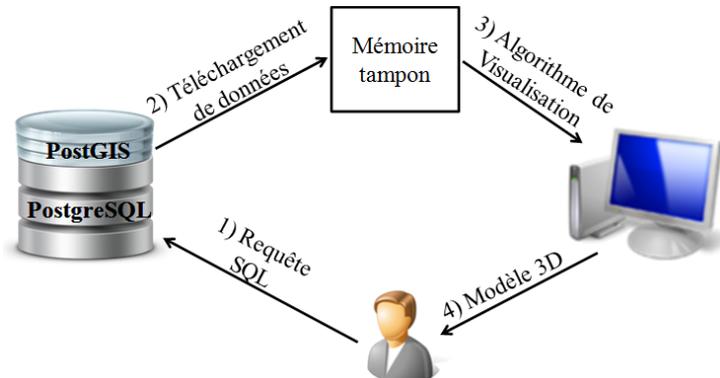


Fig7.6. Processus de visualisation d'un modèle 3D

La première étape consiste à télécharger les données requises par la visualisation de la BDS vers la mémoire tampon servant au stockage temporaire. Cette mémoire tampon est matérialisée par des matrices de coordonnées des sommets de chaque polygone du modèle 3D ainsi que toutes les données d'apparences associées (couleurs, textures, degré de transparence des formes géométriques, luminosité, etc.). Le contenu de la mémoire tampon est transparent pour l'utilisateur et réservé exclusivement au moteur 3D, dont nous verrons ci-dessous les concepts d'implémentation à des fins de visualisation tridimensionnelle.

Les motivations qui nous ont amené à privilégier une mémoire tampon pour la visualisation d'un modèle 3D plutôt qu'une interrogation directe de la BDS sont les suivantes :

- Disponibilité des données : Comme le SIG est accessible via un environnement Internet, la même BDS peut être accédée simultanément par plusieurs utilisateurs. En effet, la visualisation des modèles 3D directement à partir de la BDS peut avoir des conséquences négatives en termes de disponibilité de données. De ce fait, nous avons employé une mémoire tampon pour chaque modèle 3D. La BDS sera alors libérée après chaque téléchargement de données ce qui rend la BDS plus disponible pour un usage multiple.
- Rapidité d'accès aux données : Nous utilisons l'API (Application Programming Interface) OpenGL (Open Graphics Library) utilisé dans le cadre du moteur 3D pour la programmation des différentes fonctionnalités de visualisation et d'animation des modèles spatiaux. OpenGL utilise une technique de rafraîchissement continu des modèle 3D en utilisant la méthode "*display()*" appelée en boucle grâce à la classe "*Animator*". La méthode "*display()*" est dédiée au rendu de la scène. Et comme elle est appelée dans une boucle infinie, si l'interrogation de données se faisait directement de la BDS, cela alourdirait d'une façon rédhibitoire l'application. Une fois stockés dans les mémoires tampons, les modèles 3D sont affichés avec une rapidité et souplesse satisfaisantes.

Nous avons développé un algorithme pour le téléchargement de données spatiales vers la mémoire tampon. Le but de l'algorithme est le téléchargement des objets dans une matrice de type *Polygone* contenant toutes les informations associées aux polygones (coordonnées des sommets, textures, identifiants, etc.). Cet algorithme est détaillé dans la figure (Fig7.7).

```

Paramètres
RequêteSQL : chaîne de caractères // La requête SQL du modèle 3D
Variables
Matrice_temporaire : matrice alphanumérique // Matrice provisoire pour lire le
résultat de la requête
Matrice_modèle : matrice alphanumérique // matrice de polygones interprétée de la
Matrice_temporaire et servant de mémoire tampon
Début
    //1) Connexion à la BDS
    Connexion_BDS()
    //2) Exécution de la requête SQL correspondant à un modèle 3D
    Matrice_temporaire = Exécution_RequêteSQL(RequêteSQL)

    //3) Construire la mémoire tampon
    POUR i allant de 1 jusqu'à taille Matrice_temporaire
    FAIRE
        //3.1) définir un nouveau polygone
        Polygone = nouveau_polygone()
        //3.2) //Initialiser le polygone (coordonnée, identifiant, etc.) à
partir de la ligne à la position i de la Matrice_temporaire
        Initialiser_polygone(Polygone)

        SI élément à la position i possède une texture

        Télécharger_texture(Polygone)

        FIN SI

        //3.3) Ajouter le polygone à la Matrice_modèle
        Ajouter_Polygone(Polygone, Matrice_modèle )

    FIN POUR
Fin

```

*Fig7.7.* Algorithme de téléchargement de données spatiales vers la mémoire tampon

#### 1.4. OUTILS IHM (INTERACTION HOMME MACHINE)

Par comparaison, les outils commerciaux existants s'inscrivant dans le domaine spatial proposent des bibliothèques de "widgets" (objets graphiques) très riches permettant d'exécuter diverses fonctionnalités. La conception des outils IHM permet de définir tous les composants graphiques de l'application SIG ainsi que les interactions possibles des utilisateurs avec cette application.

Dans le SIG développé, nous avons suivi les étapes fondamentales de conception et d'évaluation de l'IHM. Il existe deux types d'approches de conception de l'IHM : une approche technocentriste centrée sur les capacités de la machine où l'utilisateur doit s'adapter aux fonctionnalités de la machine, et une approche anthropocentriste centrée sur les utilisateurs où la machine doit s'adapter aux besoins des utilisateurs.

Il faut distinguer la conception de l'application de la conception de l'interface graphique. Nous avons suivi une approche incrémentale dans laquelle la conception et la mise en œuvre de l'IHM évoluent au fur et à mesure de la réalisation des nouvelles fonctionnalités du SIG. Cependant, dès le départ, il existe au-moins toutes les fonctionnalités basiques de traitement de données spatiales.

Pour formaliser la mise en œuvre de l'IHM, nous sommes passés par les étapes suivantes représentées d'une manière simplifiée :

### 1.4.1. ETAPE D'ANALYSE

L'étape d'analyse est l'étape primordiale qui précède le stade de conception. Plusieurs définitions doivent y être mises en place.

- Définition des sites de référence
  - Systèmes existants (concurrence)

Nous avons étudié quelques outils existants. La plupart de ces outils tels que ArcGIS, MapInfo, Google Earth, Terra Explorer, etc. partagent des fonctionnalités basiques que nous classons en : déplacement (droite, gauche, haut, bas, avant et arrière pour la 3D), zoom (zoom avant et zoom arrière, fixer le zoom sur un objet particulier, etc.), rotation (sur deux axes pour la 2D et trois axes pour la 3D), orientation (nord, est, sud, ouest), sélection (sélection graphique ou textuelle), recherche d'un élément, afficher des informations sur un élément sélectionné, etc.

L'interaction de l'utilisateur avec ces outils se fait via les périphériques d'entrée-sortie, en particulier le clavier et surtout la souris pour manipuler les modèles spatiaux aisément et efficacement.

- Contexte d'utilisation

Avant de procéder à la conception de l'IHM, il est essentiel de définir le contexte d'utilisation de l'application en termes d'utilisateurs (utilisateurs experts, utilisateurs novices, utilisateurs occasionnels, nombre d'utilisateurs, etc.), utilisation collective ou individuelle, produit commercial ou gratuit, mesures de sécurité, produit open source ou non, objectifs de l'application, etc.

Nous avons considéré que les utilisateurs, experts ou novices, peuvent être nombreux dans le cas d'une application accessible en ligne. Les utilisateurs peuvent travailler individuellement ou ensemble en partageant des mêmes lots de données. Les objectifs fondamentaux de l'application sont : la mise en place d'un outil SIG pour l'intégration de données spatiale dans une BDS, l'interrogation de la BDS dans le but de structurer les données sous formes de couches répondant aux choix et préférences des utilisateurs et créées à partir de jeux de données prédéfinis, la manipulation des couches spatiales (ajout de sous-couches, suppression de couches, stockage de couches, etc.), langage graphique pour l'exécution d'opérations booléennes sur plusieurs couches, langage graphique pour l'exécution d'opérations topologiques sur les modèles 3D, langage textuel pour la recherche et l'affichage d'éléments particuliers dans le modèle en se basant sur des propriétés géométriques, topologiques et attributaires, etc.

### 1.4.2. ETAPE DE CONCEPTION

- Conception du système : les concepts informatiques

Avant d'expliquer la conception des éléments graphiques de l'application SIG, nous expliquons dans la figure (*Fig7.8*) l'architecture globale de l'application. Nous distinguons deux parties principales : la partie données (le serveur et les BDS) et la partie traitements (l'application SIG proprement dite permettant de traiter les données stockées dans le serveur). La partie traitements se compose à son tour de trois sous-parties : l'intégration des données (passage du format fichier au format BDS), la création et la superposition des couches spatiales 3D, l'interrogation et l'analyse des lots de données spatiales. Toutes les fonctionnalités de l'application sont accessibles via Internet car l'application est matérialisée par un applet Java pouvant s'exécuter dans un navigateur Web.

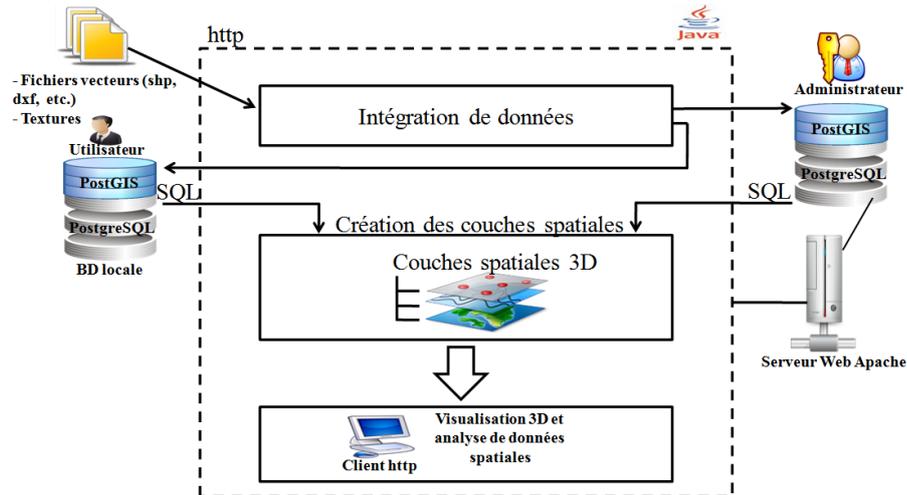


Fig7.8. Architecture globale de l'application

Pour formaliser la mise en place des différentes interfaces graphiques, nous avons développé un modèle conceptuel, matérialisé par un diagramme de classes UML, tel que décrit dans la figure (Fig7.9). Le but de l'application est de proposer une IHM simple à utiliser en adoptant les principales fonctionnalités des outils commerciaux reconnus et en rajoutant de nouvelles fonctionnalités innovantes qui font parties de l'objectif de cette thèse.

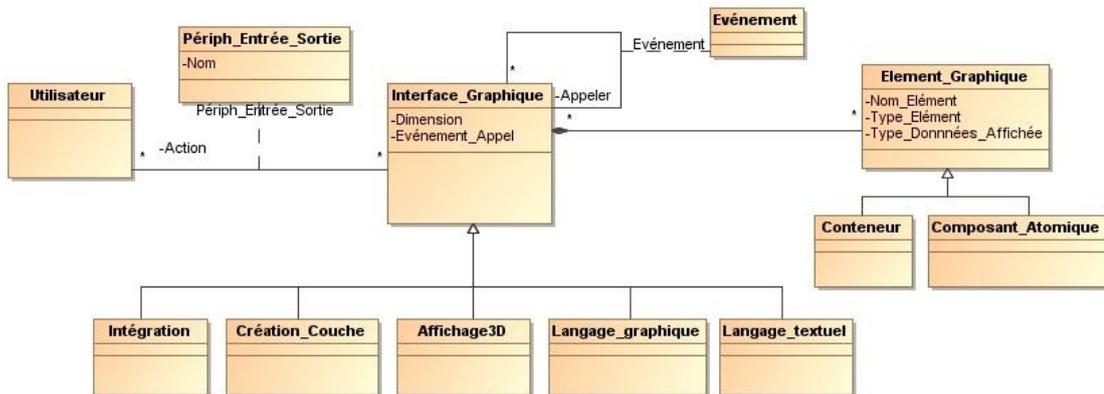


Fig7.9. Modèle conceptuel des interfaces graphiques de l'application

Le modèle conceptuel reprend brièvement les différentes interfaces graphiques mises en place dans ce SIG (interface d'intégration de données dans la BDS, interface de création de couches spatiales, interface d'affichage 3D, interface du langage graphique et interface du langage textuel pour l'interrogation du modèle 3D ou de la BDS). L'utilisateur se sert d'un périphérique d'entrée-sortie pour lancer une action sur une interface graphique sachant qu'il peut appeler une interface graphique à partir d'une autre. La conception détaillée de l'IHM doit se faire interface par interface en précisant pour chacune les sous-fenêtres, les composants graphiques atomiques, les événements possibles pour appeler l'interface, les rôles de l'interface, etc. La conception des interfaces se fait à l'aide d'un ensemble d'outils graphiques dédiés proposés par l'EDI (Environnement de Développement Intégré) Eclipse (version Eclipse-JEE-Galileo).

- Modèle de l'interaction

Le modèle d'interaction doit pouvoir décrire formellement la communication entre l'utilisateur et la machine. En revanche, les actions effectuées par l'utilisateur sur les objets graphiques doivent être

indiquées à l'avance. Pour illustrer les interactions homme-machine, nous proposons un exemple de diagramme de séquences pour la création d'une couche spatiale (Fig7.10).

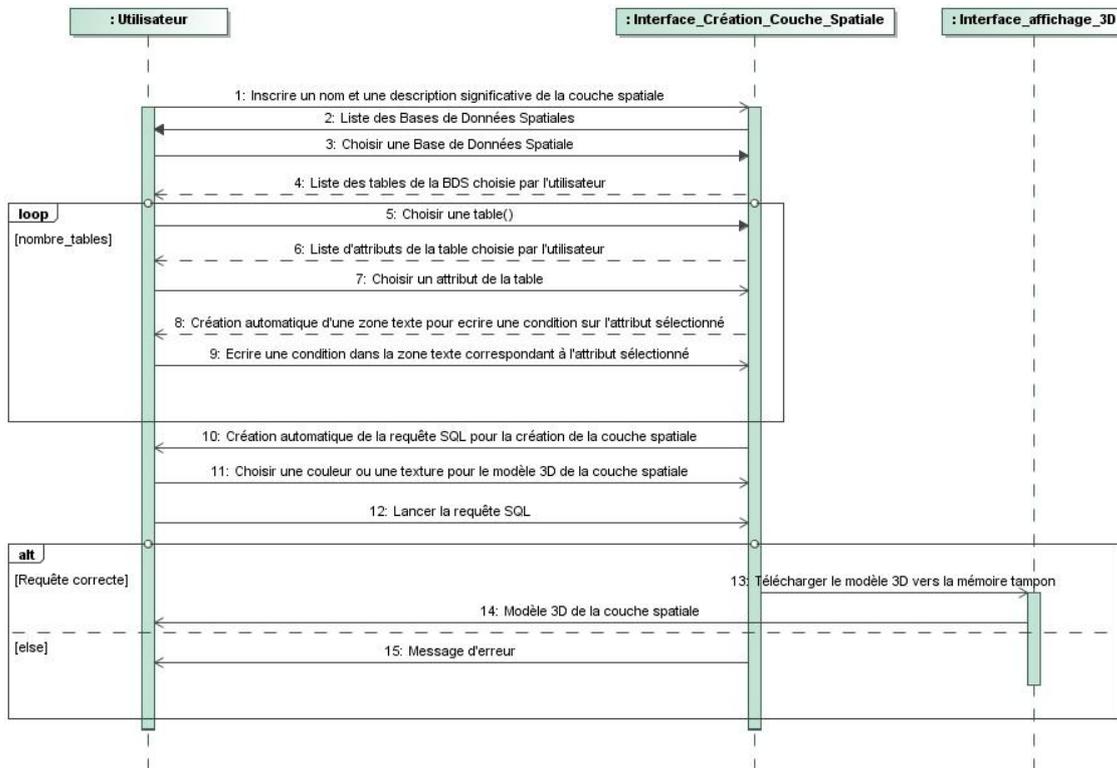


Fig7.10. Exemple d'un modèle d'interaction pour la création d'une couche spatiale

Nous avons utilisé des diagrammes de classes et de séquences pour illustrer quelques exemples de conception des interfaces graphiques ainsi que l'interaction des utilisateurs avec l'application. Pour des raisons ergonomiques, nous avons essayé d'utiliser des icônes contenant des images significatives pour les différentes actions sur les fonctionnalités de l'application. La figure (Fig7.11) montre quelques exemples d'icônes utilisés.

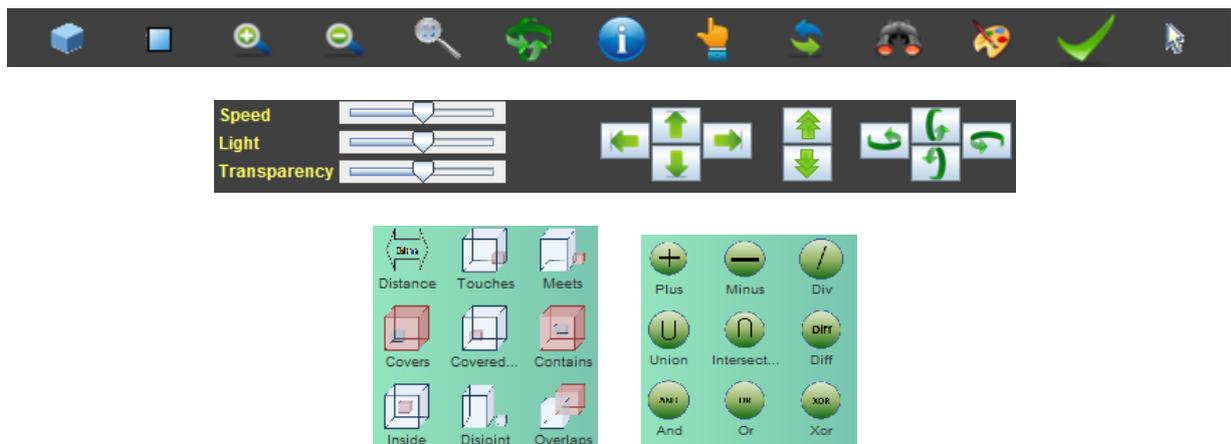
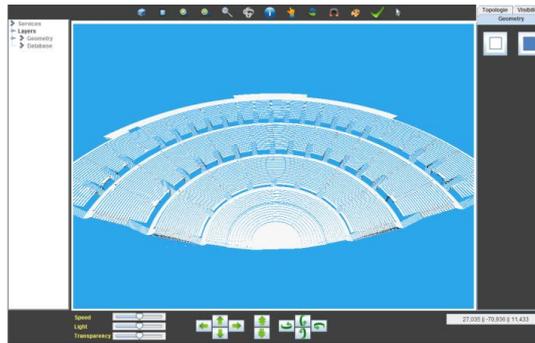


Fig7.11. Quelques éléments graphiques proposés par les interfaces du SIG

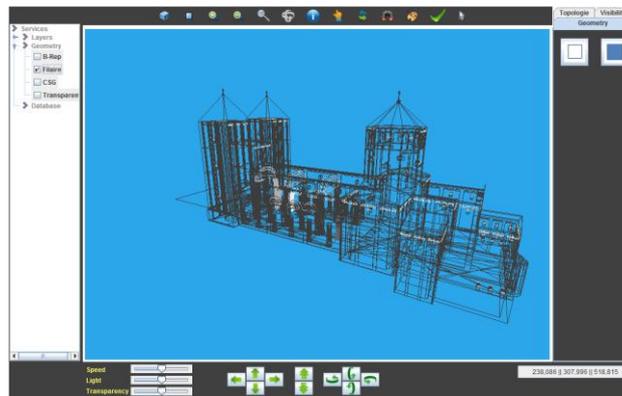
### 1.5. LES NIVEAUX DE DETAILS

Une fois le modèle 3D correspondant à une couche spatiale téléchargé et rendu à l'écran, le SIG propose plusieurs alternatives d'affichage. En effet, l'utilisateur peut afficher le modèle 3D en tant que modèle B-Rep, modèle filaire, modèle hybride B-Rep\_filaire ou modèle hybride B-Rep\_CSG. Le modèle B-Rep est matérialisé par un ensemble de polygones partageant des arêtes communes. La précision du modèle B-Rep dépend généralement du nombre de polygones. La figure (Fig7.12) montre un exemple d'un modèle B-Rep du théâtre monumental de la MANDEURE (53881 polygones).



**Fig7.12.** Exemple d'une visualisation B\_Rep d'un modèle 3D

Le modèle filaire est composé des mêmes composants que le modèle B-Rep sauf que les polygones du modèle B-Rep sont traduits dans le modèle filaire par des polygones fermés partageant des arêtes communes. La figure (Fig7.13) représente un modèle filaire de l'église canoniale de NIEDERMUNSTER (22658 arêtes).



**Fig7.13.** Exemple d'une visualisation filaire d'un modèle 3D de l'église canoniale de NIEDERMUNSTER

Dans le chapitre 4 "*Modélisation des données spatiales pour le SIG 3D*", nous avons discuté le couplage des deux méthodes B-Rep et CSG pour la représentation géométrique des objets spatiaux avec un niveau de détail élevé. Dans cette section, nous allons expliquer l'algorithme permettant de passer de B-Rep à CSG.

Parfois, le couplage des représentations B-Rep et filaire s'avère être très avantageux en termes de rendu 3D. En effet, le couplage des deux types de représentation permet d'afficher dans le même modèle toutes les surfaces ainsi que leurs contours pour chaque objet 3D.

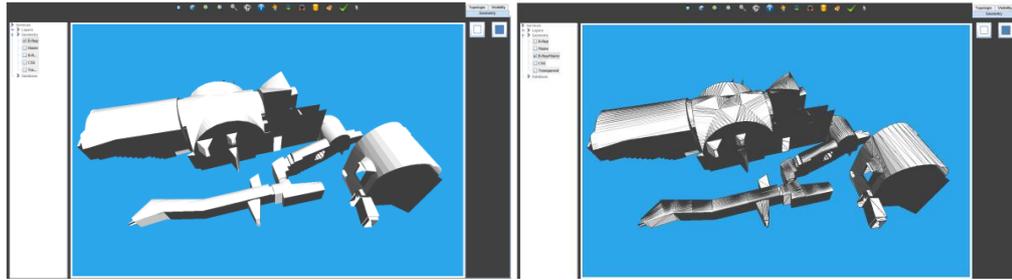


Fig7.14. Représentation B-Rep (à gauche) et B-Rep\_filaire (à droite) de Châtel-sur-Moselle (galeries de la forteresse)

• **Passage de B-Rep à CSG dans le cas d'un bâtiment**

Le passage d'une représentation à l'autre se base sur le principe de l'extrusion. Pour que l'extrusion soit toujours dans le même sens (vers l'intérieur des bâtiments), les surfaces d'un bâtiment doivent être orientées (les sommets sont ordonnés dans le sens trigonométrique ou le sens inverse). Les polygones PostGIS obtenus suite au processus d'intégration des fichiers *Shapefile* sont orientés dans le sens horaires et les trous dans le sens antihoraire [ESRI, 1998]. Cette orientation est importante pour réaliser l'extrusion. L'idée est très simple. Pour chaque surface, on calcule deux vecteurs, puis on calcule leur produit vectoriel que l'on normalise et dont on interprète la nouvelle forme extrudée. La figure (Fig7.15) explique le processus d'extrusion.

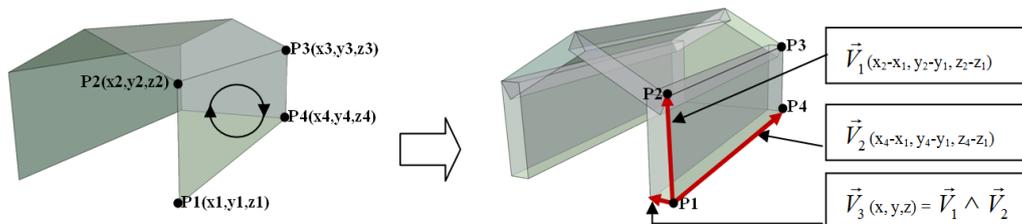


Fig7.15. Extrusion des polygones B-Rep pour obtenir des solides CSG

L'extrusion se fait perpendiculairement par rapport au plan de la surface à extruder. Cependant chaque surface est généralement connectée à d'autres surfaces. Le tableau (Tab7.1) ci-dessous montre les différentes relations de connexion entre deux surfaces et le résultat de leur extrusion.

Angle = 90°	Angle > 90°	Angle < 90°
<b>Extrusion</b>		
<p>Forme valide</p>	<p>Forme valide</p>	<p>Forme invalide</p>

Tab7.1 Les résultats d'extrusion de deux surfaces interconnectées selon l'angle d'intersection

Ce tableau indique que si l'angle d'intersection entre deux surfaces S1 et S2 est inférieur à  $90^\circ$ , la forme géométrique obtenue par l'extrusion des deux surfaces sera invalide. Il faut alors corriger automatiquement cette forme. Pour ce faire, la surface S2 sera utilisée comme un plan de coupe du solide obtenu par l'extrusion de la surface S1. Cette méthode est reprise dans la figure (Fig7.16).

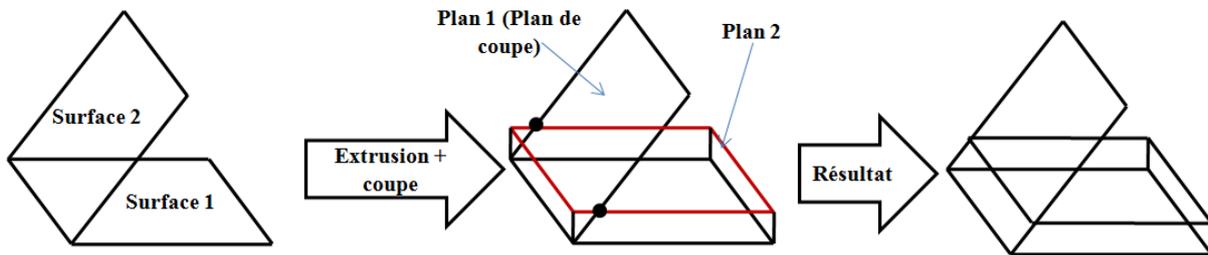


Fig7.16. Correction des formes invalides

Les étapes principales de l'algorithme de passage de B-Rep vers CSG sont expliquées dans la figure (Fig7.17).

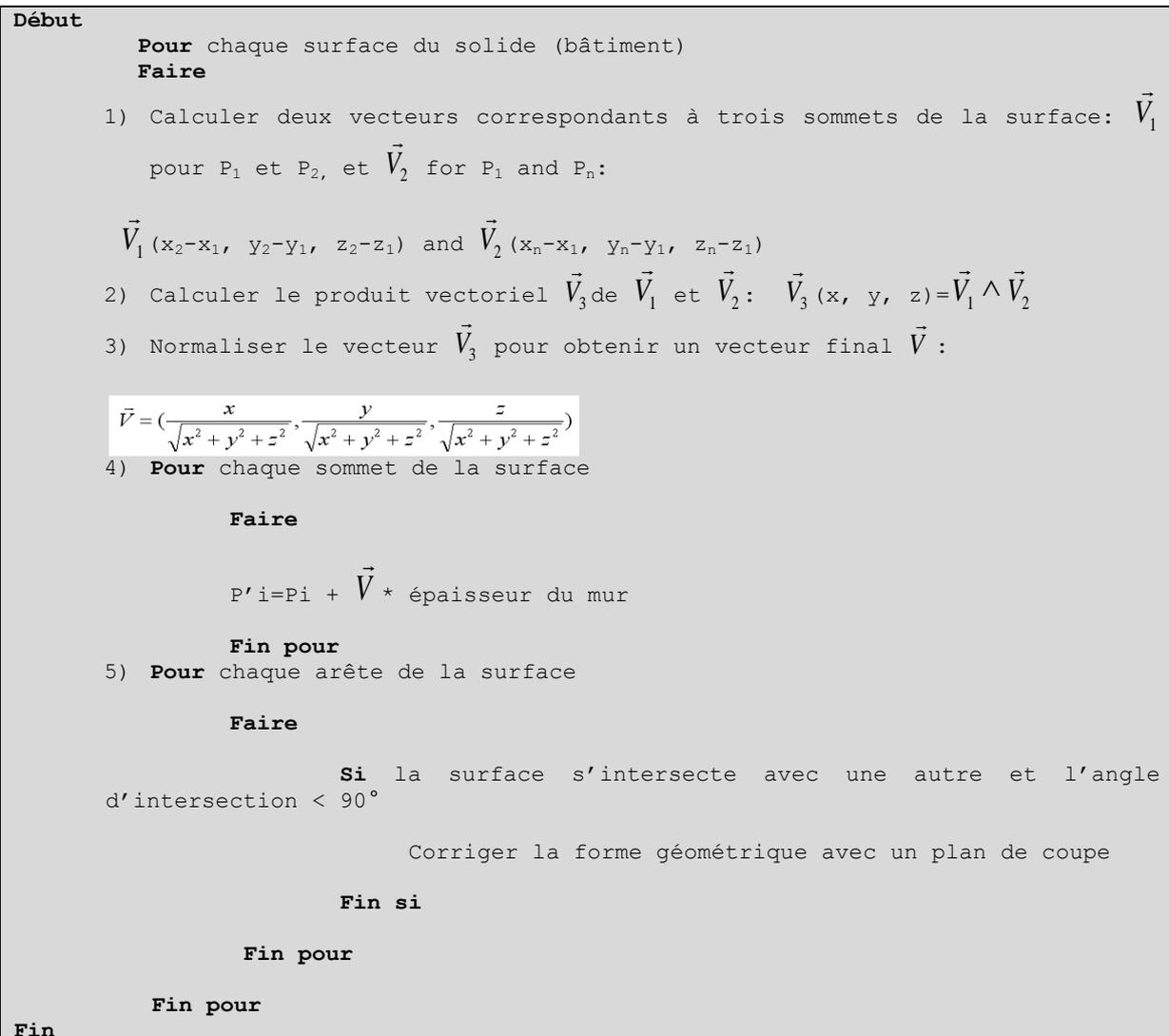


Fig7.17. L'algorithme d'extrusion des polygones B-Rep pour obtenir des solides CSG

La figure (Fig7.18) affiche un exemple d'un bâtiment représenté en tant que modèle B-Rep ainsi que le modèle B-Rep\_CSG correspondant.

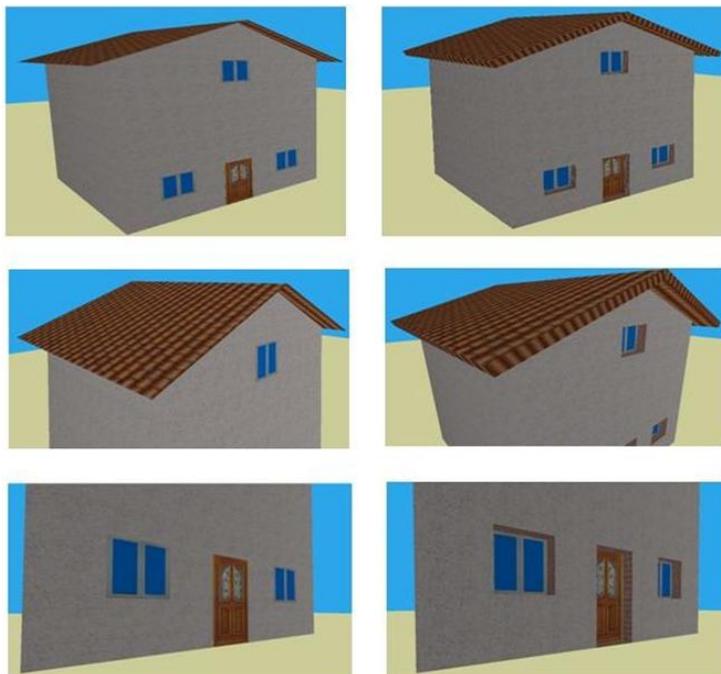


Fig7.18. Exemple d'une visualisation B\_Rep\_CSG d'un modèle 3D

Le test de l'algorithme de passage de la représentation B-Rep à la représentation CSG a montré une certaine valeur ajoutée en matière de visualisation tridimensionnelle des bâtiments (plus de réalisme) et d'optimisation de stockage de données (on n'ajoute qu'une seule donnée correspondant à l'épaisseur du mur). Cependant, cette représentation n'est utilisée actuellement que pour des raisons de visualisation. Elle pourra être étendue pour d'autres fonctionnalités notamment lors d'analyses spatiales ou de simulations, etc.

## 2. MOTEUR 3D

On entend par moteur 3D, appelé aussi moteur de rendu 3D « *des logiciels ou des cartes graphiques spéciales (matériel) qui calculent une ou plusieurs images 3D en y restituant non seulement la projection 3D mais surtout tous les effets d'éclairage (ombres, réflexions, etc.). C'est ce qu'on appelle le rendu. L'opération est assez complexe et dépend du type de moteur de rendu ; pour simplifier, le moteur de rendu 3D va analyser tous les éléments de la scène concourant à la couleur et à l'intensité de chaque pixel de ce que voit la caméra dans la vue du programme de 3D* » [SENAGENT, 2010]. Un moteur 3D implique alors différents types de matériel d'un ordinateur (processeur, carte graphique, périphériques d'entrée-sortie, etc.).

La mise en place d'un moteur 3D dépend essentiellement des besoins attendus de ce moteur 3D (un moteur 3D pour des jeux vidéo, pour un simulateur de réalité virtuelle, pour un SIG 3D, etc.). Le développement d'un moteur 3D est une étape relativement complexe car le développeur doit avoir *a priori* des connaissances théoriques variées notamment en termes de concepts 3D, de géométrie, de matrices, de mathématiques, etc. Vu qu'il n'est pas possible de décrire tous les aspects du développement du moteur 3D, nous nous limitons à la description des applications fondamentales de ce même moteur. Nous distinguons deux types d'applications : les applications statiques consistant essentiellement à définir les propriétés des objets 2D/3D surtout en termes de géométries et caractéristiques visuelles, et les applications dynamiques permettant de manipuler et d'animer ces objets 2D/3D.

Nos travaux portent essentiellement sur :

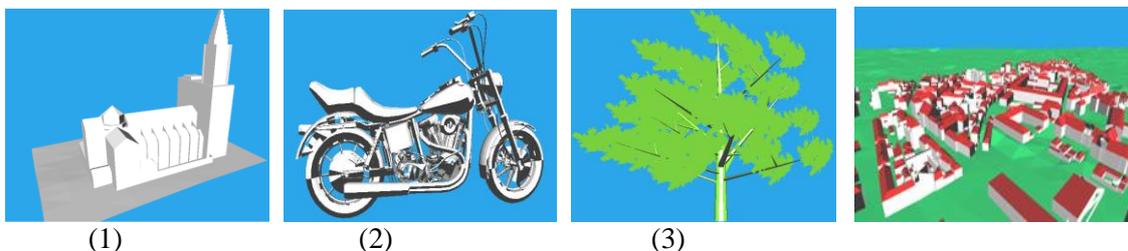
- **Géométrie**

Le moteur 3D doit pouvoir représenter les primitives géométriques que nous avons définies dans le modèle conceptuel géométrique. Comme nous utilisons OpenGL, nous distinguons deux types de primitives géométriques :

- Primitives géométriques de base : Ce sont les primitives géométriques proposées par OpenGL pour représenter tout type d'objet par des points, des triangles, des quads, des polygones, etc. OpenGL permet aussi de représenter des primitives solides comme des sphères, des cylindres ou encore des meshes<sup>2</sup>.

- Primitives géométriques solides : Nous utilisons ces primitives de base pour le dessin des solides. Pour ce faire, nous avons élaboré une structure spéciale permettant de définir des objets solides composés par un ensemble de surfaces B-Rep.

On pourra ainsi qualifier le moteur 3D comme un visionneur d'objets 2D/3D. La figure (Fig7.19) évoque quelques exemples de modèles 3D rendu par le moteur 3D.



**Fig7.19.** Exemple d'objets 3D de différents types rendus par le moteur 3D  
(Source des modèles (1), (2) et (3): Banque d'images Google)

- (1) Rendu 3D simplifié de la cathédrale de Strasbourg
- (2) Rendu 3D d'une moto
- (3) Rendu 3D d'un arbre
- (4) Rendu 3D d'un quartier de la ville de Strasbourg

Le moteur 3D est capable d'afficher aussi bien des objets isolés (bâtiments) que des groupes d'objets (quartiers, villes).

- **Apparences**

Les apports de la 3D par rapport à la 2D résident dans sa capacité à représenter des objets avec un niveau de réalisme plus ou moins important.

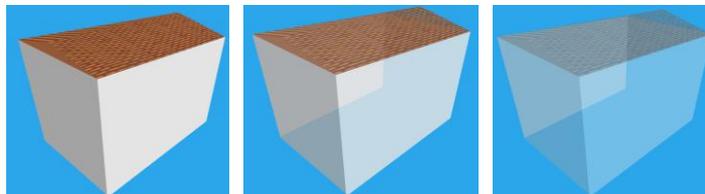
Différentes techniques sont utilisées pour représenter les apparences visuelles des objets 3D :

- Textures : les textures sont employées pour ajouter des détails sur les surfaces des objets 3D en utilisant un placage d'images. Le rendu du modèle 3D est ainsi appelé photo-réaliste. Dans notre moteur 3D, nous implémentons une possibilité de texturation automatique des surfaces.
- Couleurs : les couleurs sont très utiles pour représenter d'une manière différente un ensemble d'objets 3D rendant dans certains cas leur interprétation plus facile.

<sup>2</sup> Dans le domaine de l'infographie 3D et la modélisation des solides, un mesh est défini comme l'ensemble de sommets, arcs et faces composant un objet polyédrique.

- Effet spéciaux : Des effets spéciaux comme la luminosité, la transparence, l'ombre, etc. sont capables de rendre une scène 3D plus réaliste.

Le moteur 3D que nous avons implémenté permet à l'utilisateur de changer les apparences de son modèle de point de vue couleurs, textures, transparence, luminosité, etc.



*Fig7.20.* Exemple d'un objet représenté selon trois différents niveaux de transparence

- **Animation**

Une fois rendu à l'écran, un modèle 3D doit pouvoir être animé pour percevoir les objets des différents angles de vue, ce qui correspond à la notion de base de la 3D. Le moteur 3D permet ainsi à l'utilisateur de se déplacer dans son modèle 3D dans tous les sens, le faire pivoter selon les 3 axes, etc. OpenGL propose un ensemble de techniques pour la translation et la rotation des modèles 3D en utilisant la notion de matrices. Nous avons développé un ensemble de classes Java pour la gestion de l'animation au sein du moteur 3D. En particulier, pour la rotation des modèles 3D, nous nous basons sur la manipulation de la caméra définie par sa position, son orientation et le vecteur de l'axe utilisé (axe des X, des Y ou des Z). Pour ce faire, nous avons utilisé la notion de quaternions (représentés par des vecteurs de 4 nombres :  $w + xi + yj + zk$ ) au lieu des matrices de rotation utilisant 9 paramètres. Ceci évite de calculer des multiplications entre matrices et donne plus de souplesse et de rapidité à la rotation. La classe Java permettant de définir un quaternion est décrite dans l'annexe (page 226).

- **Manipulation d'objets**

La manipulation d'objets commence par leur sélection dans le modèle 3D. Dans OpenGL, la sélection d'objets est appelée "picking", se base sur les coordonnées du pointeur et identifie l'objet le plus proche.

La sélection graphique des objets 3D est couplée avec la sélection attributaires (en se basant sur des données provenant de la Base de Données) pour permettre à l'utilisateur une sélection d'objets plus souple.

- **Techniques d'optimisation**

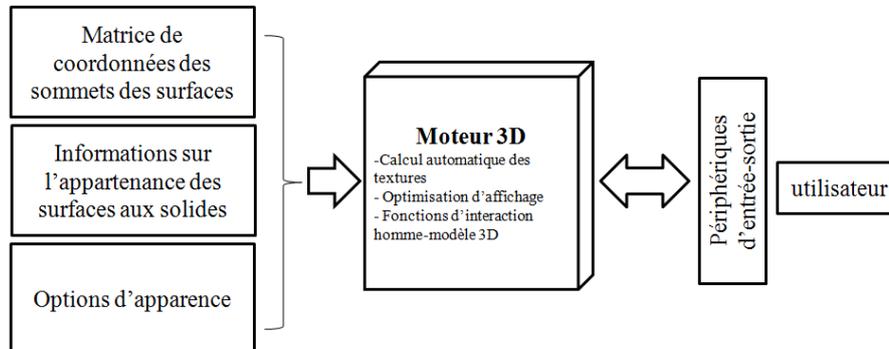
Un moteur 3D utilise généralement beaucoup de ressources (processeur, mémoire vive, carte graphique, etc.) pour rendre une scène 3D. Son optimisation s'avère être une nécessité.

Pour ce faire, nous avons appliqué plusieurs procédures d'optimisation dont les plus importantes sont :

- ⇒ Clipping : le principe de base du clipping est de ne pas calculer des objets en dehors du champ de vision. Leur calcul se fait automatiquement lorsqu'ils apparaissent à l'écran. Cette méthode est très utile pour des scènes volumineuses ou complexes.
- ⇒ Vertex Buffer Objects (VBO) : Pour optimiser le rendu nous avons testé plusieurs techniques OpenGL telles que les "Display List", "Array List" et enfin les "Vertex Buffer Objects". La technique des "Vertex Buffer Objects" est la plus rapide car elle stocke d'une manière efficace des coordonnées ou des pixels en optimisant l'utilisation de la carte graphique. L'apport de cette

technique est plus visible pour les scènes volumineuses notamment en termes de gains du point de vue de la rapidité d'affichage.

En se basant sur les différents concepts décrits ci-dessus, nous pouvons ainsi définir un schéma très simplifié du principe de fonctionnement du moteur 3D développé dans le cadre de notre travail.



**Fig7.21.** Fonctionnement du moteur 3D

### 3. COMPARAISON AVEC LES OUTILS COMMERCIAUX

Lors du choix du mode de représentation des données spatiales, nous nous sommes inspirés des outils commerciaux existants, en particulier ArcGIS et MapInfo. En effet, le principe de représentation est le même avec quelques modifications.

Les fonctionnalités communes avec les outils commerciaux sont les suivantes :

- Pour représenter plusieurs lots de données dans le même modèle, nous avons adopté un système de couches spatiales.
- Mise en place des mêmes représentations des icônes utilisés (les widgets) par la plupart des outils commerciaux pour alimenter notre SIG par quelques fonctionnalités basiques de visualisation telles que la rotation, le déplacement horizontal, vertical et en profondeur, le zoom, des fonctionnalités de recherche telles que la recherche d'un élément par ses valeurs d'attributs, l'affichage des informations concernant un élément particulier (identification), la sélection des éléments, etc.

Cependant, il nous paraissait intéressant d'ajouter de nouvelles fonctionnalités pour la représentation des couches spatiales.

Les rajouts sont décrits dans ce qui suit :

- Notre application passe par une étape d'importation des fichiers vers un format de base de données (PostgreSQL/PostGIS) puis transfère les données nécessaires pour la visualisation des modèles 3D. Les logiciels commerciaux intègrent de nombreux formats ou travaillent avec des conteneurs spécifiques comme des géodatabases.
- Notre version de SIG propose à l'utilisateur deux alternatives d'affichage : il peut soit afficher tout le modèle correspondant à un fichier particulier, soit filtrer le contenu du modèle avant son rendu 3D (notion de sous-ensemble de sélection).

Les fonctionnalités que nous avons ajoutées sont les suivantes :

- Pour créer une couche spatiale répondant à ses critères, l'utilisateur peut interroger, en même temps, plusieurs tables de la BDS. L'interrogation peut porter sur des propriétés géométriques ou

- attributaires. Cela peut rendre la création des couches spatiales plus flexible et permet ainsi de ne rendre à l'écran que les données dont l'utilisateur a besoin et donc minimiser la charge du système
- Comme l'utilisateur peut interroger plusieurs tables en même temps, la requête SQL correspondante peut nécessiter de préciser formellement les jointures entre les tables. Pour faciliter la tâche de l'utilisateur, le SIG propose à l'utilisateur d'écrire séparément, des conditions sur les attributs par table, les jointures entre les tables étant interprétées automatiquement et de manière transparente pour l'utilisateur.
  - Avant son exécution la requête SQL correspondant aux objets à sélectionner et représenter est constituée et validée automatiquement par le système.
  - L'utilisateur choisit les couleurs et les textures du modèle *a priori* avant son rendu. L'utilisateur peut aussi utiliser des textures stockées sur sa machine locale sous forme d'images. Comme cela expliqué dans la partie 2 de ce chapitre (Moteur 3D), les textures sont plaquées automatiquement sur les surfaces grâce à une fonctionnalité OpenGL dédiées.

### CONCLUSION

Nous avons souhaité offrir un prototype de SIG proposant un ensemble d'interfaces graphiques où l'utilisateur serait guidé dans ses choix mais resterait libre dans ses décisions pour la création et la représentation des modèles 3D à travers un système de couches spatiales. Avant de rendre le modèle 3D, l'utilisateur peut filtrer le contenu des sources de données en imposant des conditions sur les propriétés géométriques et attributaires des objets spatiaux. Le modèle spatial 3D rendu par l'application répondra exactement aux choix et critères de l'utilisateur.

Une fois que les couches spatiales sont rendues à l'écran, l'utilisateur peut réaliser quelques modifications sur leurs apparences. Il peut aussi effectuer des opérations booléennes sur les couches pour interpréter d'autres couches spatiales. L'une des caractéristiques fondamentales d'un SIG est sa capacité à interroger les modèles spatiaux pour en rechercher des objets particuliers, et sa capacité à les représenter par différents symboles et textures pour les distinguer les uns des autres. Cela fait partie des fonctionnalités envisagées de notre prototype de SIG. Les opérations booléennes sur les couches spatiales et leur stockage dans la base de données ainsi que les fonctionnalités d'interrogation des modèles 3D sont expliquées en détail dans le chapitre suivant.



**CHAPITRE.8 INTERROGATION DES DONNEES SPATIALES**

Résumé.....	152
Introduction .....	152
<b>1. Traitement des couches spatiales</b> .....	152
1.1. Opérations sur les couches spatiales.....	153
1.2. Stockage des couches spatiales .....	157
<b>2. Traitement des objets spatiaux</b> .....	159
2.1. Requêtes topologiques.....	159
2.2. Requêtes attributaires .....	165
<b>3. Positionnement du travail par rapport aux outils existants</b> .....	176
Conclusion.....	177

*« Les outils de type CAO (Conception Assistée par Ordinateur) ... permettent d'effectuer de la modélisation géométrique bidimensionnelle et tridimensionnelle mais sont souvent limités au niveau des méthodes de stockage et de requêtage des données géométriques »  
(POULIOT, 2003)*

### RESUME

*Un SIG dépasse la simple visualisation des données spatiales. En effet, il dispose d'un ensemble d'outils permettant à l'utilisateur de représenter, d'interroger et d'analyser les modèles spatiaux. C'est dans ce contexte que s'inscrit ce chapitre. Nous parlerons de la combinaison des couches spatiales en utilisant des opérateurs booléens. Et nous décrirons notre méthode pour l'interrogation des couches spatiales pour en extraire des objets spatiaux selon les préférences de l'utilisateur.*

### INTRODUCTION

L'avantage majeur des SIG par rapport aux outils CAO (Conception Assistée par Ordinateur) est leurs capacités de stockage, d'interrogation et de représentation des données géométriques. En effet, un SIG doit pouvoir permettre à ses utilisateurs de chercher, dans les modèles des données spatiales, des objets spatiaux répondant à des critères donnés afin d'analyser un phénomène spatial particulier. L'interrogation de tels modèles est généralement matérialisée par des requêtes sur les jeux de données sources pour en extraire un sous-ensemble d'objets spatiaux. L'interrogation peut porter sur la localisation des objets dans l'espace, sur leurs propriétés géométriques, sur leurs relations de voisinage, sur leurs propriétés attributaires, etc. Le but de ce chapitre est de présenter nos travaux en termes de traitement des couches spatiales et interrogation des modèles 3D. Le traitement des couches spatiales porte sur leur combinaison pour en interpréter d'autres couches plus sophistiquées. L'interrogation des modèles 3D est matérialisée par la proposition d'une interface graphique facilitant la tâche de l'utilisateur et optimisant l'interrogation.

Ce chapitre est organisé de la manière suivante. La première partie décrira la méthode permettant la combinaison des couches spatiales en utilisant des opérations booléennes ainsi que le langage graphique permettant de modéliser ces opérations booléennes. Les couches spatiales peuvent également être stockées pour une utilisation future ou pour être partagées par plusieurs utilisateurs. La deuxième partie présentera l'interface graphique ainsi que le principe d'interrogation des modèles 3D pour en extraire des objets spatiaux répondant aux besoins définis par l'utilisateur.

### 1. TRAITEMENT DES COUCHES SPATIALES

Le SIG permet à l'utilisateur de personnaliser le contenu ainsi que les apparences de ses modèles 3D à travers un système de couches spatiales. Il peut interroger les mêmes sources de données pour créer des modèles 3D différents, du point de vue du contenu et du point de vue des apparences tout en répondant à des besoins diversifiés.

La motivation de l'utilisation d'un système de couches se résume dans les points suivants :

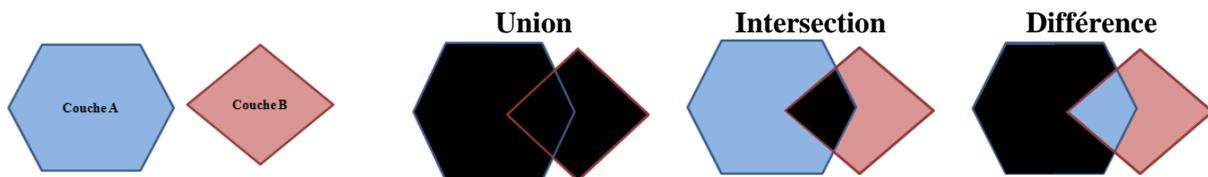
- En interrogeant la BDS pour créer une couche spatiale correspondant uniquement à un sous-ensemble de données, les modèles 3D affichés seront plus légers et les performances du système optimisées.
- La superposition des modèles 3D à travers un ensemble de couches spatiales peut apporter une valeur ajoutée en termes d'interprétation et d'analyse. En effet, les couches permettent généralement des représentations différentes des modèles 3D et facilitent la distinction des différentes caractéristiques spatiales des objets spatiaux.

- L'utilisateur peut travailler proprement et séparément sur les différentes couches spatiales. Il peut ensuite créer de nouvelles couches déduites, stocker ces couches pour un usage futur, les partager avec d'autres utilisateurs, etc.

### 1.1. OPERATIONS SUR LES COUCHES SPATIALES

Un utilisateur du SIG peut interroger la même BDS ou des BDS différentes pour créer une ou plusieurs couches spatiales. Il est libre d'effectuer différentes actions sur chaque couche (modifier ses représentations, y rechercher des objets spatiaux, y ajouter des sous-couches, etc.). Ces couches spatiales peuvent avoir des propriétés communes. Pour ce faire, le SIG propose aux utilisateurs différentes opérations booléennes pour créer des couches à partir d'autres couches.

Le SIG permet de combiner plusieurs couches dans une seule couche en utilisant les opérations booléennes : union, intersection et différence. L'union permet d'ajouter le contenu d'une couche à une autre pour obtenir une couche contenant toutes les primitives géométriques (polygones pour les modèles B-Rep et polygones pour les modèles filaires) des deux couches. L'intersection permet d'obtenir une couche composée uniquement des primitives géométriques communes aux deux couches en question. Et enfin, la différence est une opération booléenne dont le résultat dépend de l'ordre des opérandes. Elle permet d'obtenir une couche composée de toutes les primitives géométriques de la première couche démunies de toutes les primitives partagées par les deux couches.



*Fig8.1.* Schéma illustratif des opérations booléennes : union, intersection et différence entre deux couches spatiales

Le principe d'utilisation des opérations booléennes pour combiner des couches spatiales est important dans les outils SIG. Il permet d'interpréter des modèles 3D à partir d'autres permettant ainsi à l'utilisateur plus de possibilités d'interprétation et de prise de décision. Cependant, il faut trouver une méthode optimale pour réaliser ces opérations booléennes.

Comme la création des couches spatiales repose sur des requêtes SQL sur la BDS, nous utilisons également des requêtes pour mettre en place les différents types d'opérations booléennes. En effet, SQL permet de réaliser les différentes opérations booléennes à partir de requêtes de type SELECT. Nous utilisons cette fonctionnalité de SQL pour appliquer les différentes opérations booléennes sur les modèles 3D.

Notre choix d'utiliser les opérations booléennes de SQL est motivé par :

- Performance de SQL : l'utilisation de SQL peut nous assurer une performance importante en termes de rapidité d'exécution, d'exactitude des résultats, etc.
- Rapidité : l'utilisation de SQL nous évite d'écrire des algorithmes pouvant être lourds ou mal optimisés.
- Simplicité : l'utilisation de SQL permet d'optimiser le code et donc de limiter les risques d'erreurs tout en offrant une plus grande rapidité d'exécution du code.
- Non redondance de données : les opérations booléennes de SQL sont optimisées de façon à ce que les éléments des tables interrogées ne se répètent pas dans le résultat (distinct).

L'organigramme d'une opération booléenne entre deux requêtes SQL correspondant à deux couches spatiales est décrit dans la figure (Fig8.2).

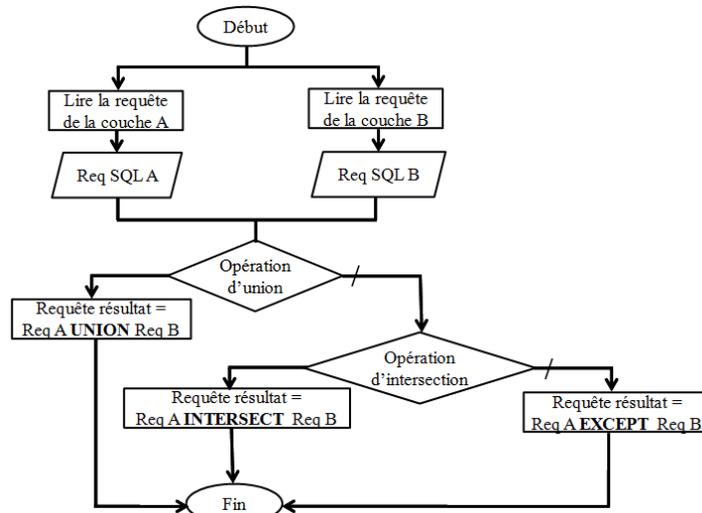


Fig8.2. Organigramme d'une opération booléenne entre deux requêtes SQL correspondant à deux couches spatiales

SQL simplifie la tâche. La méthode se base sur des notions SQL connues et faciles à mettre en œuvre. Cependant, il faut trouver une méthode simplifiée pour l'utilisateur afin qu'il puisse effectuer aisément de telles opérations. Pour ce faire, nous avons développé une méthode graphique à travers laquelle l'utilisateur peut créer ses opérations booléennes à partir d'un ensemble de manipulations simples et entièrement graphiques. La figure (Fig8.3) illustre la représentation graphique d'une opération booléenne entre deux couches spatiales.

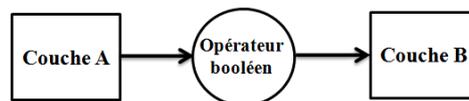


Fig8.3. Représentation graphique d'une opération booléenne entre deux couches spatiales

Une couche spatiale est représentée par un rectangle, un opérateur par un cercle et les liens sont représentés par des flèches. Cette méthode graphique est plus importante lorsque l'utilisateur désire enchaîner, dans la même requête, des opérations booléennes sur plusieurs couches spatiales. Cette technique permettra de personnaliser des opérations booléennes plus sophistiquées, d'une part, et, d'autre part, accélèrera la procédure d'interprétation d'une couche spatiale à partir d'autres couches. La figure (Fig8.4) montre un exemple d'une opération booléenne sur plusieurs couches spatiales.

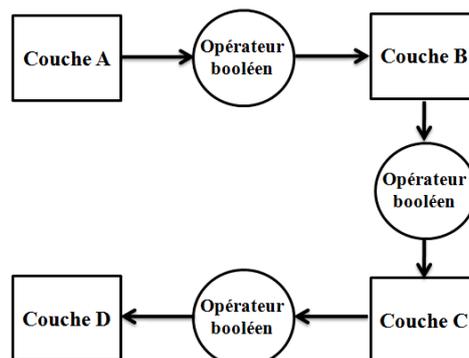


Fig8.4. Représentation graphique d'une opération booléenne entre plusieurs couches spatiales

Le résultat des opérations booléennes de la figure (Fig8.4) peut être interprété comme :

((Couche A "opérateur" Couche B) "opérateur" Couche C) "opérateur" Couche D)

La méthode peut être considérée comme incrémentale car chaque opérateur booléen est appliqué sur le résultat de l'opération précédente et sur une nouvelle couche spatiale.

Le SIG propose une bibliothèque de symboles et de liens pour que l'utilisateur puisse dessiner aisément et rapidement le graphique correspondant à ses opérations booléennes sur les couches spatiales. La figure (Fig8.5) montre l'interface graphique qui permet la création de tels graphiques.

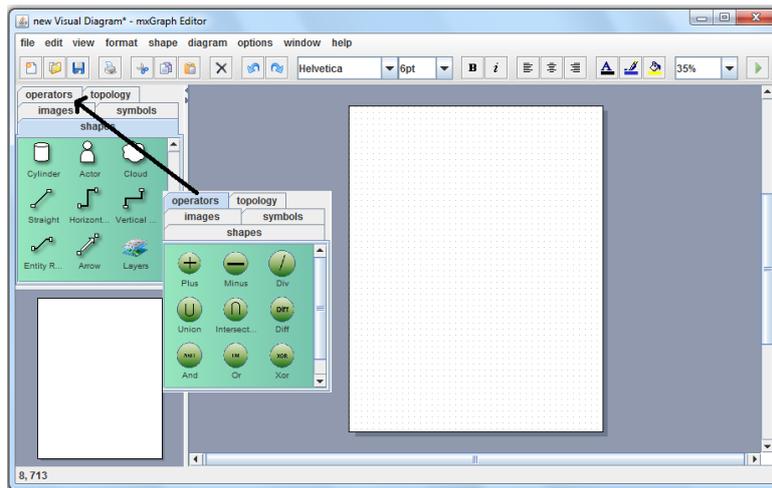


Fig8.5. Quelques exemples de symboles utilisés pour dessiner une opération booléenne

D'un point de vue fonctionnel, nous avons développé un algorithme pour l'interprétation de la requête SQL correspondant à la couche spatiale, résultat des opérations booléennes sur deux ou plusieurs couches spatiales. L'algorithme utilise la représentation graphique comme un graphe connexe non orienté (les flèches sont utilisées pour des raisons de programmation pour distinguer le nœud source du nœud cible pour chaque lien). Les couches spatiales et les opérateurs booléens constituent les nœuds et les flèches constituent les liens du graphe. L'algorithme qui calcule la requête SQL finale est expliqué dans la figure (Fig8.6).

#### Paramètres

Nœuds Graphe : Tableau de chaînes de caractères //Liste d'identifiants des nœuds du graphe  
Liens Graphe : Tableau de chaînes de caractères //Liste d'identifiants des liens du graphe

#### Variables

RequêteSQL\_Résultat : chaîne de caractère // La requête SQL de la couche spatiale résultat  
Est\_Connexe : Variable booléenne // Pour tester si le graphe est connexe  
Est\_Bien Formé : Variable booléenne // Pour tester si le premier et le dernier nœud //du graphe ne sont pas des opérateurs  
Est\_Valide : Variable booléenne // Pour tester si entre chaque deux nœuds //correspondant à des couches il existe un opérateur booléen

#### Début

```
//1) Vérifier si le graphe est connexe (pour tester si le graphe contient ou non des
nœuds isolés)
    Est_Connexe = vrai
    POUR i allant de 1 jusqu'à la taille du tableau Nœuds_Graphe ET Est_Connexe =
vrai
        FAIRE
            SI le nœud à la position i n'est pas une source ou une //cible d'un
lien Le nœud est isolé
                Est_Connexe = faux
            FIN SI
        FIN POUR
//2) Vérifier si le premier et le dernier nœud correspondent à des couches //spatiales
(ne sont pas des opérateurs booléens)
    Est_Bien Formé = vrai
    SI Est_Connexe = faux OU (premier nœud est un opérateur OU dernier nœud est un
opérateur)
```

```

    Est Bien Formé = faux
    FIN SI
    //3) Vérifier si entre chaque deux nœuds correspondant à des couches spatiales //existe
    un opérateur booléen
    POUR i allant de 1 jusqu'à la taille du tableau Liens_Graphe ET Est_Connexe =
    vrai ET Est_Bien_Formé = vrai
    FAIRE
        SI (le nœud source et le nœud cible du lien à la position i
        correspondent à des couches) OU
            (le nœud source et le nœud cible du lien à la position i
            correspondent à des opérateurs)

            Est_Valide = faux

    FIN SI

FIN POUR

//4) Calcul de la requête de la couche spatiale résultat

SI Est_Connexe = vrai ET Est_Bien_Formé = vrai ET Est_Valide = vrai
    POUR i allant de 1 jusqu'à le nombre de nœuds de type opérateur
    FAIRE
        SI RequêteSQL_Résultat = ""
            RequêteSQL_Résultat = "(" + RequêteSQL_Source()
+ Opérateur + RequêteSQL_Cible() + ")"
        SINON
            RequêteSQL_Résultat = "(" + RequêteSQL_Résultat
+ Opérateur + RequêteSQL_Cible() + ")"
    FIN POUR
FIN SI
Fin

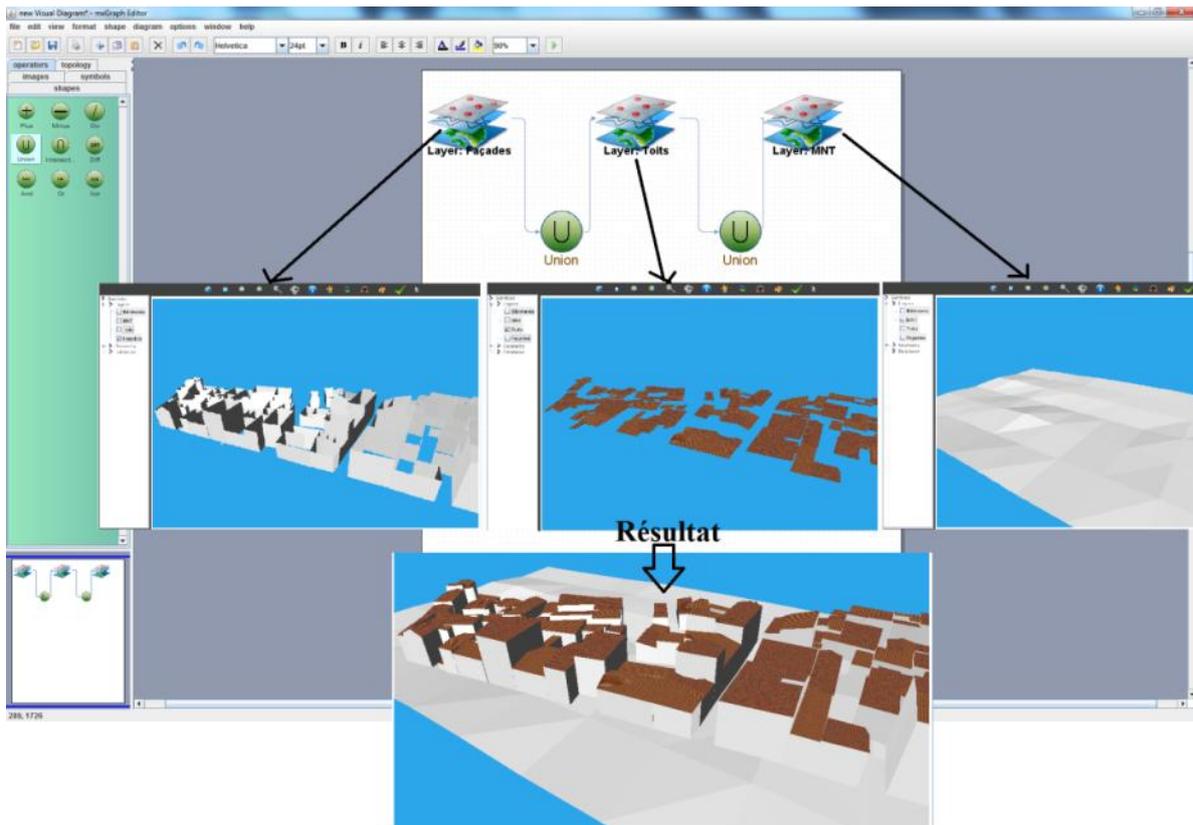
```

**Fig8.6.** Algorithme de calcul d'une opération booléenne entre deux couches spatiales

Cet algorithme est divisé en quatre étapes principales :

- 1) La première étape consiste à vérifier si le graphe est connexe afin d'éviter d'avoir des nœuds isolés.
- 2) La deuxième étape permet de vérifier si le premier et le dernier nœud correspondent à des couches et non à des opérateurs.
- 3) La troisième étape vérifie si le graphe est bien formé, c'est-à-dire qu'il existe un opérateur booléen entre chaque couple de nœuds correspondant à des couches spatiales.
- 4) Si les trois premières étapes aboutissent à des résultats positifs, on calcule la requête SQL représentant le résultat des différentes opérations booléennes interprétées à partir du graphe.

L'algorithme s'exécute automatiquement et d'une manière transparente à l'utilisateur. Quand l'algorithme a reconstitué la requête SQL permettant d'obtenir la nouvelle couche spatiale, l'application demande à l'utilisateur un nom et une description significative à associer à cette nouvelle couche. L'utilisateur peut alors personnaliser les apparences (couleurs, textures, etc.) de la couche. Si les couches à combiner ont des apparences différentes et si l'utilisateur ne personnalise pas les apparences de la couche résultat, cette dernière aura autant d'apparences que les couches combinées (des couleurs ou des textures différentes par exemple).



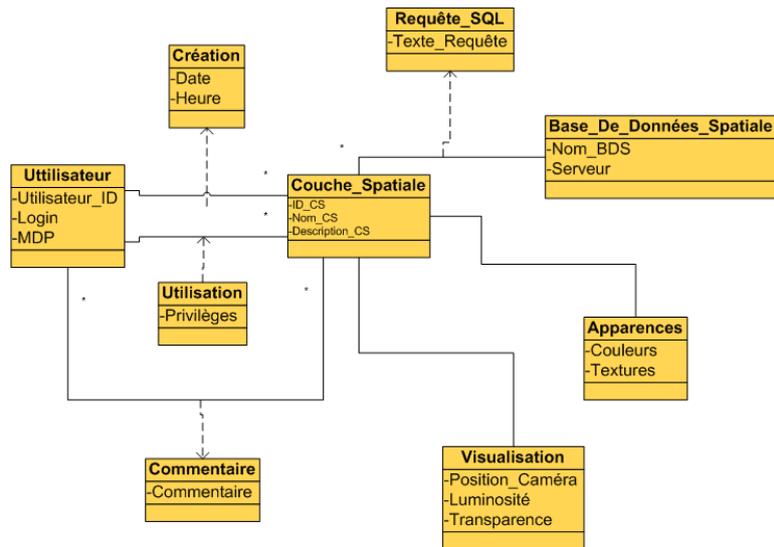
**Fig8.7.** Exemple d'une requête d'union entre trois couches spatiales : Façades, Toits et MNT

L'opération booléenne, expliquée dans la figure (Fig8.7) a eu comme objectif la création d'une couche spatiale appelée "Bâtiments" en appliquant deux opérations d'union entre les couches spatiales "Façades", "Toits" et "MNT". La figure (Fig8.7) désigne la requête graphique matérialisée par un graphe contenant cinq nœuds (un nœud pour chaque couche spatiale et un nœud pour chaque opérateur booléen) et quatre arcs (les liens entre les nœuds).

## 1.2. STOCKAGE DES COUCHES SPATIALES

Le système à couches a justifié son importance dans les SIG car il autorise l'utilisateur à personnaliser la représentation des ses lots de données en superposant plusieurs modèles spatiaux. Comme notre SIG permet de créer une couche spatiale personnalisée en interrogeant une BDS, nous considérons que le stockage de telle couche est important si l'utilisateur désire réutiliser la même couche spatiale ou la partager avec d'autres utilisateurs.

Nous expliquons d'une manière simplifiée les caractéristiques d'une couche spatiale à travers un méta-modèle dans la figure (Fig8.8).



**Fig8.8.** Méta-Modèle d'une couche spatiale

Selon le méta-modèle, une couche spatiale est définie par :

- Nom : Un nom donné par l'utilisateur à la couche facilitant ainsi son identification et sa recherche pour des utilisations futures.
- Description : Une description sémantique et significative créée par l'utilisateur permettant ainsi sa recherche pour des utilisations futures et sa compréhension par les autres utilisateurs.
- Date de création : La date où la couche est créée pour la première fois.
- Créateur : Un utilisateur identifié par un login et un mot de passe.
- Requête SQL : La requête utilisée pour interroger la BDS pour créer la couche spatiale.
- BDS : La BDS interrogée pour la création de la couche spatiale.
- Commentaire : Les commentaires de l'utilisateur sur la couche spatiale.
- Apparences : Les apparences physiques de la couche spatiale (couleurs, textures, etc.).
- Visualisation : Les caractéristiques de visualisation de la couche spatiale telle que la position de la caméra, la luminosité, la transparence, etc.
- Utilisation : La liste d'utilisateurs autorisés à accéder à cette couche spatiale selon différents privilèges.

Toutes les caractéristiques de la couche spatiale sont stockées dans la base de données pour qu'elle puisse être réutilisée dans le futur. Nous traduisons donc le méta-modèle vu ci-dessus en un modèle logique tel que c'est décrit dans la figure (Fig8.9).

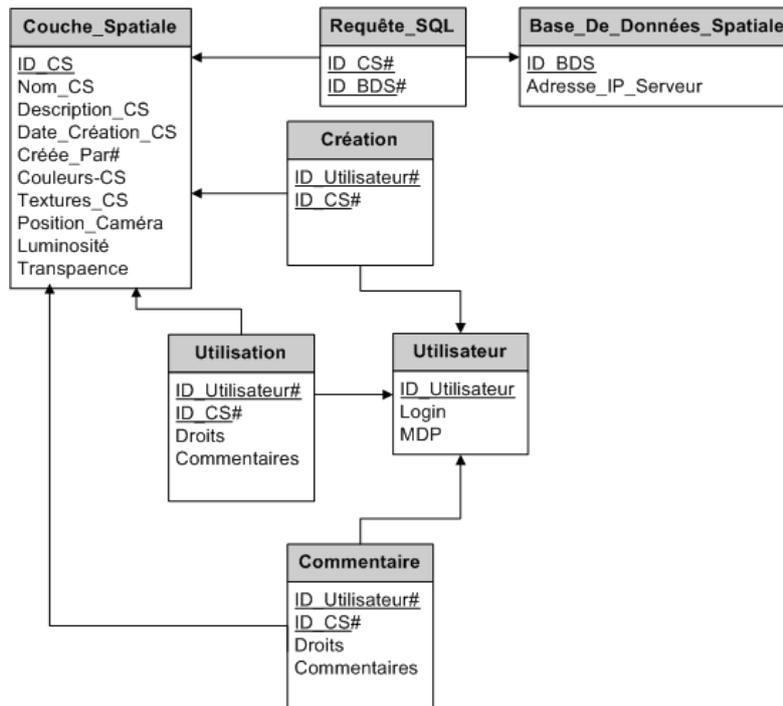


Fig8.9. Modèle logique relationnel d'une couche spatiale

Les caractéristiques des différentes couches spatiales sont stockées dans la même BDS à partir de laquelle les jeux de données sont extraits pour la création de la couche spatiale.

## 2. TRAITEMENT DES OBJETS SPATIAUX

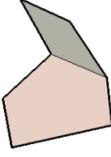
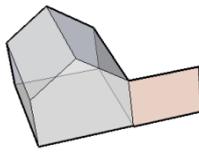
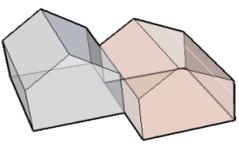
L'importance des SIG réside dans leur manière de représenter les données spatiales. Ils sont capables d'organiser les données à travers des sous-ensembles ayant des apparences visuelles différentes. L'interprétation des données devient alors plus facile selon la thématique de chaque couche spatiale. Mais le SIG doit offrir la possibilité d'interroger les couches spatiales pour en extraire des sous-ensembles d'objets spatiaux répondant à des besoins particuliers ainsi que les relations de voisinage de ces mêmes objets. Cette partie apporte les informations nécessaires à la compréhension des concepts développés.

### 2.1. REQUETES TOPOLOGIQUES

L'analyse topologique est l'une des fonctionnalités les plus importantes pouvant être fournies par un SIG. Une réflexion sur les relations topologiques entre des objets spatiaux 3D, en particulier des bâtiments, est menée dans le cadre de ce travail. Parmi les motivations d'utilisation du SGBD (Système de Gestion de Base de Données) spatial PostgreSQL/PostGIS on retrouve sa capacité à stocker de données spatiales et à fournir des fonctions spatiales très intéressantes pour manipuler les données. Il propose un ensemble de fonctions topologiques telles que *ST\_INTERESECTS*, *ST\_CONTAINS*, *ST\_OVERLAPS*, pour lesquelles les fonctions correspondent à leur traduction (intersecte, contient, recouvre), etc. Cependant, ces fonctions sont exécutées sur des objets 2D. Dans ses versions actuelles, PostGIS ne propose pas encore de fonctions pour calculer des opérations topologiques sur des objets 3D tels que les solides et les polyèdres. Comme ce travail porte sur des objets 3D, en particulier des bâtiments, les fonctions spatiales de PostGIS seront utilisées et étendues par des fonctionnalités topologiques pour calculer les relations spatiales entre des objets 3D.

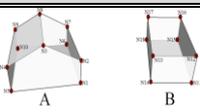
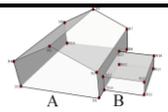
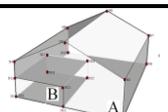
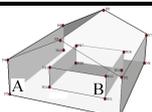
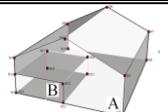
Nous distinguons trois types de relations topologiques entre des objets spatiaux 3D : Relation polygone-polygone (relation entre deux façades, par exemple), relation polygone-solide (relation entre

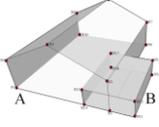
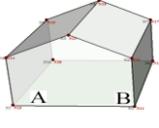
une façade et un bâtiment, par exemple) et relation solide-solide (relation entre deux bâtiments, par exemple). Le tableau (Tab8.1) illustre de telles relations.

		
(a) Polygone-Polygone	(b) Polygone-Solide	(c) Solide-Solide

Tab8.1 Différents types de relations topologiques entre deux solides 3D

Comme PostGIS n'offre pas non plus la possibilité de stocker des solides 3D, nous avons implémenté une structure particulière pour la gestion des solides. Cette structure a été décrite dans le chapitre 5 « *Intégration & création des jeux de données spatiales* ». Cette extension de structure permet le stockage des solides 3D sous la forme d'une collection de surfaces B-Rep et d'un ensemble non ambiguë de sommets. D'autres fonctions de PostGIS comme l'intersection, la distance, etc. sont utilisées pour calculer des relations topologiques entre les surfaces composant les objets 3D, les façades des bâtiments, par exemple. Les fonctions de PostGIS ont été complétées par un ensemble de fonctions pour le calcul des opérations topologiques sur des solides 3D, en particulier des bâtiments. Le tableau (Tab8.2) illustre les différentes relations topologiques pouvant apparaître entre deux bâtiments et inspirées de [ZLATANOVA, 2000].

Opération topologique	Illustration	Primitives partagées	Intérieur	Extérieur
Disjoint (A, B)		$\emptyset$		
Rencontre (A, B)		Face, arc, nœud		✓
Touche (A, B)		Face, arc, nœud	✓	✓
Contient (A, B) Contenu_dans (B, A)		Volume	✓	
Couvre (A, B) Couvert_par (B, A)		Volume	✓	

Chevauche (A, B)		Volume	✓	
Egal (A, B)		Volume	✓	

**Tab8.2** Tableau des relations topologiques entre des objets 3D

Pour notre application, la structure B-Rep est légèrement modifiée pour modéliser les solides 3D, en particulier les bâtiments. Ces solides sont représentés par les surfaces composant leurs frontières ainsi que leurs sommets. Les surfaces sont représentées par leurs sommets et les sommets sont représentés par des coordonnées X, Y, Z. Il est très important de distinguer deux types de primitives géométriques : i) Primitives simples telles que les points, les surfaces, les solides, etc. ii) Primitives composées telles que les multi-points, les multi-surfaces, les multi-solides, etc.

Pour formaliser les différentes opérations topologiques sur des primitives géométriques (essentiellement des surfaces et des solides) représentant deux objets spatiaux A et B, nous avons défini quelques notations et quelques conditions :

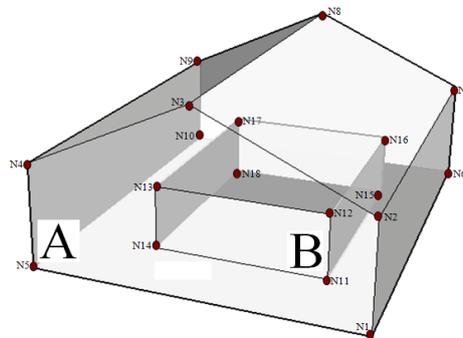
- Soient  $Dim(A)$  et  $Dim(B)$  les dimensions respectives des objets A et B, sachant que  $Dim(Objet) = 0$  si *Objet* est un point,  $Dim(Objet) = 1$  si *Objet* est un arc,  $Dim(Objet) = 2$  si *Objet* est une surface,  $Dim(Objet) = 3$  si *Objet* est un solide.
- Soient  $Prim(A)$  et  $Prim(B)$  l'ensemble de primitives simples composant respectivement les objets spatiaux A et B, et  $Dim(Prim(A)) = n$  et  $Dim(Prim(B)) = n'$ , sachant que  $(0 \leq n \leq Dim(A))$  et  $(0 \leq n' \leq Dim(B))$ .
- Soient  $Front(A)$  et  $Front(B)$  les primitives simples formant respectivement les frontières des objets spatiaux A et B, sachant que  $Front(A)$  est un sous ensemble de  $Prim(A)$  et de dimension  $Dim(A)-1$ , et  $Front(B)$  est un sous ensemble de  $Prim(B)$  et de dimension  $Dim(B)-1$ .
- Soient  $Sommets(A)$  et  $Sommets(B)$  les sommets respectifs des objets spatiaux A et B.
- Deux algorithmes sont développés pour tester respectivement si un point est à l'intérieur d'une surface et si un point est à l'intérieur d'un solide. Une relation appelée « Dans » est développée pour effectuer de tels tests. PostGIS propose une fonction appelée «ST\_WITHIN » permettant de vérifier si un point est à l'intérieur d'une surface. Et pour tester si un point est à l'intérieur d'un solide, nous avons développé un algorithme spécifique, expliqué dans les figures (Fig8.13 et Fig8.14).

Les relations topologiques sont essentiellement calculées sur des objets spatiaux de type surface (une façade, par exemple) et solide (un bâtiment, par exemple). Les différentes opérations topologiques entre deux objets spatiaux A et B sont expliquées dans ce qui suit :

- A « Disjoint » B :
  - distance (A, B)  $\neq 0$ , et,
  - $\forall p \in Sommets(A)$  et  $\forall k \in Prim(B)$  sachant que  $Dim(k) \geq 2$ ,  $\Leftrightarrow p \llcorner \neg \text{Dans} \llcorner k$ , et,
  - $\forall p' \in Sommets(B)$  et  $\forall k' \in Prim(A)$  sachant que  $Dim(k') \geq 2$ ,  $\Leftrightarrow p' \llcorner \neg \text{Dans} \llcorner k'$ .
- A « Touche » B :
  - distance (A, B) = 0,
  - $\exists p \in Sommets(A)$  et  $\exists k \in Front(B)$ , tel que  $p \llcorner \text{Dans} \llcorner k$ , ou,
  - $\exists p' \in Sommets(B)$  et  $\exists k' \in Front(A)$ , tel que  $p' \llcorner \text{Dans} \llcorner k'$ .
- A « Rencontre » B :

- distance  $(A, B) = 0$ ,
- $\exists p_1, p_2 \in \text{Sommets}(A)$  et  $\exists k \in \text{Front}(B)$ , tel que  $p_1 \ll \text{Dans} \gg k$  et  $p_2 \ll \neg \text{Dans} \gg k$ , ou,
- $\exists p'_1, p'_2 \in \text{Sommets}(B)$  et  $\exists k' \in \text{Front}(A)$ , tel que  $p'_1 \ll \text{Dans} \gg k'$  et  $p'_2 \ll \neg \text{Dans} \gg k'$ .
- $A \ll \text{Contient} \gg B$  ou  $B \ll \text{Contenu\_dans} \gg A$  :
  - $\forall p \in \text{Sommets}(B)$  et  $\forall k \in \text{Front}(A)$ ,  $\Rightarrow p \ll \text{Dans} \gg A$  et  $p \ll \neg \text{Dans} \gg K$ .
- $A \ll \text{Couvre} \gg B$  ou  $B \ll \text{Couvert\_par} \gg A$  :
  - $\forall p \in \text{Sommets}(B)$  et  $\exists p' \in \text{Sommets}(B)$  et  $\exists k \in \text{Front}(A)$ , tel que  $p \ll \text{Dans} \gg A$  et  $p' \ll \text{Dans} \gg K$ .
- $A \ll \text{Overlaps} \gg B$  :
  - $\exists p_1 \in \text{Sommets}(A)$  et  $\exists p_2 \in \text{Sommets}(A)$ , tel que  $p_1 \ll \text{Dans} \gg B$  et  $p_2 \ll \neg \text{Dans} \gg B$ , ou,
  - $\exists p'_1 \in \text{Sommets}(B)$  et  $\exists p'_2 \in \text{Sommets}(B)$ , tel que  $p'_1 \ll \text{Dans} \gg A$  et  $p'_2 \ll \neg \text{Dans} \gg A$ .
- $A \ll \text{Egal} \gg B$ 
  - $\forall p \in \text{Sommets}(A)$  et  $\exists k \in \text{Sommets}(B)$ ,  $p=k$ .

A titre d'exemple, nous expliquons l'opération topologique entre deux objets spatiaux A et B : "A  $\ll \text{Contient} \gg B$ ". Soit le schéma illustratif des deux objets représenté dans la figure (Fig8.10).



**Fig8.10.** Schéma illustratif d'un solide qui contient un autre

Soit  $\text{Prim}(A)$  l'ensemble des primitives simples composant le solide A composé de l'ensemble de sommets  $\text{Sommets}(A) = N1, N2, \dots, N10$  et  $\text{Front}(A)$  composé des surfaces correspondantes  $(N1, N2, N3, N4, N5), (N1, N6, N7, N2)$ , etc.

Et soit  $\text{Prim}(B)$  l'ensemble des primitives simples composant le solide B composé de l'ensemble de sommets  $\text{Sommets}(B) = N11, N12, \dots, N18$  et  $\text{Front}(B)$  composé des surfaces correspondantes  $(N11, N12, N13, N14), (N11, N15, N16, N12)$ , etc.

Pour vérifier si le solide A contient le solide B, on vérifie :

- Si tous les sommets appartenant au  $\text{Sommets}(B)$  du solide B sont à l'intérieur du solide A en utilisant la relation  $\ll \text{Dans} \gg$  permettant de vérifier si un point est à l'intérieur ou non d'un solide.
- Si aucun sommet des  $\text{Sommets}(B)$  est dans l'une des surfaces appartenant au  $\text{Front}(A)$  en utilisant la relation  $\ll \text{Dans} \gg$  permettant de vérifier si un point est à l'intérieur ou non d'une surface.

Le méta-modèle d'une opération spatiale est expliqué dans la figure (Fig8.11) ci-dessous. Une opération topologique peut être une opération de calcul d'adjacence, de composition ou d'intersection. Un opérateur topologique peut être un opérateur de calcul (calcul de distance entre deux objets spatiaux, par exemple) ou un opérateur de vérification (vérifier si deux objets spatiaux ont une surface mitoyenne par exemple). Un objet spatial est matérialisé par un ensemble de primitives géométriques. Le résultat d'une opération topologique peut être géométrique (calculer, par exemple, la géométrie de la surface mitoyenne entre deux objets spatiaux), numérique (calculer, par exemple, la distance entre deux objets spatiaux) ou booléenne (vérifier, par exemple, si un objet spatial est à l'intérieur ou non d'un autre).

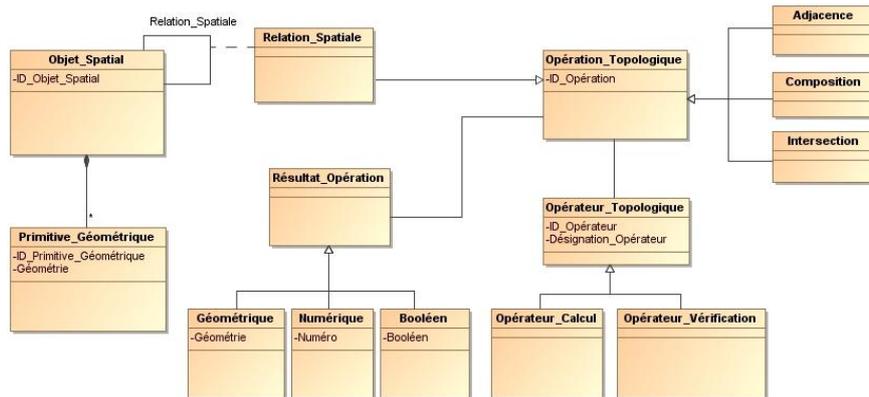


Fig8.11. Méta-modèle d'une opération topologique

A partir de ce méta-modèle, nous pouvons interpréter l'algorithme, que nous appelons "*Algorithme topologique*", permettant de calculer une opération topologique entre deux objets spatiaux (Fig8.12).

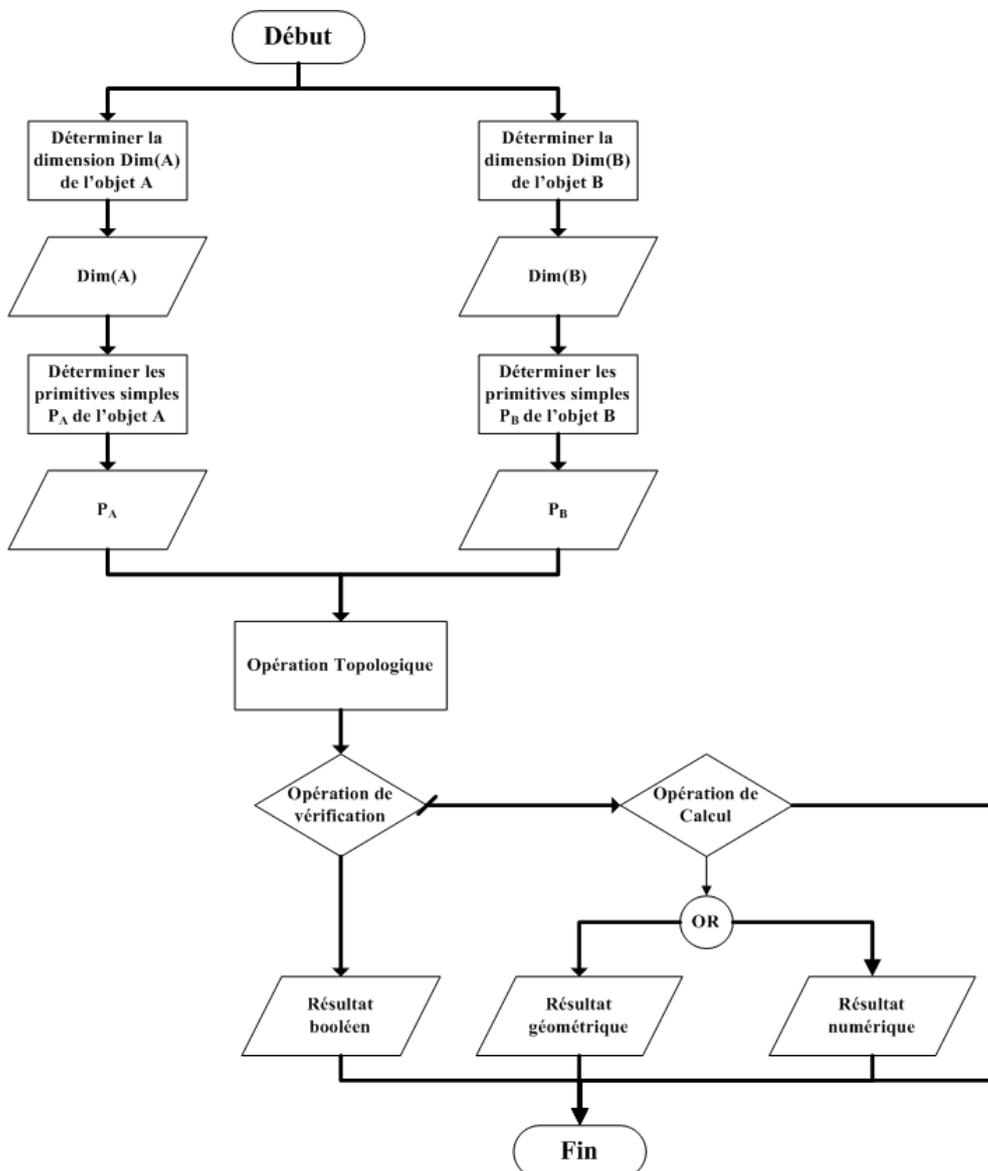
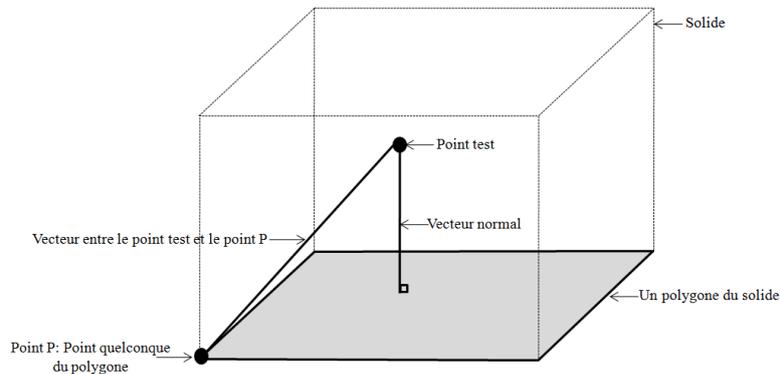


Fig8.12. Organigramme de l'algorithme topologique

L'algorithme, que nous appelons "*Algorithme Test\_point\_dans\_solide*" permettant de vérifier si un point est à l'intérieur d'un solide est utilisé dans l'algorithme topologique et en particulier dans la

relation « Dans ». L'idée de base est de vérifier si l'un des sommets d'un solide est à l'intérieur, sur les frontières ou à l'extérieur d'un autre pour tester les relations entre tels solides. Par exemple, pour tester si un solide B est complètement inclus dans un solide A, il faut que tous les sommets du solide B soient à l'intérieur du solide A. L'idée de la méthode est expliquée dans la figure (Fig8.13). Pour chaque polygone du solide on calcule deux vecteurs : un vecteur normal au polygone et passant par le point test et un vecteur passant par le point test et un point quelconque du polygone (un sommet du polygone, par exemple). On calcule le produit scalaire des deux vecteurs. Si les produits scalaires calculés pour tous les polygones du solide ont le même signe (sont tous positifs, tous négatifs ou tous nuls) alors le point est à l'intérieur du solide, et il est à l'extérieur le cas contraire.



**Fig8.13.** Schéma illustratif de la méthode permettant de vérifier si un point est l'intérieur d'un solide

Nous nous basons principalement sur les sommets des solides pour déterminer leurs relations de voisinage. L'algorithme permettant de tester si un point est à l'intérieur ou non d'un solide est expliqué dans la figure (Fig8.14).

```

Paramètres
Solide : Tableau numérique //Liste d'identifiants polygones d'un solide
Point Test : Tableau numérique //Coordonnées X, Y et Z du point à tester
Variables
Est_Intérieur : Variable booléenne // Pour tester si point est à l'intérieur du solide
Polygone1 : Matrice de réels // Matrice de coordonnées du premier polygone d
Vect_Normall : Matrice de réels// Matrice de coordonnées du vecteur normal au Polygone0
Vect_Temp1 : Matrice de réels// Matrice de coordonnées d'un vecteur temporaire
Scalaire1 : double // Le produit scalaire des vecteurs Vect Normal0 et Vect Temp0
Signe Num1 : Entier // Signe Num = 0 si num = 1, Signe Num > 0 si num = 0 et Signe Num = -1
si num < 0 : signe du premier polygone
Polygone : Matrice de réels // Matrice de coordonnées d'un polygone quelconque
Vect Normal : Matrice de réels// Matrice de coordonnées du vecteur normal à un polygone
quelconque
Vect Temp : Matrice de réels// Matrice de coordonnées d'un vecteur temporaire quelconque
Scalaire : double // Le produit scalaire de deux vecteurs quelconques
Signe_Num : Entier // Signe_Num1 = 0: si num = 0, 1 si num>0, -1 si num<0 // Le signe d'un
scalaire quelconque
Début
    Est Intérieur = vrai
    Signe Num1 = 0
    //1) Traiter le premier polygone du solide
        //1.1) Obtenir le premier polygone du solide
            Polygone1 = ObtenirPolygone(Solide,1)
        //1.2) Calculer le vecteur normal du Polygone1
            Vect Normal1 = Vecteur Normal(Polygone1);
        //1.3) Calculer un vecteur entre le Point_Test et le premier point du Polygone1
            Vect Temp1= Construire_Vect(TestPoint, Premier point du polygone1);
        //1.4) Calculer le produit scalaire entre Vect_Normall et Vect_Temp1
            Scalaire1 = Produit Scalaire( Vect Normal1 , Vect Temp1);
            Signe Num0 = Signe(Scalaire0);
    //2) Traiter les autres polygones du solide Est Intérieur = vrai
        POUR i allant de 2 jusqu'à le nombre de polygones du solide ET Est_Intérieur =
vrai
        FAIRE
            //2.1) Obtenir le premier polygone à la position i du solide
                Polygone = ObtenirPolygone(Solide,i)
            //2.2) Calculer le vecteur normal du polygone
                Vect_Normal = Vecteur_Normal(Polygone);
            //2.3) Calculer un vecteur entre le Point_Test et le premier point du

```

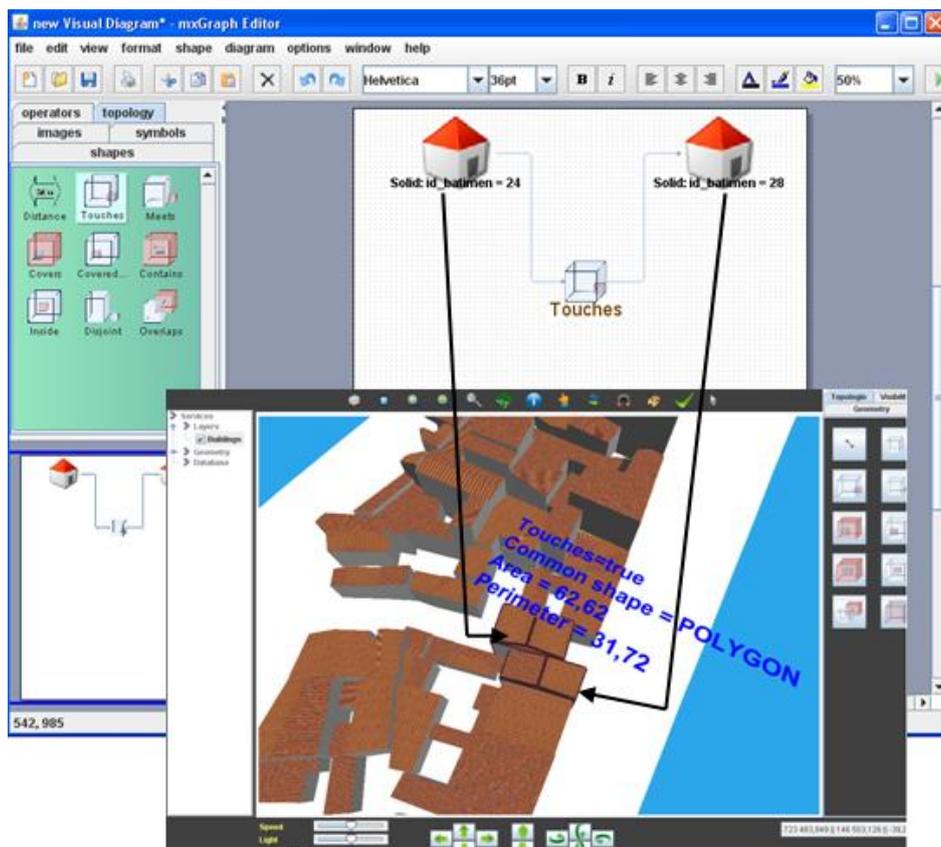
```

Polygone
Vect_Temp= Construire_Vect(TestPoint, Premier point du polygone);
//2.4) Calculer le produit scalaire entre Vect_Normal et Vect_Temp
Scalaire = Produit_Scalaire( Vect_Normal , Vect_Temp);
Signe Num = Signe(Scalaire);
SI (Signe_Num != Signe_Num1 && Signe_Num != 0)
    Est Intérieur = faux
FIN SI
FIN POUR
Fin

```

*Fig8.14.* Algorithme Test\_point\_dans\_solide

Les différentes opérations topologiques sont traitées cas par cas à travers des fonctions matérialisées par des classes Java. La figure (*Fig8.15*) illustre un exemple de requête topologique permettant de vérifier si deux les deux bâtiments numéro 24 et 28 se touchent ou non.



*Fig8.15.* Exemple d'une opération topologique sur deux bâtiments 3D

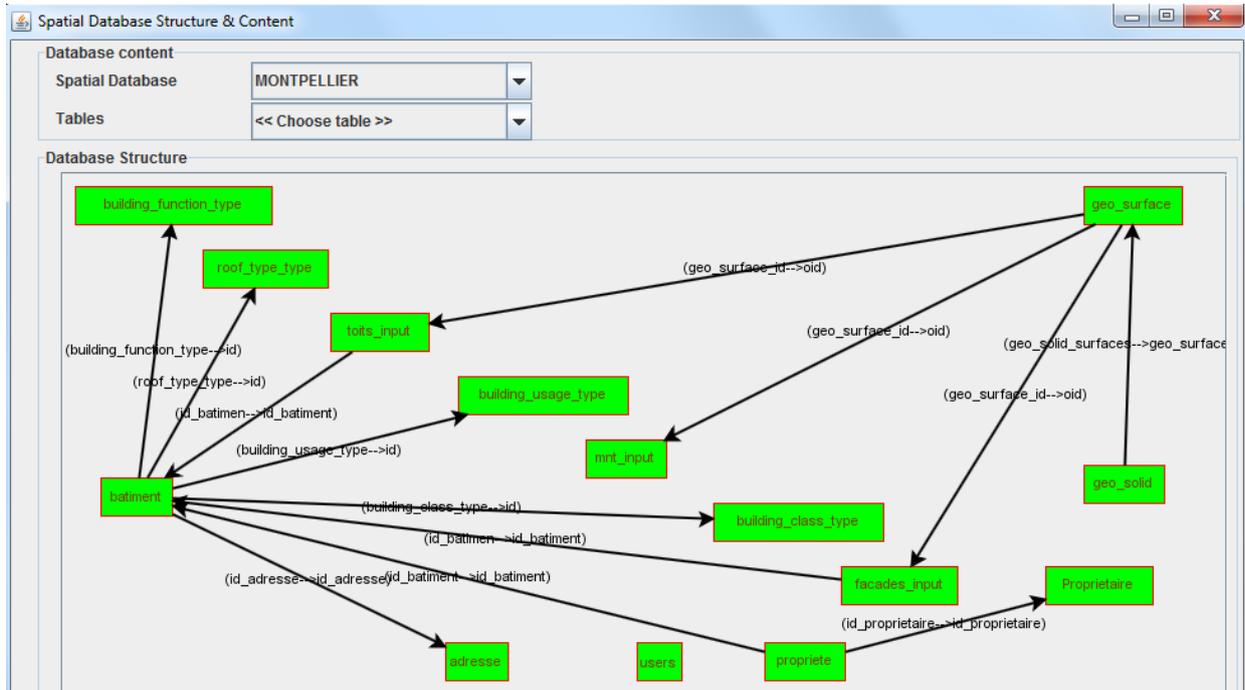
A l'instar de cette opération topologique, nous obtiendrons un résultat booléen (se touchent = vrai), un résultat géométrique (la surface mitoyenne est un polygone) et un résultat numérique (la surface du polygone = 62.62 m<sup>2</sup> et son périmètre = 31.72 m).

## 2.2. REQUETES ATTRIBUTAIRES

Certes les données spatiales constituent les données de base, voire la raison d'être, de tout SIG. Elles décrivent la localisation dans l'espace des objets spatiaux, leurs géométries, leurs apparences, les relations topologiques entre les objets, etc. Cependant, les données spatiales deviennent plus importantes si elles sont associées à des données attributaires. En effet, la force réelle de tout SIG réside dans sa capacité de décrire des objets spatiaux à travers leur localisation dans l'espace ainsi que les données sémantiques associées. La diversification de données dans un SIG offre plus de possibilités en termes d'interrogation des BDS pour en extraire des objets spatiaux sophistiqués.

Avant de mettre en place n'importe quelle requête pour interroger la BDS, l'utilisateur désire parfois avoir une idée claire sur la structure (les différentes tables et vues ainsi que les relations entre elles) et

le contenu de la BDS (enregistrements de chaque table). Cela aide l'utilisateur à comprendre la BDS à interroger et donc à mieux exprimer ses attentes des requêtes spatiales. Nous avons développé une fonctionnalité particulière permettant à l'utilisateur d'afficher à tout moment la structure ainsi que le contenu d'une BDS donnée. L'idée consiste à afficher le modèle conceptuel de la BDS composé de tables et vues en tant que des cellules d'un graph et les liens entre elles (clés étrangères) en tant que des flèches indiquant les attributs composant les clés étrangères et les attributs référencés. Une illustration de cette fonctionnalité est décrite dans la figure (Fig8.16) et elle consiste à afficher le modèle conceptuel de la BDS exemple appelée "Bâtiments" décrite dans le chapitre précédent.



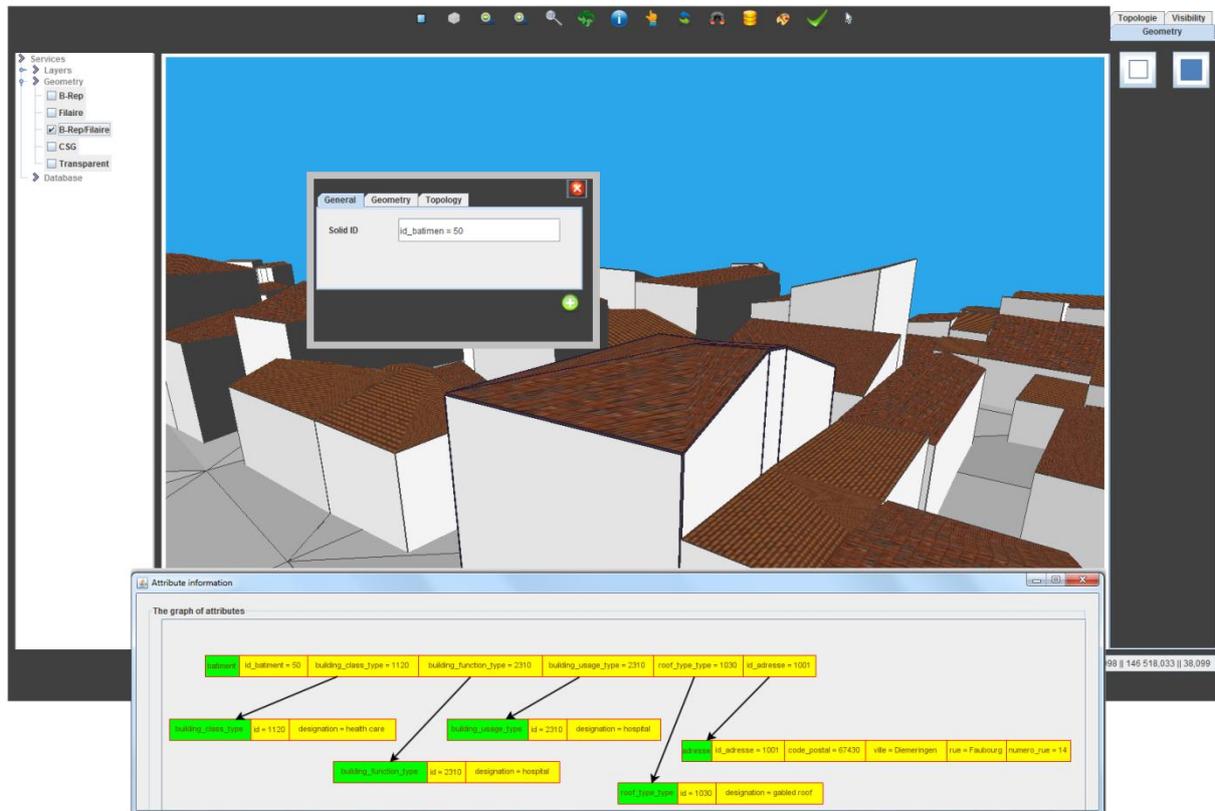
**Fig8.16.** Affichage graphique du modèle conceptuel d'une BDS (exemple de la BDS "Bâtiments")

Dans le graphe de la figure (Fig8.16) ci-dessus, le modèle conceptuel de la BDS est décrit par un graphe dans lequel les tables sont représentées par des rectangles et les jointures par des liens (flèches partant de la table contenant la clé étrangère vers la table contenant la clé référencée). Pour chaque lien entre deux tables, les attributs de jointures sont également renseignés.

L'utilisateur peut s'intéresser aussi au contenu et dépendances (jointures) d'une table particulière dans la BDS. Pour ce faire, l'application SIG permet aussi de sélectionner une table de la BDS pour afficher automatiquement dans un tableau tous ses enregistrements et indiquer, dans le graphe du modèle conceptuel de la BDS, avec des couleurs différentes toutes les tables reliées à la table sélectionnées. Une illustration de cette fonctionnalité est décrite dans la figure (Fig8.17) et permettant de montrer le contenu ainsi que les dépendances de la table "Bâtiment".



deuxième stade, l'utilisateur affiche des informations plus détaillées provenant de la BDS à propos de l'objet sélectionné. Le principe de l'idée est de coupler dans le même affichage des informations tabulaires et graphiques. Le système cherche dans la BDS tous les enregistrements des tables comportant des données à propos de l'objet sélectionné.



**Fig8.18.** Affichage d'informations attributaires concernant un objet spatial sélectionné par l'utilisateur

Par exemple dans la figure (Fig8.18) ci-dessus, l'utilisateur a sélectionné un bâtiment (numéro 50) et demande l'affichage des informations détaillées à propos de ce bâtiment. Comme les bâtiments dans la BDS sont définis par les tables "Building\_Class\_Type", « Building\_Function\_Type", "Adresse", etc., les informations attributaires porteront sur tous les enregistrements dans ces tables et concernant le bâtiment numéro 50. Le graphe contient les noms des tables (cellules vertes) ainsi que les noms et les valeurs des attributs correspondants (cellules jaunes) et les liens entre les tables

Outre la sélection graphique permettant d'afficher des informations géométriques et attributaires d'un objet spatial sélectionné directement dans un modèle 3D, l'application SIG offre la possibilité d'une recherche avancée d'objets spatiaux à l'aide d'un assistant pour l'écriture de requêtes textuelles pouvant porter à la fois sur les propriétés variées des objets spatiaux.

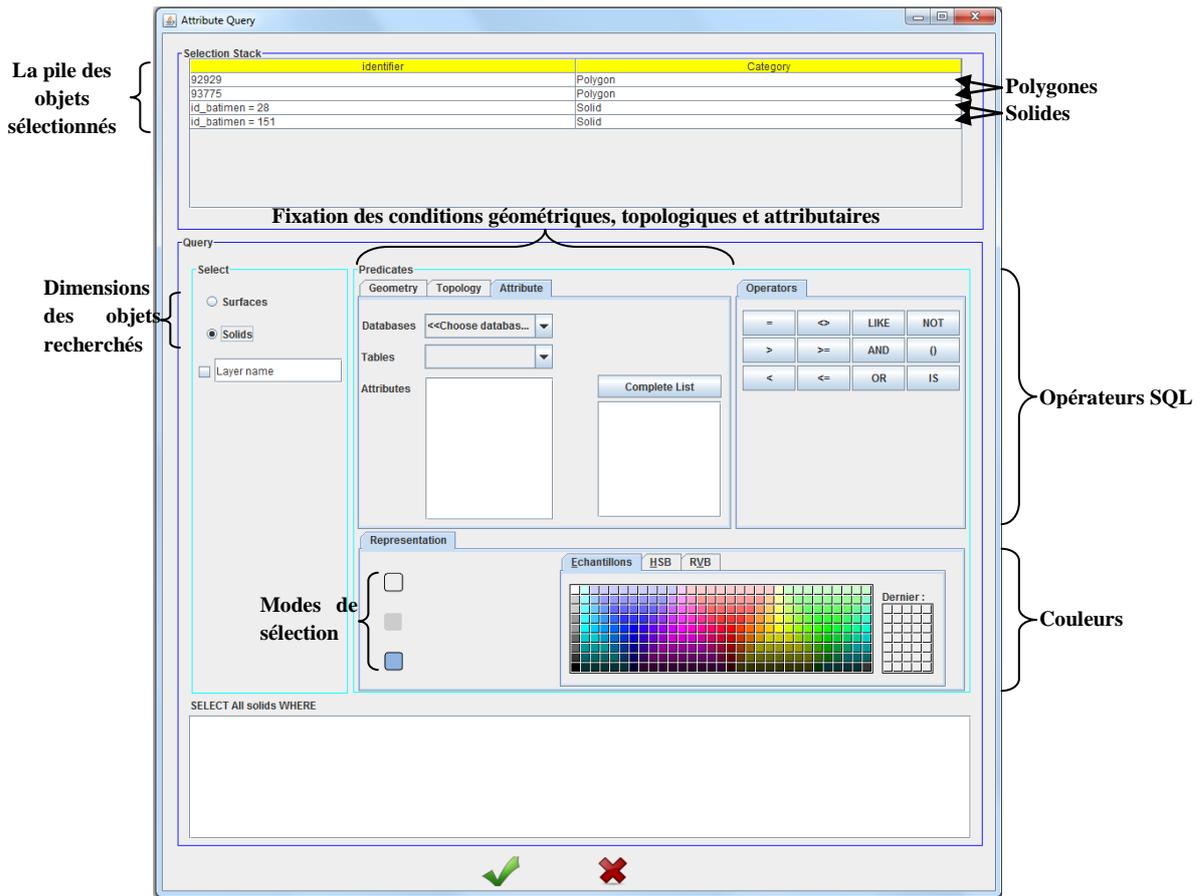


Fig8.19. L'interface permettant de créer une requête pour chercher des objets spatiaux

Nous expliquons ci-dessous les différentes parties de l'interface graphique permettant d'écrire une requête pouvant porter sur la géométrie, la topologie ou les caractéristiques attributaires des objets spatiaux.

- **Partie géométrique**

Dans cette partie de la requête, l'utilisateur fixe ses choix géométriques sur un objet géométrique particulier. Le SIG propose deux opérateurs géométriques principaux : le calcul de la surface et le calcul du périmètre d'un objet spatial particulier. Par exemple, le calcul de surface peut être appliqué sur des surfaces (par exemple, façades des bâtiments) et sur des solides (par exemple, bâtiments) pour le calcul des surfaces de tous les polygones d'un solide. On applique le même principe pour le calcul du périmètre.

On peut alors écrire une requête du type :

```
« Déterminer tous les bâtiments ayant au moins une façade dont la surface est supérieur à 30 m2 »
```

Ces deux opérateurs sont simples à développer, mais insuffisants car l'utilisateur peut avoir d'autres besoins en termes de calculs géométriques. Et comme parmi l'un des buts principaux de ce SIG est d'offrir à l'utilisateur la possibilité de personnaliser ses traitements, nous proposons une structure permettant à l'utilisateur de définir les opérateurs géométriques de son choix tel que c'est décrit dans la figure (Fig8.20).

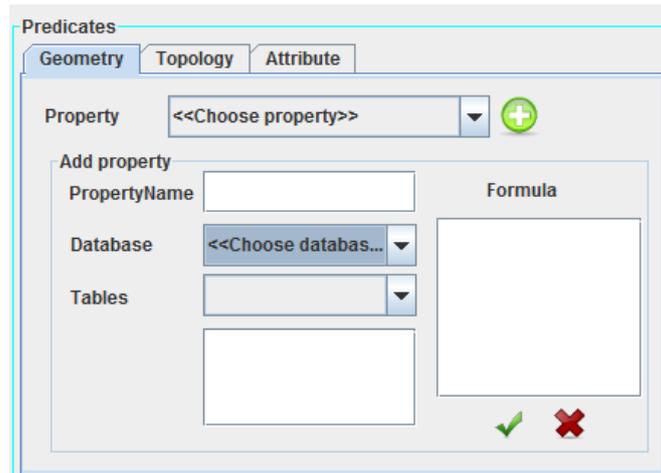


Fig8.20. L'outil graphique permettant de créer un opérateur géométrique

Le principe de création d'un opérateur géométrique se base principalement sur un ou plusieurs attributs des tables de la BDS et leur combinaison en utilisant les opérateurs mathématiques (addition, soustraction, multiplication et division, etc.). En effet, le SIG permet à l'utilisateur de choisir une ou plusieurs tables de la BDS et d'en choisir un ou plusieurs attributs et les combiner avec des opérateurs mathématiques pour déterminer une formule géométrique.

Par exemple, admettons que nous avons l'attribut "*Hauteur\_Façade*" de la table "*Façades*", l'attribut "*Hauteur\_Toit*" de la table "*Toits*" et supposons que la hauteur moyenne d'un étage d'un bâtiment est 3 mètres, l'utilisateur peut créer une propriété appelée "*Nombre\_Etages*" permettant de calculer le nombre d'étages d'un bâtiment et matérialisée par la formule suivante :

```
Nombre_Etages = (Façades.Hauteur_Façade+ Toits.Hauteur_Toit)/3
```

L'utilisateur peut utiliser cette propriété personnalisée pour calculer les nombres d'étages des bâtiments.

- **Partie Topologique**

Dans cette partie, nous reprenons les opérateurs topologiques que nous avons décrits dans la section 2.1 ci-dessus. Il suffit de choisir l'opérateur ainsi que les objets spatiaux correspondant (des surfaces et des volumes). L'application propose un symbole graphique pour chaque opérateur topologique, ce qui facilitera leur choix.

Par exemple, on peut écrire la requête suivante :

```
« Déterminer tous les bâtiments qui touchent le bâtiment numéro 5100 »
```

- **Partie Attributaire**

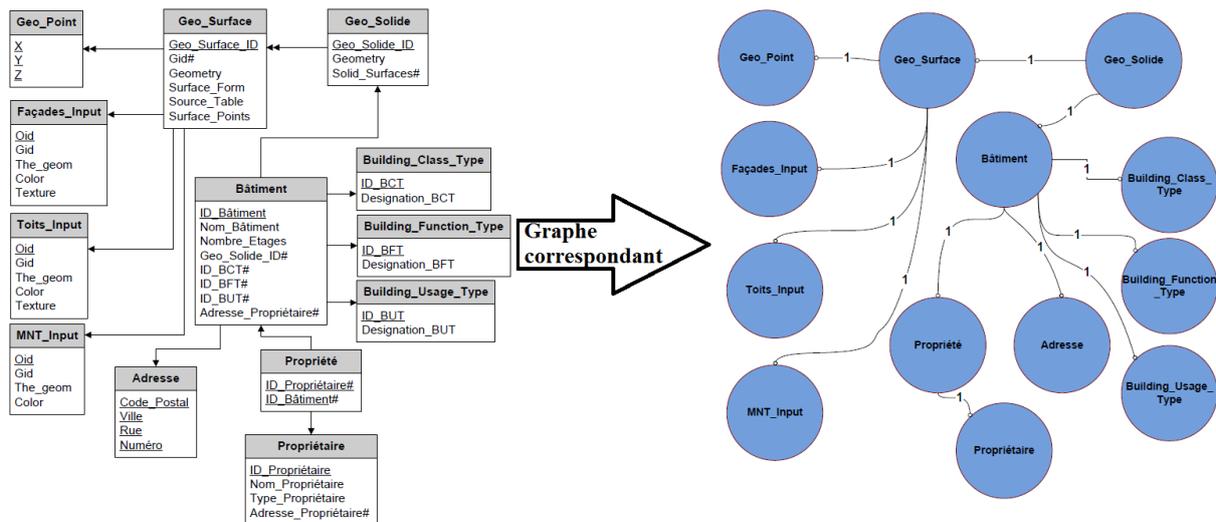
Dans cette partie, la requête portera sur les propriétés attributaires des objets spatiaux. En effet, l'interrogation va dépasser les propriétés géométriques ainsi que les relations de voisinage des objets spatiaux pour s'intéresser aux caractéristiques sémantiques des objets spatiaux, pouvant être facilement perçues par les utilisateurs du SIG. Par exemple, on peut écrire une requête comme :

```
« Déterminer tous les bâtiments commerciaux appartenant au propriétaire numéro XXX »
```

- **Combinaison des trois parties**

Ces trois parties peuvent être combinées dans la même requête pour offrir à l'utilisateur plus de possibilités d'interrogation. La combinaison se fait en utilisant les opérateurs SQL "AND" et "OR". Les jointures entre les tables interrogées sont interprétées automatiquement grâce à un algorithme appelé « *Algorithme des jointures automatiques* ».

L'idée de l'algorithme est de considérer le schéma logique de la BDS comme un graphe connexe non orienté où les tables constitueront les nœuds et les relations entre les tables constitueront les arcs, ayant un poids égal à 1, dans le graphe. Pour déterminer la relation de jointure entre deux tables quelconques de la BDS, on cherche le chemin le plus court entre les deux nœuds du graphe. La figure (Fig8.21) illustre cet algorithme sur le schéma de la BDS "Bâtiments", décrite dans le chapitre précédent, que nous prenons comme exemple.



**Fig8.21.** Schéma illustratif de la traduction d'un schéma de base de données en graphe

Pour chercher le chemin le plus court dans le graphe correspondant au schéma de la BDS, nous appliquons l'algorithme de Dijkstra. La recherche de jointures entre les tables se fait alors d'une façon automatique et transparente à l'utilisateur. La création du graphe correspondant à une BDS se base sur une requête proposée par PostgreSQL et permettant de retourner la liste des contraintes de type « *clé étrangère* ». Un exemple d'une requête de recherche de jointures entre les tables d'une base de données PostgreSQL est :

```
«SELECT Distinct(C.conrelid) AS Identifiant_Table_Source, /* L'identifiant de la
table contenant la clé étrangère*/
T1.relname AS Nom_Table_Source, /* Le nom de la table contenant la
clé étrangère*/
C.confrelid AS Identifiant_Table_Cible, /* L'identifiant de la
table référencée*/
T2.relname AS Nom_Table_Cible, /* Le nom de la table référencée*/
C.conkey AS Colonnes_Sources, /* La liste des colonnes formant la
clé étrangère*/
C.confkey AS Colonnes_Cibles /* La liste des colonnes référencés*/
FROM pg_constraint C, pg_class T1, pg_class T2
WHERE contype LIKE 'f'
AND T1.relfilenode = C.conrelid
AND T2.relfilenode = C.confrelid»
```

Par exemple, si on cherche toutes les façades des bâtiments appartenant à un propriétaire XXX, le parcours du graphe va retourner le chemin de plus court comme expliqué dans la figure (Fig8.22).

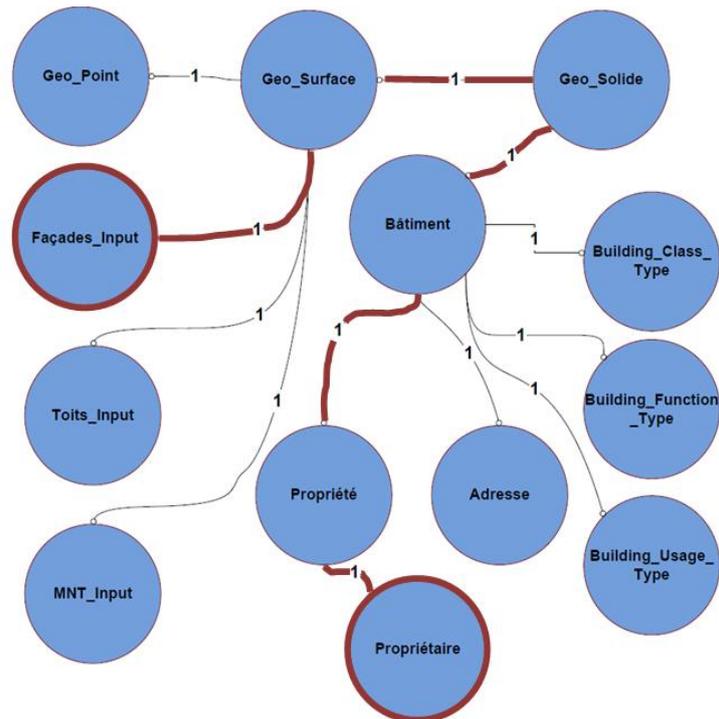


Fig8.22. Exemple illustratif du calcul du plus court chemin

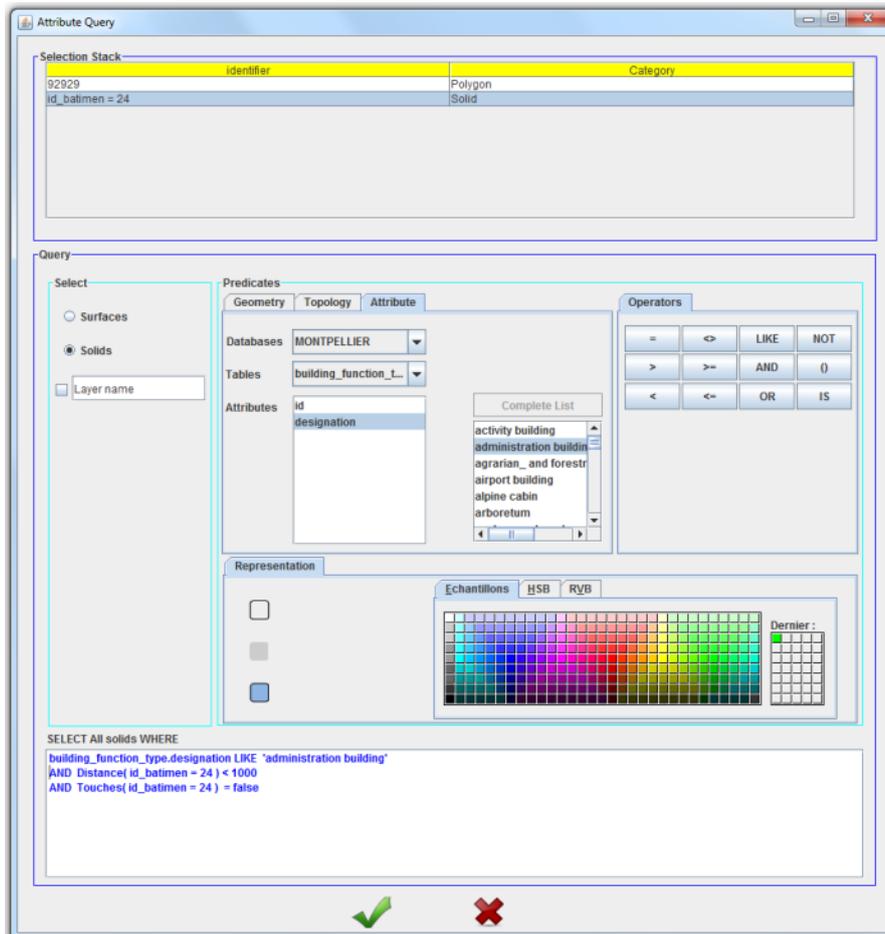
A l'issue de ce chemin, la requête SQL calculée automatiquement par le système est :

```
« SELECT Geo_Surface.*
FROM Geo_Surface, Geo_Solide, Facades_Input, Batiment, Propriete, Proprietaire
WHERE   Facades_Input.gid = Geo_Surface.gid
AND     Geo_Surface.geo_surface_id = ANY (solid surfaces) LIMIT 1
AND     Batiment.geo_solide_ID = Geo_Solide.geo_solide_ID
AND     Batiment.id_batiment = Propriete.id_batiment
AND     Propriete.id_proprietaire = Proprietaire.id_proprietaire
AND     Proprietaire.id_proprietaire = XXX
```

Quand l'utilisateur a fixé ses choix géométriques, topologiques et attributaires, il peut personnaliser le mode de représentation des objets spatiaux recherchés en termes de :

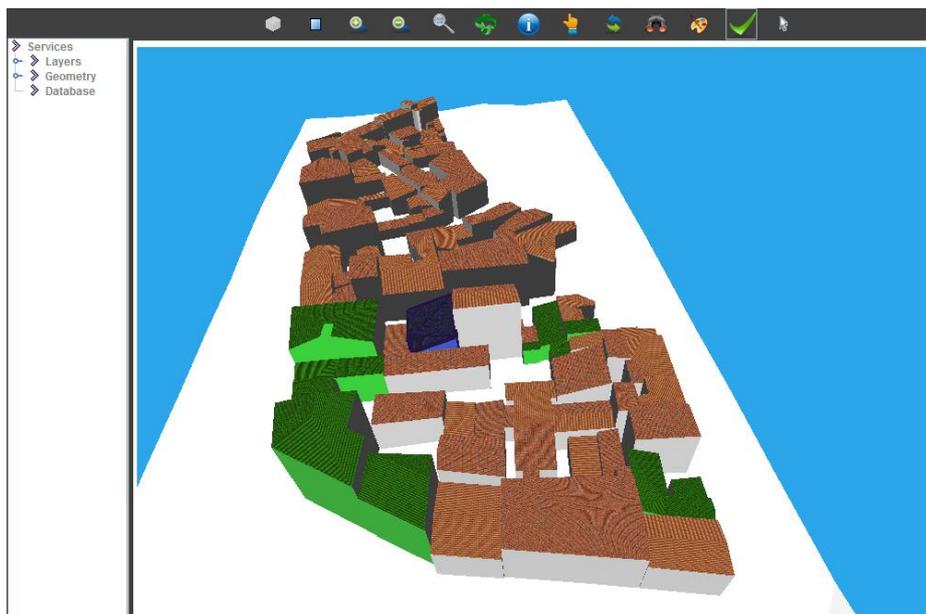
- Dimension des objets spatiaux : Il s'agit d'identifier la dimension d'objets spatiaux recherchés : des polygones ou des solides. Nous considérons que le choix de la dimension d'objets spatiaux est très important car un solide peut contenir des surfaces ayant des caractéristiques différentes. Par exemple, supposons que nous avons une requête qui cherche les "toits plats" dans un modèle 3D de bâtiments. Si on choisit une recherche par surfaces, on n'aura comme résultat que les surfaces correspondant aux toits des bâtiments. Alors que si on choisit une recherche par solide, on aura comme résultat tous les solides correspondant à des bâtiments ayant des toits plats.
- Le nom d'une couche spatiale : Il s'agit d'indiquer un nom pour la couche spatiale si l'utilisateur désire regrouper les objets spatiaux recherchés dans une couche spatiale indépendante.
- Le mode de sélection : Il s'agit d'indiquer le mode de sélection des objets spatiaux recherchés : Sélectionner que le contour des polygones, sélectionner l'intérieur des polygones ou sélectionner le contour et l'intérieur des polygones.
- Apparences : Il s'agit de choisir une couleur pour la représentation des objets spatiaux recherchés.
- Les couches spatiales : L'interrogation ne portera que sur les couches spatiales actives, c'est-à-dire les couches spatiales sélectionnées lors du processus d'interrogation.

Ci-dessous nous pouvons voir un exemple de requête qui permet de chercher : *Tous les bâtiments de type "administratif" et qui se trouvent à une "distance" inférieure à 1 km du bâtiment numéro 24 sélectionné par l'utilisateur et qui ne "touchent" pas ce bâtiment.*



**Fig8.23.** Exemple d'une requête portant sur des propriétés géométriques, topologiques et attributaires

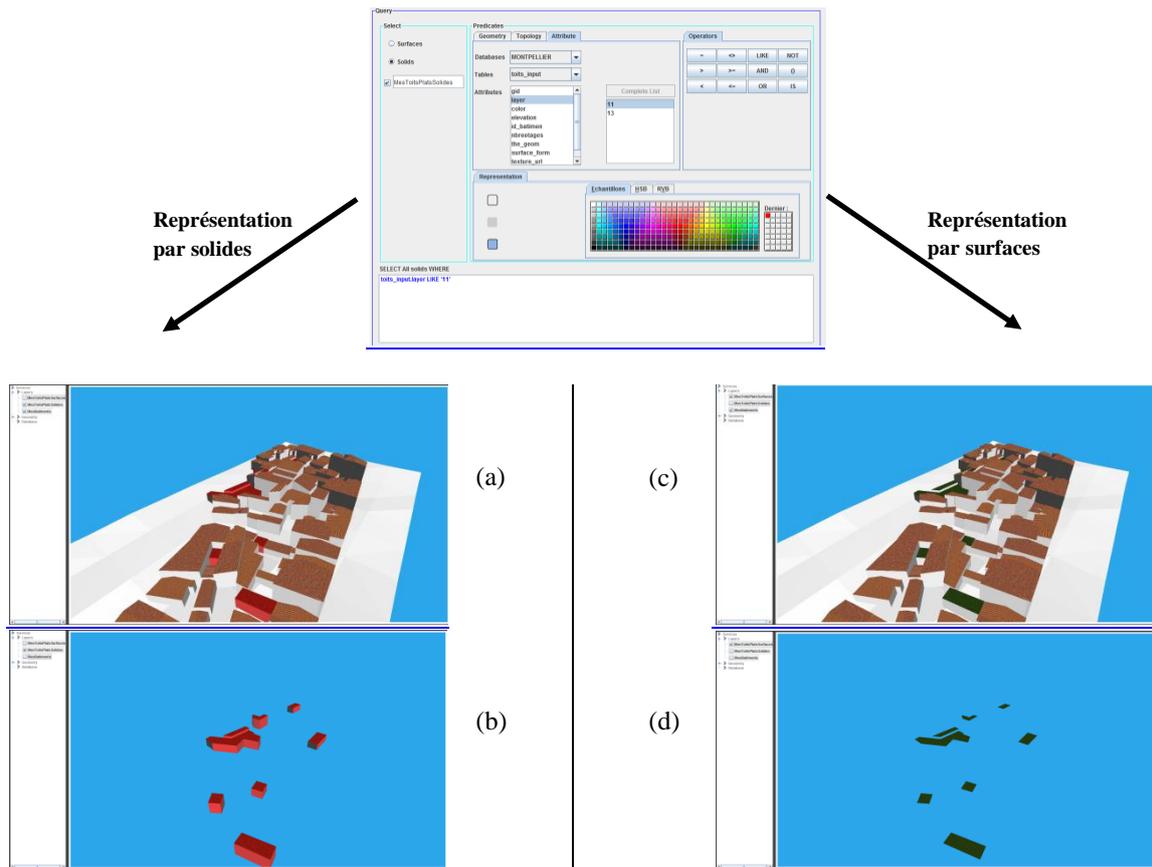
Le résultat de la requête dans le modèle 3D est décrit dans la figure (Fig8.24).



**Fig8.24.** Le résultat de la requête décrite dans la figure (Fig8.23)

Le bâtiment qui apparaît en bleu est le bâtiment numéro 24 sélectionné par l'utilisateur et les bâtiments qui apparaissent en vert constituent les bâtiments répondant au prédicat défini par l'utilisateur. Le SIG permet deux principaux types de représentation à travers cette application d'interrogation attributaire : représentation par surface et représentation par solide. En effet, l'utilisateur peut écrire une requête pour chercher des surfaces répondant à un certain nombre de critères.

Les alternatives d'affichage sont : i) Afficher que les surfaces recherchées; ii) Afficher les solides auxquels appartiennent les surfaces recherchées. Par exemple, la figure (Fig8.25) illustre le résultat d'une requête qui permet de chercher les toits plats (attribut *layer* = 11 de la table *toits\_input*). L'utilisateur peut ainsi afficher que les surfaces appartenant aux toits plats ou tous les bâtiments ayant des toits plats.



**Fig8.25.** Exemple d'une requête attributaires permettant de retourner un résultat sous forme de solides (droite) et sous forme de surfaces (gauche).

Les deux images du haut (a) et (c) de la figure (Fig8.25) représentent, respectivement, le résultat de la requête dans le modèle 3D original selon les deux alternatives d'affichage. Tandis que les deux images du bas (b) et (d) de la même figure montrent, respectivement, les résultats de la requête dans des modèles 3D séparés. Le but est de pouvoir, à tout moment, distinguer le résultat d'une requête dans et en dehors du modèle 3D original sur lequel on a exécuté cette requête. Multiplier les alternatives d'affichage peut aider l'utilisateur à bien percevoir le contenu ainsi que les résultats de requêtes sur un modèle 3D donné.

La dernière application que nous allons décrire consiste à représenter les objets spatiaux d'un modèle 3D selon le contenu d'un ou de plusieurs attributs des tables de la BDS. L'idée de l'application est d'aider l'utilisateur à étudier le contenu d'un modèle 3D selon les valeurs d'un attribut ou de la combinaison de plusieurs attributs de la BDS. Après le choix d'un ou de plusieurs attributs, l'utilisateur choisit le nombre de plages de valeurs à représenter. Le système affiche ainsi à l'aide d'une couleur pour chaque plage de valeurs les objets spatiaux correspondant.

Pour illustrer le principe de cette application, nous avons ajouté des données à la BDS "Bâtiments" (décrite dans le chapitre 6 "Représentation des données spatiales dans le SIG") pour calculer l'émission annuelle du CO<sub>2</sub> (Dioxyde de Carbone) due à l'utilisation des chauffages dans les bâtiments. Pour ce faire, nous avons créé une table "type\_chauffage" indiquant la quantité (en kilogrammes) d'émission de CO<sub>2</sub> pour chaque catégorie de chauffage (chauffage électrique, chauffage au gaz, etc.). Comme dans un bâtiment il peut exister un ou plusieurs chauffages, l'émission du CO<sub>2</sub> de ce bâtiment sera calculée en fonction du nombre de ces chauffages. La figure (Fig8.26) reprend quelques exemples d'enregistrements des tables "batiment" et "type\_chauffage" reliées respectivement par les attributs "categorie\_chauffage" et "categorie" pour expliquer davantage le contexte de la BDS.

id_batiment [PK] numeric	building_class numeric	building_func numeric	building_usage numeric	roof_type_ty numeric	id_adresse oid	categorie_chauffage numeric	nombre_chauffages numeric	oid	categorie [PK] numeric	description character varying(50)	emission_co2 numeric
1	1	1010	1670	1670	1030	1	8	1	981272	chauffage électrique	25
2	2	1000	1060	1060	1030	1	12	2	981273	chauffage à gaz	40
3	3	1110	1320	1320	1030	2	3				

(a)

(b)

Fig8.26. Exemples de quelques enregistrements des tables (a) : "batiment", (b) : "type\_chauffage"

Le calcul de l'émission du CO<sub>2</sub> pour chaque bâtiment se fait en fonction de l'attribut "nombre\_chauffages" de la table "batiment" et de l'attribut "emission\_CO2" de la table "type\_chauffage". La formule permettant de calculer l'émission du CO<sub>2</sub> est la suivante :

```
batiment.nombre_chauffages * type_chauffage.emission_CO2
```

L'interface graphique permettant ce type de requête est décrite dans la figure (Fig8.27).

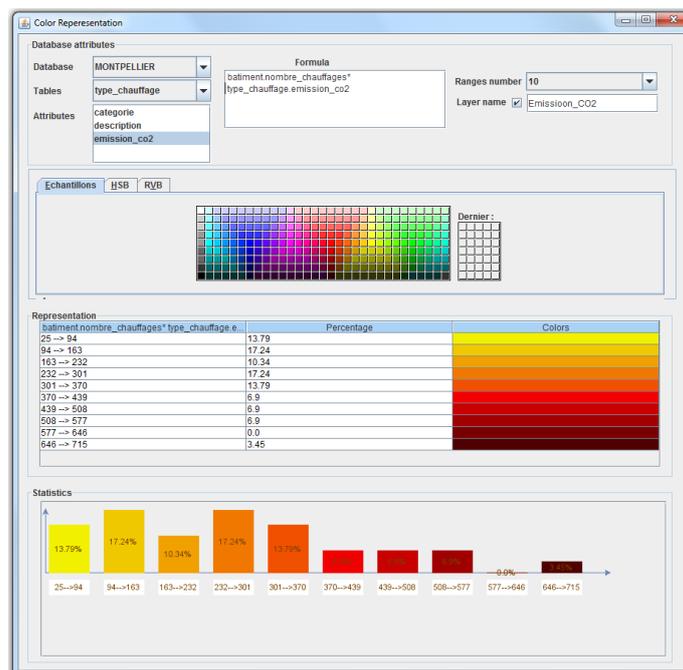
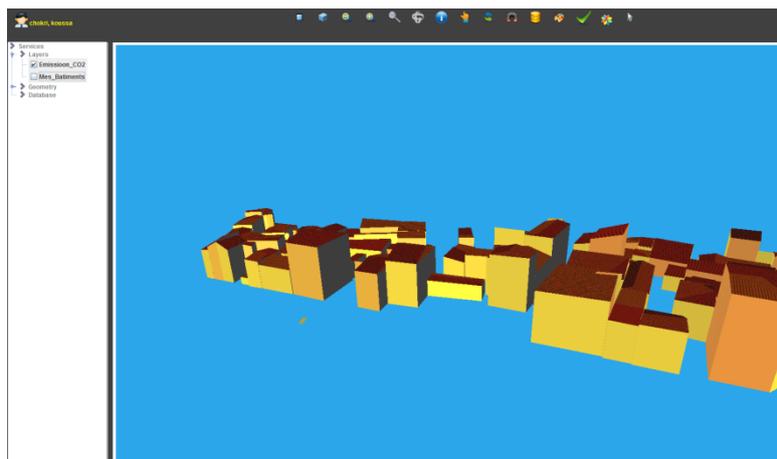


Fig8.27. Interface graphique permettant de représenter la BDS selon son contenu

La figure (Fig8.27) montre la formule utilisée pour calculer l'émission du CO<sub>2</sub> et sa représentation en 10 plages de valeurs différentes. En bas de la figure, nous avons développé une fonctionnalité permettant d'afficher un histogramme simplifié pour comparer les pourcentages des bâtiments pour chaque plage de valeurs. Le résultat de cette requête est décrit dans la figure (Fig8.28).

Il est possible que l'utilisateur change les limites des plages des valeurs, leurs nombres, les couleurs pour représenter les objets spatiaux pour chaque plage de valeurs, etc. Le but est d'offrir une certaine liberté et flexibilité pour l'utilisateur permettant ainsi des représentations personnalisées selon ses besoins.



**Fig8.28.** Résultat de la requête représentée dans la figure (Fig8.27)

Nous avons essayé avec cette application de créer un outil simple pour l'interprétation d'un ensemble de représentations du modèle 3D en fonction du contenu de la BDS. Cette application permet un accès dynamique à tous les attributs de la BDS pour en formuler une combinaison à l'aide des opérateurs mathématiques (+, -, \*, /) et afficher les objets spatiaux selon différentes couleurs. Une requête de ce type est calculée dynamiquement par le système et les jointures entre les tables sont interprétées de manière transparente à l'utilisateur du SIG 3D. En fin, le résultat d'une requête peut être stocké dans la BDS en tant qu'une nouvelle couche spatiale pour un usage futur.

Toutes les fonctionnalités proposées dans ce chapitre ne constituent qu'un premier pas vers la création d'un SIG 3D complet. Elles ont exprimé des points communs avec les outils SIG existants. Mais elles ont aussi révélé des nouvelles fonctionnalités traduisant nos idées et notre perception dans le domaine des SIG. La comparaison entre les outils existants et nos travaux constitue l'objet de la partie suivante de ce chapitre.

### 3. POSITIONNEMENT DU TRAVAIL PAR RAPPORT AUX OUTILS EXISTANTS

Chaque outil SIG propose un ensemble d'outils pour la manipulation des données spatiales. Les logiciels les plus répandus dans le domaine des SIG sont ArcGIS et MapInfo en tant qu'outils commerciaux et QGIS et GRASS en tant qu'outils libres.

ArcGIS d'ESRI propose, avec sa boîte à outils appelée "ArcGIS toolbox", un ensemble de fonctionnalités organisées en sous-groupes tels que "3D Analyst toolbox", "Analysis toolbox", "Data management tools", etc.

MapInfo propose de son côté toute une gamme d'outils pour représenter sur des cartes des données comportant des informations géographiques. En effet, pour l'exploitation de ses différents outils, MapInfo fournit un ensemble de modules ou extensions tels que "MapBasic", "MapInfo ProViewer", "AutoCart", "GooMap", "Mapinfo Geomarketing", etc.

QGIS propose une boîte à outils pour visualiser, gérer, éditer, analyser des données spatiales de différents formats et composer des cartes imprimables.

Enfin GRASS est utilisé pour la gestion et l'analyse des données spatiales, le traitement d'images, la production de cartes, la modélisation et la visualisation spatiale, etc.

⇒ Concernant l'accès aux données, ArcGIS et MapInfo permettent d'accéder à de multiples types de données telles que des données CAO (*DWG*, *DXF* et *DGN*), des données ESRI, etc. en mode natif, c'est-à-dire accéder et travailler directement avec de telles données sans devoir les convertir au préalable.

Dans notre proposition de SIG, les données doivent être converties vers un format de base de données et intégrées dans PostgreSQL/PostGIS pour être exploitées. Ceci constitue encore une limite importante de notre système.

⇒ Concernant les traitements des couches spatiales, la plupart des SIG utilisent un système de couches. Pour leur combinaison, ArcGIS utilise la notion de "GeoProcessing" qui consiste à créer des nouvelles informations en appliquant des opérations sur des données existantes. Sa boîte à outils "Analysis tools" permet ainsi de combiner deux couches spatiales A et B en utilisant deux opérations (intersection et différence) et la boîte à outils "Data management tools" permettant l'opération d'union pour obtenir une nouvelle couche C.

GRASS permet, de son côté, des opérations booléennes grâce à son outil "Overlay operations" permettant les opérations : Union, intersection, différence et XOR.

Notre SIG propose les trois opérations booléennes : union, intersection et différence. Notre contribution par rapport aux outils existants est la possibilité de mettre en place ce type d'opérations sous la forme d'une interrogation graphique permettant ainsi la combinaison de plusieurs opérations dans une même requête.

⇒ Concernant les traitements topologiques, jusqu'à la version 10.0 d'ArcGIS Desktop, la topologie ne s'applique que sur des points, lignes et surfaces (la version 10.0 n'est pas encore livrée en fin 2010 et la dernière version courante est la 9.3.1.). En effet, dans ArcGIS, la topologie est généralement gérée dans les "GeoDatabases". Elle permet de définir comment les points, lignes et surfaces partagent des géométries communes. Un utilisateur peut créer des classes de topologie dans des GeoDatabases à travers un ensemble de rôles comme, par exemple, le rôle "Must not overlap" pour contrôler si deux géométries se chevauchent ou non.

MapInfo ne propose pas de véritable gestion topologique. Il faut combiner plusieurs outils pour obtenir un semblant de topologie. En effet, MapInfo permet quelques opérations topologiques comme, par exemple, rendre connexes des tronçons qui s'intersectent, créer des polygones à partir d'un réseau filaire, supprimer des trous de polygones, etc.

GRASS gère mieux la topologie depuis qu'il gère intelligemment les géométries partagées. Par exemple, si deux surfaces partagent une ligne commune, cette ligne n'est stockée qu'une seule fois et non pas deux fois, ce qui n'est pas le cas pour les fichiers Shapefile d'ESRI. Cette propriété permet aisément de déterminer des relations de voisinage entre deux géométries.

La contribution de notre SIG réside essentiellement dans la sélection d'objets spatiaux directement dans le modèle 3D et le calcul des opérations topologiques entre tels objets d'une manière entièrement graphique ce qui facilitera énormément la tâche de l'utilisateur. Nous avons effectué aussi une réflexion sur le calcul des relations topologiques entre des objets 3D.

⇒ Concernant, la recherche d'un objet spatial, nous nous inspirons d'ArcGIS en utilisant des requêtes SQL sur la BDS. Mais, en interrogeant plusieurs tables en même temps lors de cette recherche des objets spatiaux pouvant être recherchés à partir d'autres objets, on peut par exemple chercher tous les bâtiments qui touchent un bâtiment sélectionné par l'utilisateur dans le modèle 3D affiché à l'écran.

### CONCLUSION

Après avoir décrit le principe de représentation des données spatiales par le SIG dans le chapitre précédent, nous avons expliqué dans ce chapitre comment les données peuvent être interrogées pour en déduire d'autres données, pour rechercher des objets spatiaux selon des critères fixés par l'utilisateur, etc. Le but majeur de l'application est d'assister l'utilisateur en lui offrant des outils simples de manipulation de couches spatiales.

Pour ce faire, nous proposons une interrogation graphique, simple à mettre en place, afin d'effectuer des opérations booléennes sur les couches spatiales. Le résultat de telles opérations booléennes est matérialisé par des nouvelles couches spatiales pouvant contenir des propriétés des différents modèles

3D interrogés. La combinaison de plusieurs couches spatiales dans une seule couche peut faire apparaître des nouvelles propriétés des données spatiales et dégager de nouvelles possibilités d'interprétations, etc.

La manipulation des couches spatiales est généralement une opération sur des grands lots de données. Mais, l'utilisateur peut avoir besoin d'affiner sa recherche pour chercher des objets spatiaux particuliers et répondant à des critères spécifiques. Pour ce faire, nous proposons, dans cette application, un outil d'interrogation des couches spatiales pour en extraire des objets spatiaux en se basant sur des requêtes portant sur les propriétés géométriques, topologiques et/ou attributaires des objets spatiaux recherchés. Le but de cet outil est d'offrir à l'utilisateur un outil simple pour mettre en place une recherche d'objets spatiaux simple et sophistiquée. Par exemple, si l'utilisateur désire étudier la déperdition de chaleur dans des bâtiments résidentiels, il n'aura pas besoin d'afficher les bâtiments industriels ou administratifs. Ainsi, à partir de la requête de création de la couche spatiale correspondante, il impose une condition sur le type de bâtiments pour afficher que les bâtiments résidentiels.

Dans ce chapitre et le chapitre précédents, nous avons décrit notre contribution en termes de représentation et interrogation de données spatiales. Comme le but de notre prototype de SIG est qu'il soit accessible via Internet, le chapitre suivant présentera comment diffuser les différentes fonctionnalités du SIG dans le cas de son exploitation en ligne.

**CHAPITRE.9 DIFFUSION & EXPLOITATION DES DONNEES SPATIALES SUR INTERNET**

Résumé.....	180
Introduction .....	180
<b>1. Définitions .....</b>	<b>180</b>
1.1. Concepts de SIG.....	180
1.2. Concepts d'Internet .....	181
<b>2. Motivations de la diffusion en ligne du SIG.....</b>	<b>181</b>
<b>3. Objectifs du SIG Web .....</b>	<b>182</b>
<b>4. Architecture de l'application SIG Web.....</b>	<b>183</b>
<b>5. Internet au service du SIG.....</b>	<b>187</b>
<b>6. Problématiques rencontrées .....</b>	<b>189</b>
Conclusion.....	190

## **RESUME**

*L'utilisation des SIG ne se limite plus à des personnes ou des services particuliers au sein d'une organisation. Cette utilisation étendue des SIG est due au progrès des nouvelles technologies de télécommunication et de partage des données. Des nouveaux besoins sont ainsi apparus notamment en ce qui concerne le traitement de données 3D et la diffusion de ces mêmes données tridimensionnelles à travers Internet. C'est dans ce cadre que s'inscrit le présent chapitre qui présente l'implémentation de l'application SIG Web 3D.*

## **INTRODUCTION**

La portée des SIG s'est accrue d'une manière remarquable tant au niveau du nombre des utilisateurs qu'au niveau des domaines d'application. En effet, les fonctionnalités d'un logiciel SIG sont de plus en plus diversifiées et surtout de mieux en mieux reconnues par une vaste communauté d'utilisateurs. Le progrès conséquent des SIG est lié principalement aux progrès dans le domaine des nouvelles technologies. Les technologies de communication, en particulier Internet, jouent un rôle extrêmement important dans le développement des SIG. Des nouvelles notions combinant les SIG et Internet sont ainsi apparues telles que "WebGIS ou SIG Web", "Web-based GIS ou SIG basé sur Internet", "Web mapping ou cartographie Web", etc. Les SIG Web peuvent être définis comme étant des sites Internet permettant d'accéder à distance à des fonctionnalités SIG telles que l'édition et l'analyse des cartes géographiques, l'affichage des données attributaires à propos un objet spatial particulier, l'analyse d'un phénomène spatial particulier, etc. Les SIG Web implantés tout autour du monde sont alors aujourd'hui très nombreux et s'appliquent sur des domaines assez diversifiés. Cependant, la plupart des SIG proposent des fonctionnalités 2D. Les SIG 3D sont rares mais constituent un domaine de recherche en pleine expansion. C'est dans ce cadre s'inscrit le présent chapitre. Nous expliquons ici notre contribution quant au développement d'un SIG Web tout en se basant entièrement sur des outils libres ce qui en fait son originalité.

Ce chapitre est organisé comme suit. La première section définit brièvement les concepts de SIG et d'Internet. Nous recensons, par la suite dans la deuxième section, nos motivations pour la diffusion du SIG en ligne sur Internet. La troisième section, quant à elle, précise les objectifs principaux du développement du SIG Web. A partir de la quatrième section, nous commençons à expliquer l'architecture du SIG Web ainsi que les différents choix techniques. La cinquième et la dernière section présente un petit aperçu des problèmes que nous avons rencontré lors de la mise en place du SIG Web.

## **1. DEFINITIONS**

Avant d'attaquer la partie qui explique le fonctionnement de l'application SIG sur Internet, nous nous permettons de définir brièvement quelques concepts qui semblent essentiels pour éclaircir quelques notions relatives aux SIG et à Internet.

### **1.1. CONCEPTS DE SIG**

Nous considérons deux concepts principaux entrant dans la composition de tout SIG : les données et les traitements. Les données sont matérialisées par l'ensemble des lots de données spatiales décrivant des objets spatiaux en termes de localisation dans l'espace, géométries et informations attributaires. Les données spatiales sont généralement stockées dans une BDS qui est, à son tour, stockée dans un

SGBDS (Système de Gestion de bases de Données Spatiales). Quant aux traitements, ils sont matérialisés par une "boîte à outils" proposée par le SIG pour le traitement des données spatiales.

### 1.2. CONCEPTS D'INTERNET

Internet, connu aussi comme "le réseau des réseaux" ou "la toile (d'araignée)" ou encore "le Web", est un réseau informatique mondial rendant accessibles des services variés autour du monde. Les concepts fondamentaux d'Internet, outre les concepts technologiques, sont : les types d'informations, l'architecture client/serveur et les outils de développement des applications Web.

On distingue principalement deux types d'informations dans une page Web : les informations multimédia (texte, images, sons, vidéos, programmes, etc.) représentant le contenu du site Web et les informations de liens (textes ou images) permettant de naviguer d'une page Web à l'autre.

L'architecture client/serveur est un mode de communication entre un ensemble d'ordinateurs d'un réseau. Elle est matérialisée par un ordinateur serveur qui contient généralement l'application Web ainsi que les données associées et les ordinateurs clients pouvant se connecter au serveur pour accéder à l'application Web. Cette architecture permet un accès instantané aux informations proposées par l'application Web disponibles pour plusieurs internautes.

Plusieurs langages spécifiques ou adaptés permettent de développer des applications Web. Comme les SIG sont matérialisés par des données spatiales stockées dans une BDS et les fonctionnalités permettant de traiter ces données, une application Web peut offrir les fonctionnalités d'un SIG grâce aux avancés technologiques dans le domaine.

### 2. MOTIVATIONS DE LA DIFFUSION EN LIGNE DU SIG

Les SIG constituent un domaine interdisciplinaire depuis qu'ils font appel à des technologies diverses (bases de données, techniques d'acquisition de données spatiales, algorithmique et programmation, visualisation 2D ou 3D, etc.). Le progrès continu des SIG au fil du temps a permis d'augmenter, en termes de quantité et de qualité, les fonctionnalités qu'ils proposent et a ainsi amplifié le nombre d'utilisateurs. Des nouveaux besoins sont alors apparus en termes de mode de diffusion et d'exploitation des fonctionnalités des SIG. Les SIG ne sont plus une application isolée au sein d'une organisation où son utilisation est limitée à des personnes ou des services particuliers. Ils constituent, de plus en plus, un acteur principal dans leur processus de développement et leur utilisation devient ouverte à un grand nombre d'utilisateurs.

Internet constitue un moyen incontournable pour faire partager les fonctionnalités d'un SIG par un nombre important d'utilisateurs. Le SIG peut alors profiter des avantages d'Internet pour diffuser ses fonctionnalités à travers la "toile".

Nous utilisons le terme "*SIG Web*" pour notre SIG diffusé en ligne. Les motivations de proposer le SIG Web se résument dans les points suivants :

- **Disponibilité et accès instantané aux données**

Internet permet un accès instantané aux données. En effet, les données ainsi que les fonctionnalités du SIG peuvent être accédées à tout moment et à partir de n'importe quel ordinateur connecté à Internet. Les données sont alors accédées à partir des ressources de données distantes. L'avantage majeur est la centralisation des données car généralement toutes les données sont regroupées dans un serveur et les utilisateurs peuvent y accéder à travers des clients Web. Cela peut rendre les données spatiales,

généralement assez conséquentes en volume, plus disponibles et surtout non redondantes car les données sont stockées dans un seul endroit et non pas sur chaque machine locale.

- **Partage de données**

Un utilisateur de SIG peut travailler sur les données spatiales individuellement ou en groupe. Il peut partager les mêmes données avec d'autres utilisateurs pour un usage collectif. Cela peut être avantageux dans le cas des projets demandant la coopération de plusieurs personnes, le travail sur les mêmes jeux de données pour des prises de décision collectives, etc. Internet constitue un outil approprié pour le partage et la communication de données pour un usage étendu.

- **Retour d'expérience**

Parmi les avantages majeurs des SIG on retrouve leur capacité de représentation de données spatiale d'une manière facilitant les prises de décisions. Dans une organisation, un utilisateur de SIG est amené généralement à travailler avec d'autres personnes sur les mêmes lots de données ou sur les mêmes projets. Un utilisateur peut faire profiter les autres utilisateurs de son expérience sur un ensemble particulier de données spatiales, ce qui peut être un avantage en termes de qualité de travail évitant de réappliquer les mêmes traitements sur les mêmes lots de données. Internet, avec ses avantages de communication et d'accès aux données, permet aux utilisateurs de communiquer et de partager leurs expériences, leurs commentaires, leurs décisions, etc. pour l'analyse d'un phénomène spatial particulier.

- **Avantages d'internet**

Internet est le moyen de communication de données le plus répandu et le plus utilisé autour du monde. Il est devenu omniprésent dans la quasi-totalité des activités quotidiennes. Un SIG peut profiter des avantages d'Internet pour l'accès en ligne à ses diverses fonctionnalités.

Une application SIG sur Internet ne sera pas installée sur toutes les machines des utilisateurs. Elle sera centralisée pour un usage commun et étendue. Ce qui pourra être très avantageux en matière de facilité de mise à jour, libération de l'espace sur les machines clientes, etc. Vu l'usage étendue d'Internet autour du monde, plusieurs langages de programmation sont inventés ou sont orientés vers Internet. Plusieurs fonctionnalités avancées du SIG peuvent être alors aisément développées pour fonctionner sur un navigateur Web.

Internet peut rendre les SIG plus interactifs. En effet, un SIG accessible via Internet peut offrir des fonctionnalités de communication, instantanées ou différées, entre un groupe d'utilisateurs concernant un phénomène spatial particulier. Cela facilitera énormément les prises de décision, le travail collectif, le partage des tâches, etc.

Internet peut aussi offrir la possibilité de mixer ou faire lier une application SIG avec d'autres applications d'une organisation. Le SIG peut jouer alors un rôle important dans le processus de développement d'une organisation.

### **3. OBJECTIFS DU SIG WEB**

Avant de choisir de diffuser le SIG sur Internet, nous avons fixé des objectifs que le SIG Web doit assurer. Les objectifs se résument dans les points suivants :

- **Rapidité et flexibilité**

Le SIG Web doit pouvoir permettre une rapidité et une flexibilité quant à l'accès à ses différentes fonctionnalités. Il faut que le SIG Web propose une performance similaire ou rapprochée de celle d'un SIG local. Une réflexion doit être menée pour choisir les techniques les plus appropriées pour le développement et la mise en ligne des différentes fonctionnalités du SIG.

- **Fonctionnalités du SIG Web**

Toutes les fonctionnalités du SIG doivent être consultées aisément en ligne. En effet, l'application SIG doit pouvoir s'exécuter intégralement dans les principaux navigateurs Web (Mozilla Firefox, Internet Explorer, Google Chrome, etc.) ainsi que sur les différentes plateformes (Windows, Linux, etc.). L'application doit aussi proposer une interface ergonomique facile à percevoir et à utiliser.

- **Utilisation du SIG Web**

Parmi les motivations de la mise en ligne du SIG est de pouvoir partager, entre les utilisateurs, les données ainsi que les traitements des objets spatiaux. L'application peut permettre alors un usage individuel ou collectif du SIG Web. Le SIG Web présentera donc un outil de communication entre plusieurs utilisateurs.

- **Gestion des utilisateurs**

La mise en ligne des fonctionnalités du SIG peut engendrer des problèmes d'accès car les mêmes données peuvent être accédées simultanément par plusieurs utilisateurs. De ce fait, une gestion d'accès doit être mise en place pour assurer le bon fonctionnement de l'application SIG en ligne.

- **Sécurité des données**

Un compte protégé par un login et un mot de passe doit être créé pour chaque utilisateur pour qu'il puisse accéder aux données spatiales ainsi que les différentes fonctionnalités du SIG Web. L'objectif de création des comptes utilisateurs est de savoir qui fait quoi. Cela permettra aussi de sauvegarder les tâches des utilisateurs sur les couches spatiales qu'ils avaient créées pour une utilisation future ou pour les partager avec d'autres utilisateurs.

#### 4. ARCHITECTURE DE L'APPLICATION SIG WEB

Dans cette section, nous décrivons l'architecture du SIG Web. Deux concepts doivent être pris en compte : Les concepts d'une application Web en général et les concepts d'un SIG Web. L'ensemble de techniques utilisées doivent permettre, d'une part, le développement de toutes les fonctionnalités attendues du SIG, et d'autre part, la mise en ligne de l'application SIG sur Internet. Plusieurs hypothèses ont été posées avant le choix des techniques les plus appropriées pour la mise en place du SIG Web :

- 1) **Recours aux outils libres** : Il s'agit de l'utilisation d'un outil SIG libre et l'enrichir par les fonctionnalités attendues du SIG Web.
- 2) **Application orientée-Web** : Il s'agit du développement de l'application du SIG en utilisant des techniques dédiées entièrement au développement Web.
- 3) **Adaptation d'une application non orientée-Web** : Il s'agit du développement de l'application SIG en utilisant des techniques non dédiées, *a priori*, au développement Web et après adapter l'application SIG pour qu'elle puisse fonctionner sur Internet.

Mais avant de choisir une parmi les trois hypothèses, nous avons défini un ensemble de critères de choix pour comparer les différentes techniques.

- **Capacité du développement 3D**

Comme l'objectif du travail est la mise en place d'un SIG pour la gestion des modèles tridimensionnels, il est important de vérifier les capacités des techniques à utiliser en termes de développement 3D, c'est-à-dire les possibilités offertes par les techniques en question pour la visualisation ainsi que le traitement d'objets 3D.

- **Capacité du développement Web**

Le SIG 3D réalisé dans le cadre de cette recherche doit pouvoir être accessible via Internet. De ce fait, les techniques utilisées pour le développement de tel SIG doivent être capables de diffuser les différents services du SIG en ligne.

- **Connexion avec PostgreSQL/PostGIS**

La BDS constitue le cœur de tout SIG. Après analyse des différentes architectures de BDS disponibles dans le monde libre, notre choix d'un SGBD pour le stockage des données spatiales s'est porté sur PostgreSQL/PostGIS. Celui-ci répond également aux besoins de procédure de connexion, d'exécution des requêtes SQL, etc.

Nous avons étudié chacune des hypothèses vues ci-dessus selon ces différents critères de choix.

⇒ Du côté des outils libres (première hypothèse), il fallait que nous étudions les SIG libres existants par rapport aux objectifs que nous avons fixés. Trois principaux SIG libres constituaient l'objet de notre étude. Il s'agit des SIG GRASS (Geographic Resources Analysis Support System), QGIS (Quantum GIS) et OpenJUMP (Open Java Unified Mapping Platform).

Développé initialement par l'U.S ACERL (United States Army Construction Engineering Research Laboratories) pour la gestion et la planification territoriales, GRASS est devenu un SIG libre avec une licence GPL (General Public License). Ce SIG libre propose une large étendue de fonctionnalités et d'applications dans plusieurs domaines travaillant sur des données spatiales en mode raster ou vectoriel. Les différents modules de GRASS sont écrits en C et C++, mais on parle aussi de Java GRASS (JGRASS) où les modules sont écrits en Java. Comme GRASS offre une licence GPL, il est possible d'accéder à son code source pour exécuter le logiciel pour des usages différents, étudier le fonctionnement des programmes proposés par le logiciel et les adapter à des besoins particuliers en accédant aux codes sources, etc. De point de vue développement 3D, GRASS permet une visualisation et animation 3D grâce au module NVIZ, outil généré à partir du module SG3D qui sert à visualiser les données matricielles, vectorielles et sites tridimensionnelles ainsi qu'à faire des requêtes 3D ou à créer des animations [GDF HANNOVER, 2005].

Quant au développement Web, il faut mettre en place une structure particulière pour pouvoir exploiter GRASS via Internet telle que la structure proposée par [RAGHAVAN, *et al*, 2002] ou celle proposée par [XIANFENG, *et al*, 2004], etc. Enfin, concernant PostgreSQL/PostGIS, il est possible de connecter GRASS à PostgreSQL grâce à un pilote dédié appelé "pg" et permettant ainsi l'exécution de toutes les commandes et les opérateurs SQL.

Le logiciel QGIS, à l'origine n'était qu'un outil de visualisation de données provenant de GRASS. Mais, il est rapidement devenu un logiciel SIG libre, publié à son tour sous licence GPL, montrant de très grandes capacités en termes de gestion de données matricielles et vectorielles. Malgré que QGIS

est capable d'interagir avec PostgrSQL/PostGIS et qu'il est possible d'accéder en ligne aux fonctionnalités de QGIS en implémentant des structures particulières telles que [IOSIFESCU, *et al*, 2010], QGIS est essentiellement destiné pour l'affichage de données 2D ce qui constitue un frein important pour son utilisation dans le cadre de ce travail.

OpenJUMP a été créé dans l'objectif d'améliorer les fonctionnalités du logiciel SIG JUMP (Java Unified Mapping Platform). OpenJUMP, publié lui aussi sous licence GPL, permet de lire et d'éditer des fichiers *Shapefile* et GML ainsi que d'afficher des données provenant des Web-services WFS (Web Feature Service) et WMS (Web Map Service). OpenJUMP permet d'accéder à des données PostGIS, mais est essentiellement destiné à l'affichage de données 2D ce qui constitue, là aussi, un frein pour son utilisation dans le cadre de ce travail.

A l'issue de cette brève présentation des principaux SIG libres, GRASS paraît l'outil SIG libre le plus approprié pour constituer un noyau pour notre SIG Web. Cependant, nous avons choisi de ne pas faire recours aux outils libres pour le développement de notre SIG pour des raisons multiples : i) Le développement d'un nouveau prototype offre plus de flexibilité et de liberté pour la mise en place des différentes fonctionnalités attendues du SIG Web. ii) Acquérir une expérience en termes de développement intégral d'un SIG (moteur 3D, gestion de la BDS, traitement des objets spatiaux, etc.). iii) Le lancement dans les codes sources de ces différents SIG libres pour comprendre les principes de leur fonctionnement a constitué un frein majeur pour leur adoption. iv) Le SIG va constituer un prototype sur lequel pourront se greffer des travaux futurs au sein de l'équipe LSIT-TRIO, etc.

⇒ La deuxième hypothèse consistait à étudier la possibilité du développement du SIG en utilisant des techniques de développement Web. Pour ce faire, nous avons cherché les techniques permettant, d'une part, d'afficher et de manipuler des objets 3D dans un navigateur Web, et d'autre part, de se connecter à une BDS PostgreSQL/PostGIS.

La représentation d'un Objet 3D sur Internet augmente d'une manière considérable l'interactivité d'un site Web. Les progrès parallèles des technologies 3D, d'une part, et des technologies Internet, d'autre part, ont créé le besoin de mettre en place des technologies dédiées pour le développement 3D sur Internet. Plusieurs langages normalisés sont alors apparus pour permettre des présentations 3D dans un navigateur Web tels que VRML (Virtual Reality Markup Language) en tant que norme ISO (International Organization for Standardization), X3D (eXtensible 3D) en tant qu'une spécification du consortium Web3D et successeur de VRML, Xj3D en tant qu'une version java de X3D, Flash 3D, O3D (Open 3D) en tant qu'une librairie Google, WebGL (Web Graphic Language), etc.

VRML, comme son nom l'indique, permet la description des modèles virtuels 3D interactifs. Il s'agit d'un langage de représentation (formes, couleurs, lumières, matériaux, textures, etc.) et d'animation (collision, translation, rotation, etc.) d'objets 3D et non pas un langage de programmation 3D. Pour le rendre plus interactif, il est possible de manipuler des modèles virtuels VRML en utilisant des scripts Java, JavaScript, etc.

X3D, successeur de VRML, permet à son tour la représentation d'objets 3D en s'appuyant sur une structuration de type graphe de scène. Il peut être également exprimé à l'aide de trois langages différents : VRML, un langage basé sur XML ou un langage binaire. Comme VRML, les modèles virtuels X3D peuvent être manipulés en utilisant des langages de programmation tels que Java, ECMAScript, etc.

O3D, une technique récente publiée par Google en 2010, permet d'afficher et de manipuler des modèles 3D très détaillés dans des pages Web. Elle se base sur JavaScript et l'API (Application Programming Interface) OpenGL (Open Graphics Library) ou DirectX pour créer des modèles 3D

interactifs. C'est une technique très intéressante depuis qu'elle combine les forces de JavaScript pour le développement de scripts principalement utilisé dans les pages web interactives ainsi que les performances d'OpenGL ou DirectX pour le développement 3D.

O3D est une technique très appropriée pour le développement de notre SIG Web. Cependant, elle n'existait pas encore pendant les débuts de nos travaux de thèse. VRML et X3D sont très performant pour la représentation d'objets 3D, cependant la nécessité de les coupler avec un langage de programmation pour les rendre plus interactifs et leurs formats basés sur XML non adéquats pour des gros volumes de données ainsi que la non existence d'un langage puissant pour l'interrogation des fichiers XML volumineux ont défavorisé telles technologies pour être utilisées dans le développement de notre SIG Web.

⇒ La troisième hypothèse consistait à étudier la possibilité de développer une application, *a priori* non orientée-Web, puis l'adapter pour qu'elle puisse être exploitée sur Internet. Il fallait donc utiliser un langage de programmation implémentant une API permettant le développement 3D. Les langages de programmation orientés-objets tels que Java et C++ peuvent être utilisés pour le développement des applications SIG car ils peuvent définir les propriétés ainsi que les comportements des objets spatiaux grâce aux notions de l'orienté-objet (OO). Quant aux API 3D, nous avons étudié principalement OpenGL et DirectX pouvant être utilisés dans Java et C++. OpenGL est une API multiplateformes proposant environ 250 fonctions pour l'affichage des scènes tridimensionnelles complexes construites à partir des primitives géométriques simples. DirectX, produit de Microsoft, propose une API qui s'appelle Direct3D dédiée pour le développement d'applications 3D.

Nous avons choisi d'adopter la troisième hypothèse car elle offre plus de liberté et de flexibilité quant au développement des différentes fonctionnalités du SIG. Pour ce faire, nous avons choisi le langage de programmation Java pour le développement du SIG car il permet la création des Applet Java pouvant être affichées et exécutées dans un navigateur Web. Et de point de vue API 3D, notre choix est arrêté sur OpenGL car il est multi-plateformes, il propose une bibliothèque de fonctions très sophistiquée, il est supporté par la quasi-totalité des cartes graphiques, etc.

Une applet Java peut être définie comme « *un programme écrit dans le langage de programmation Java et pouvant être inclus dans une page HTML. Lorsque vous utilisez un navigateur supportant la technologie Java pour afficher une page contenant une applet, le code de l'applet est transféré à votre système et exécuté par la JVM (Machine Virtuelle Java) du navigateur* » [SUN, 2010].

A l'issue de cette définition, l'architecture de l'application SIG Web est ainsi décrite dans la figure (Fig9.1) suivante.

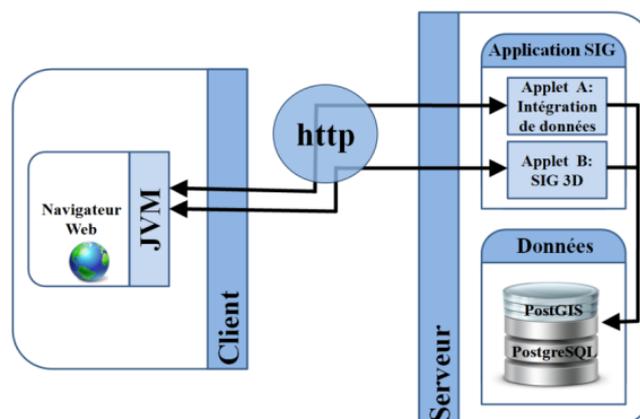
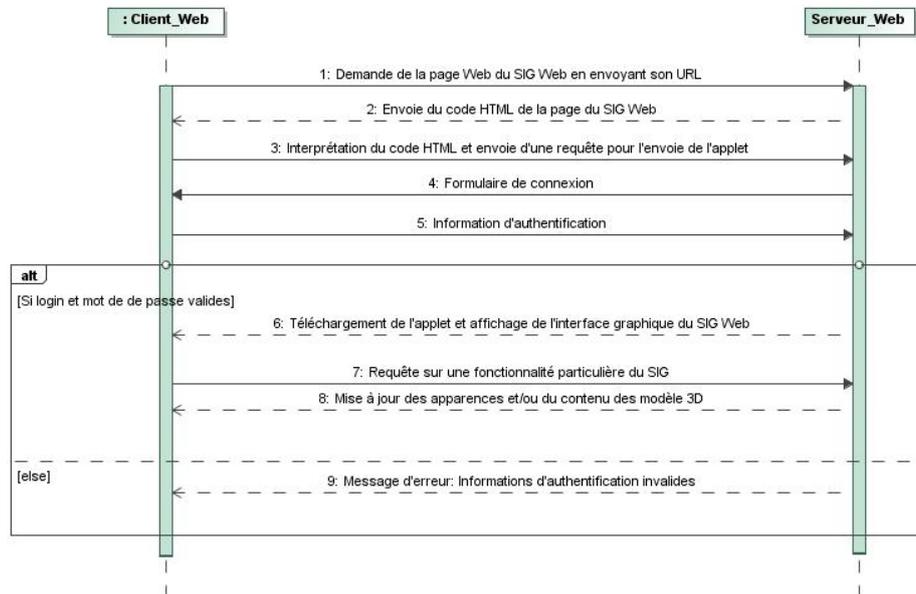


Fig9.1. Architecture simplifiée du SIG Web

Lorsque l'utilisateur accède à la page contenant l'applet, le code de l'applet (classe java de l'application SIG) est téléchargé vers la machine cliente et exécuté par la JVM du navigateur Web utilisé. La figure (Fig9.2) montre un diagramme de séquences UML (Unified Modeling Language) expliquant le fonctionnement du SIG Web matérialisé par une communication entre le poste client (navigateur Web) et le serveur Web.



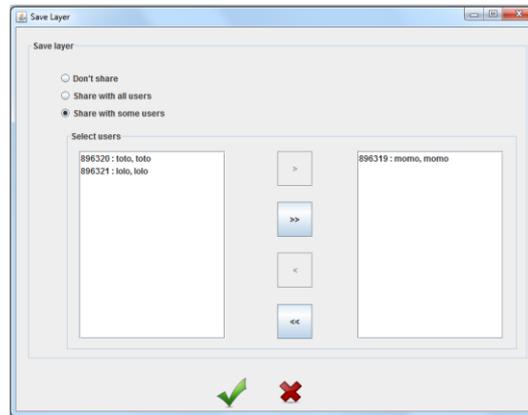
**Fig9.2.** Diagramme de séquences de la communication entre un client Web et le serveur Web dans le cadre du SIG Web

Les choix d'utiliser une applet Java pour le développement du SIG Web est motivé par les points suivants :

- Une applet combine la force de Java pour la programmation des applications interactives avancées (programmation orienté-objet, développement 3D, etc.) et les avantage d'Internet pour l'accès à distance à tel type d'applications.
- L'applet s'exécute dans le navigateur du poste client ce qui diminue considérablement la charge sur le serveur.
- La plupart des navigateurs Internet contiennent des plugins pour la JVM et donc capables de faire tourner des applets Java.
- Quelque soit la complexité de l'application, son implémentation en ligne dans une applet Java est simple à mettre en œuvre.
- La mise à jour de l'application SIG Web est très facile car il suffit de changer le code de l'applet Java et le stocker dans le serveur.
- Le langage de programmation Java est connu par une large communauté de programmeurs. Cela facilitera l'amélioration des fonctionnalités du SIG Web dans le cadre des travaux futurs de l'équipe LSIIT/TRIO.

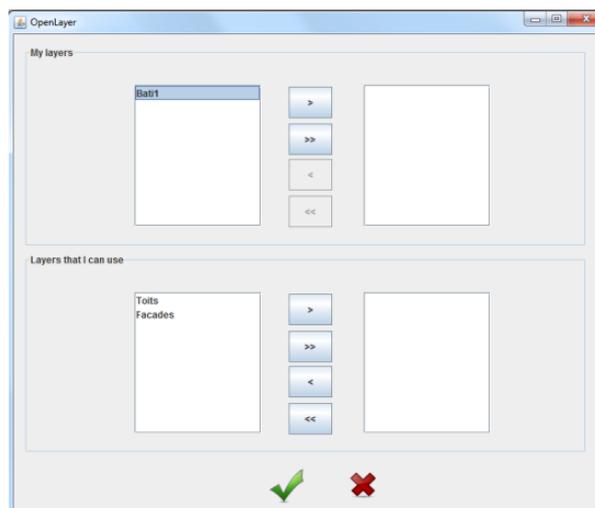
## 5. INTERNET AU SERVICE DU SIG

Dans les parties précédentes de ce chapitre, nous avons parlé de l'implémentation technique du SIG Web. Alors que dans cette partie, nous allons expliquer comment les fonctionnalités du SIG sont exploitées en ligne. Outre l'accès à distance aux différents services du SIG Web, la motivation majeure du déploiement du SIG via Internet est la disponibilité et le partage en temps réel des données ainsi que des tâches utilisateurs sur les lots de données spatiales.



**Fig9.3.** Mise en ligne d'une couche spatiale et les différentes alternatives de son partage

Lors de la création des couches spatiales, l'utilisateur connecté à l'application SIG Web grâce à son login et mot de passe a plusieurs alternatives quant au stockage de la couche spatiale. Il peut stocker la couche spatiale pour une utilisation privée, une utilisation partagée avec un certain nombre d'utilisateurs ou une utilisation partagée avec tous les utilisateurs de l'application. Une couche spatiale partagée peut être ainsi accédée, à tout moment, par plusieurs utilisateurs. La figure (Fig9.4) montre une capture d'écran de l'interface graphique permettant d'ouvrir une ou plusieurs couches spatiales. Dans cette interface, l'utilisateur peut accéder aux couches spatiales dont il avait créés ainsi que celles dont il a les droits suffisant pour y accéder.

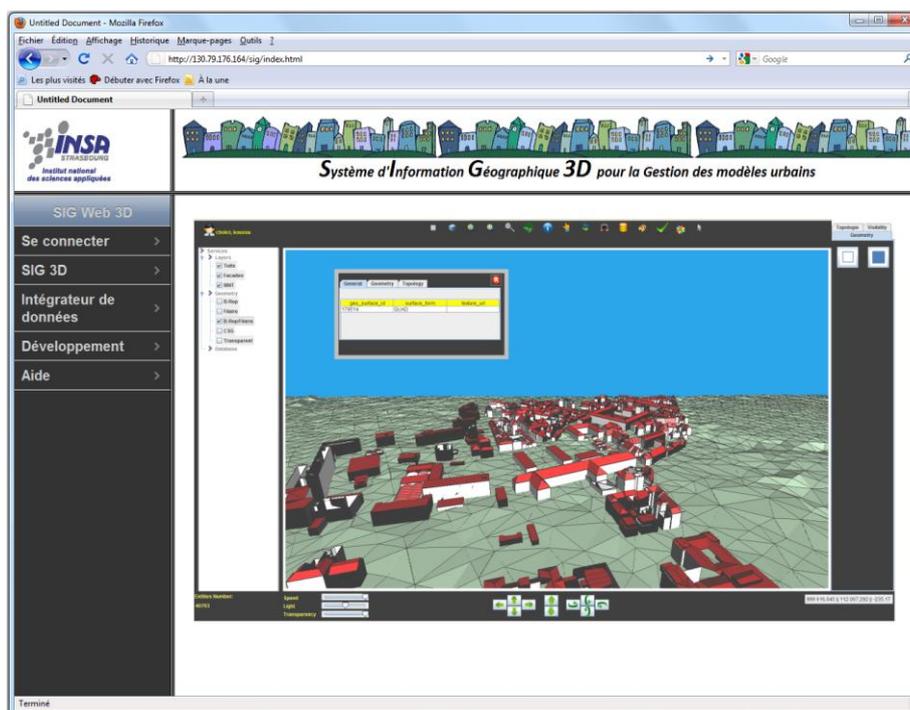


**Fig9.4.** Interface graphique pour la visualisation d'une couche spatiale stockée dans la BDS

Le partage des couches spatiales en matière de contenu et apparences constitue nos premiers travaux pour une exploitation réelle des différentes fonctionnalités du SIG. En effet, nos travaux en cours portent essentiellement sur :

- Le stockage des tâches utilisateur sur une couche spatiale issues de requêtes (géométriques, topologiques et attributaires), modification des apparences, commentaires, etc.
- La mise en place d'un système de communication en temps réel entre des utilisateurs travaillant sur les mêmes lots de données.

La figure (Fig9.5) présente un aperçu de l'application SIG exécutée dans un navigateur Web.



*Fig9.5.* Une capture d'écran de l'application SIG Web

## 6. PROBLEMES RENCONTRES

Pour le développement du SIG Web, notre objectif initial était d'adopter une technique performante pour la programmation des différentes fonctionnalités du SIG et permettant de l'intégrer facilement dans une application Web. Notre choix s'est arrêté sur l'utilisation des applets Java surtout pour profiter des capacités avancées de Java et la simplicité de son implémentation dans une application Web.

Cependant, nous avons rencontré quelques problèmes :

- Le temps de téléchargement de l'applet sur le poste client peut être important. Cela dépend de la charge du serveur, de la qualité de connexion internet de l'utilisateur, etc.
- Nous n'avons pas abordé des notions importantes dans le domaine d'Internet telles que le streaming de données sur Internet, la problématique d'accès concurrents à des mêmes ressources, les protocoles de communication, etc.
- Les moteurs de recherche ne peuvent pas indexer les applets binaires. Le site Internet hébergeant l'applet doit la compléter par un code HTML décrivant l'objectif du site pour pouvoir être indexé par les moteurs de recherche.
- Java doit être installé sur le poste client. Il peut arriver parfois des problèmes de version de JRE (Java Runtime Environment) ou de JDK (Java Développement Kit) ce qui peut gêner le fonctionnement de l'application.

Pour éviter les problèmes de version de JRE, JDK ou de Jogl (Java OpenGL), nous avons utilisé un fichier JNLP (Java Network Launching Protocol) permettant de déployer facilement une application Java sur le Web. Le principe consiste à déposer, dans le serveur hébergeant le site Web du SIG, un fichier JNLP permettant de décrire les dépendances de l'application en termes de fichiers JAR (Java ARchive) utilisés. Il suffit après de préciser l'adresse de ce fichier pour que le système déploie directement l'application. La figure (*Fig9.6*) montre un exemple d'un fichier JNLP que nous avons utilisé dans le cadre du SIG Web.

```

<?xml version="1.0" encoding="utf-8"?>
<!-- edit: change the href attribute to the same name that you give to this file. -->
<jnlp href="JNLFile.jnlp">
  <!-- edit: generic information, edit or remove all the values except
    the update check at the bottom -->
  <information>
    <title></title>
    <vendor></vendor>
    <description></description>
    <description kind="short"></description>
    <update check="background"/>
  </information>
  <resources>
    <j2se version="1.6+" />
    <property
      name="sun.java2d.noddraw"
      value="true" />
    <!-- edit: the jar containing your main class -->
    <jar href="jgraphx.jar" />
    <jar href="postgresql-8.4-701.jdbc4.jar" />
    <jar href="GISJAR.jar" main="true" />
    <extension name = "jogl" href =
"http://download.java.net/media/jogl/builds/archive/jsr-231-webstart-current/jogl.jnlp" />
  </resources>

  <!-- edit: all four details need to be changed -->

  <applet-desc
    name = "GIS Web INSA"
    main-class = "editor.GIS3DViewer"
    width = "1200"
    height = "800" >
  </applet-desc>
</jnlp>

```

**Fig9.6.** Exemple d'un fichier JNLP utilisé dans le cadre du SIG Web

La programmation des différentes fonctionnalités du SIG 3D a constitué la partie la plus importante du point de vue du temps consacré à son développement. En effet, le SIG Web 3D a été développé intégralement dans le cadre de ce travail et nous n'avons pas utilisé d'applications existantes. Les applications les plus importantes que nous avons développées sont : l'intégration des données spatiales dans la BDS, la conception et la programmation du moteur 3D, la création des modèles 3D en utilisant un système de couches spatiales, les requêtes topologiques graphiques, les requêtes spatiales, etc.

## CONCLUSION

Pour le développement de notre SIG, nous nous sommes contraints par l'obligation de l'utilisation des outils libres tant au niveau du stockage des données spatiales ainsi que les données attributaires associées (le SGBDS PostgreSQL/PostGIS) qu'au niveau d'outils de programmation du SIG 3D (Java, OpenGL, applet Java, etc.). Nous avons pu atteindre les principaux objectifs en nous basant entièrement sur l'open source, ce qui rend le coût de développement de ce SIG Web quasi-nul (hormis les coûts humains !).

Le contenu de ce chapitre était principalement technique puisqu'il s'intéressait à l'implémentation du SIG sur Internet. Avant de prendre une décision nous engageant sur la poursuite du travail, nous avons mené une réflexion relativement approfondie quant aux techniques les plus appropriées pour la

diffusion en ligne des fonctionnalités du SIG. Nous avons posé au début trois hypothèses pour le développement du SIG Web : i) Le recours aux outils libres. ii) La mise en place d'une application orientée-Web. iii) L'adaptation d'une application non orientée-Web. Nous avons adopté la troisième hypothèse en implémentant la notion d'applet Java. L'utilisation des applets Java pour le développement du SIG Web est motivée par la performance du langage Java pour le développement d'applications sophistiquées, l'utilisation de l'API OpenGL pour le développement 3D, l'intégration d'applets Java dans un navigateur Web, etc. La combinaison des différents outils et techniques libres ont aboutit à la création d'un prototype d'un SIG Web. L'évaluation de cette application SIG Web ainsi que les perspectives ou les améliorations qu'on peut y apporter constituent l'objet du chapitre suivant.



SECTION.IV BILAN SUR L'APPLICATION SIG & PERSPECTIVES

CHAPITRE.10 BILAN SUR L'APPLICATION SIG & PERSPECTIVES

Résumé.....	194
Introduction .....	194
<b>1. Bilan sur l'application SIG.....</b>	<b>194</b>
1.1. La modélisation conceptuelle du SIG Web .....	194
1.1.1. Modélisation des données spatiales.....	194
1.1.2. Modélisation de la Base de Données Spatiale (BDS) .....	195
1.1.3. Modélisation des traitements.....	195
1.2. Evaluation technique du SIG Web .....	195
1.2.1. Le code .....	196
1.2.2. L'application SIG.....	198
1.2.3. L'application Web.....	199
<b>2. Perspectives.....</b>	<b>199</b>
2.1.1. Perspectives techniques/technologiques.....	199
2.1.2. Perspectives fonctionnelles .....	201
2.1.3. Perspectives d'utilisation.....	203
2.1.4. CityGML .....	203
Conclusion.....	204

« On a besoin de s'appuyer sans cesse sur des évaluations et des jugements, et sur toutes les passions qui en résultent »  
(RIVIERE, p. 13, 1908)

## **RESUME**

*Ce chapitre reprend les travaux menés dans le cadre de cette thèse sous la forme d'un bilan. En effet, Ce bilan porte essentiellement sur le côté technique et le côté fonctionnel de notre SIG Web. A l'issue de ce bilan, nous définissons nos travaux futurs qui consistent à apporter des améliorations ainsi que des nouvelles fonctionnalités selon les besoins ou les attentes des utilisateurs du SIG Web.*

## **INTRODUCTION**

Après avoir décrit, dans les sections précédentes, les fondements théoriques et techniques de nos travaux pour la mise en place d'un prototype de SIG accessible via Internet, nous retraçons dans ce chapitre un bilan sur ces travaux. En effet, l'objectif principal de ce chapitre consiste à reprendre les différents travaux menés dans le cadre de cette thèse en matière de modélisation et de développement et les situer par rapport à nos attentes initiales du SIG. Et sur la base de ce bilan, nous pouvons identifier les points forts ainsi que les points faibles de l'application pour y apporter des améliorations, des corrections, des perspectives, des nouvelles fonctionnalités, etc.

Nous organisons ainsi ce chapitre en deux parties principales : une partie pour le bilan sur l'application SIG et une partie pour les perspectives. La partie "*Bilan sur l'application SIG*" porte sur la modélisation (données spatiales, BDS et traitements), ainsi que sur l'aspect technique du SIG Web. Quant à la partie "*Perspectives*", elle porte sur l'amélioration du présent SIG Web en termes de perspectives techniques/technologiques, perspectives fonctionnelles, perspectives d'utilisation et d'intégration d'autre sources de données dans le SIG.

## **1. BILAN SUR L'APPLICATION SIG**

Par le biais de cette partie, nous réalisons un bilan de l'application SIG Web, menée dans le cadre de nos travaux de thèse, en termes de modélisation conceptuelle du SIG Web, de technologies employées pour le développement ainsi que l'utilisation du SIG Web.

### **1.1. LA MODELISATION CONCEPTUELLE DU SIG WEB**

La modélisation des données et des traitements est une étape primordiale avant la mise en place de tout SIG. Dans le cadre de ce travail, nous avons consacré une partie importante de nos travaux pour la modélisation (données spatiales, Base de Données Spatiales, une réflexion sur les traitements, etc.).

#### **1.1.1. MODELISATION DES DONNEES SPATIALES**

La modélisation spatiale est une discipline extrêmement importante dans le domaine des SIG. Pour ce faire, nous nous sommes basés sur quelques approches existantes pour la modélisation 3D géométrique et topologique des données spatiales.

Cependant, il est difficile d'auto-évaluer la modélisation des données spatiales pour plusieurs raisons : les objectifs de représentation peuvent varier d'un modèle 3D à un autre (le niveau de détail, le volume de données, la redondance ou non des données, etc.), des contraintes technologiques peuvent influencer le choix d'une approche de modélisation tel que le choix d'un Système de Bases de Données Spatial (SGBDS), les concepts de modélisation peuvent varier d'une approche à l'autre, etc. Dans le cadre de notre SIG, la modélisation consistait principalement à proposer un modèle vectoriel pour la représentation de tout objet 3D, en particulier des modèles urbains. Pour ce faire nous avons utilisé une approche surfacique (B-Rep modifiée) pour plusieurs raisons : la capacité de la méthode pour la représentation d'objets complexes, la représentation surfacique peut être adaptée pour

le stockage dans PostgreSQL/PostGIS, l'intuitivité et la simplicité de la méthode, etc. L'avantage de B-Rep est qu'elle était une approche très adaptée pour la mise en place de notre algorithme de calcul topologique ainsi que pour le stockage des solides dans la BDS.

### 1.1.2. MODELISATION DE LA BASE DE DONNEES SPATIALE (BDS)

La BDS constitue le cœur de tout SIG. Elle enregistre les données spatiales ainsi que les données attributaires associées, voire des fonctions et des procédures stockées pour la manipulation de ces données. La BDS est généralement stockée dans un SGBDS, mais elle peut être aussi stockée dans un SGBD normal en implémentant une structure pour le stockage des données spatiales.

L'architecture à plusieurs niveaux, que nous avons utilisée pour la mise en place de la BDS, a facilité son interrogation. Par contre la redondance de données peut être vue de deux côtés : un côté positif matérialisé par la rapidité d'interrogation et d'affichage des modèles 3D (en évitant surtout les jointures) et un côté négatif matérialisé par le grand volume de données car il existe quelques données qui se répètent plusieurs fois comme par exemple les coordonnées des sommets des surfaces et des solides. Pour assurer un certain niveau de fiabilité, la BDS est modélisée à travers les trois niveaux de modélisation : conceptuel, logique et physique.

Nous avons mis en place un algorithme pour l'intégration de certains types de fichiers dans la BDS tels que les fichiers SIG comme les fichiers *Shapefile* et les fichiers CAO (Conception Assistée par Ordinateur) comme les fichiers *DXF* et *DWG*. Le processus d'intégration aboutit à une BDS cohérente ayant une structure facile à comprendre et à interroger. Cependant, en appliquant certains tests de ce processus d'intégration sur des fichiers de différentes tailles, nous avons constaté que le temps d'intégration peut être relativement long pour des fichiers volumineux. Cela peut être dû principalement à la lecture exhaustive de tous les champs des tables sources pour en extraire les données géométriques (coordonnées, surfaces, solides, etc.) ainsi que les données attributaires dans des vues SQL (Structured Query Language). Cependant ce processus d'intégration est fait une seule fois et la BDS sera optimisée pour une interrogation et un rendu 3D plus aisés et plus rapides. Nous avons aussi utilisé des vues SQL pour stocker les données géométriques matérialisées par des requêtes SQL sur les tables sources. Le but était principalement d'isoler les données géométriques dans des vues pour, d'une part, optimiser le rendu des modèles 3D, et d'autre part, sécuriser les données sources.

### 1.1.3. MODELISATION DES TRAITEMENTS

Les traitements complètent les données spatiales stockées dans la BDS pour la mise en place du SIG. Comme il n'existe pas jusqu'à maintenant, des méthodes standardisées pour la conception des données et des traitements des SIG, nous n'avons pas suivi une méthode formelle pour la conception des traitements du SIG.

Du côté fonctionnalités du SIG, nous n'avons pas suivi une approche de modélisation. La mise en place d'une nouvelle fonctionnalité est réalisée au fur et à mesure de l'avancement du développement du SIG. Nous nous sommes inspirés des outils SIG existants pour le traitement des données spatiales 2D et 3D. Les principales fonctionnalités sont expliquées ci-dessous dans la section 1.2.2 de ce chapitre.

Du côté visualisation 3D, il était nécessaire que nous développions un moteur 3D pour le rendu tridimensionnel d'objets spatiaux. Pour ce faire, nous avons utilisé le langage Java ainsi que l'API (Application Programming Interface) OpenGL.

## 1.2. EVALUATION TECHNIQUE DU SIG WEB

L'évaluation technique porte principalement sur les technologies mise en place pour le développement du SIG. Il s'agit de l'évaluation du code de l'application Java, les fonctionnalités de l'application SIG elle-même ainsi que les technologies utilisées pour la mise en ligne sur Internet telle application.

### 1.2.1. LE CODE

Nous commençons par quelques statistiques du code de l'application SIG Web (en utilisant le plugin "Eclipse Metrics" implémenté dans l'environnement de développement Java Eclipse). La figure (Fig10.1) montre quelques chiffres de l'application tels que le nombre d'attributs, le nombre de méthodes, le nombre de classes, le nombre de ligne de codes, etc.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum
▶ Number of Overridden Methods (avg/max per type)	21	0,143	0,508	4	/MyGIS/src/editor/ImageSelection.java
▶ Number of Attributes (avg/max per type)	998	6,789	39,542	469	/MyGIS/src/editor/GIS3DViewer.java
▶ Number of Children (avg/max per type)	5	0,034	0,216	2	/MyGIS/src/Geometry/GeometricForm.java
▶ Number of Classes (avg/max per packageFragn)	147	24,5	36,39	105	/MyGIS/src/editor
▶ Method Lines of Code (avg/max per method)	21079	14,844	32,51	570	/MyGIS/src/editor/FreeGraphEditor.java
▶ Number of Methods (avg/max per type)	1365	9,286	28,064	312	/MyGIS/src/editor/GIS3DViewer.java
▶ Nested Block Depth (avg/max per method)		2,042	1,382	10	/MyGIS/src/editor/GIS3DViewer.java
▶ Depth of Inheritance Tree (avg/max per type)		2,884	1,992	7	/MyGIS/src/editor/EditorActions.java
▶ Number of Packages	6				
▶ Afferent Coupling (avg/max per packageFragn)		3,167	3,078	9	/MyGIS/src/Geometry
▶ Number of Interfaces (avg/max per packageFragn)	1	0,167	0,373	1	/MyGIS/src/editor
▶ McCabe Cyclomatic Complexity (avg/max per type)		2,399	3,377	42	/MyGIS/src/Topology/TopologicalOperation.java
▶ Total Lines of Code	27395				
▶ Instability (avg/max per packageFragment)		0,607	0,254	1	/MyGIS/src
▶ Number of Parameters (avg/max per method)		0,894	1,473	14	/MyGIS/src/Geometry/PolyLine.java
▶ Lack of Cohesion of Methods (avg/max per type)		0,29	0,369	0,995	/MyGIS/src/editor/GIS3DViewer.java
▶ Efferent Coupling (avg/max per packageFragn)		6	8,103	24	/MyGIS/src/editor
▶ Number of Static Methods (avg/max per type)	55	0,374	0,793	5	/MyGIS/src/editor/GIS3DViewer.java
▶ Normalized Distance (avg/max per packageFragn)		0,391	0,256	0,75	/MyGIS/src/Geometry
▶ Abstractness (avg/max per packageFragment)		0,002	0,004	0,009	/MyGIS/src/editor
▶ Specialization Index (avg/max per type)		0,108	0,448	3	/MyGIS/src/Hyperlink.java
▶ Weighted methods per Class (avg/max per type)	3406	23,17	74,787	779	/MyGIS/src/editor/GIS3DViewer.java
▶ Number of Static Attributes (avg/max per type)	30	0,204	0,764	6	/MyGIS/src/editor/EditorRuler.java

Fig10.1. Statistiques du code Java du SIG Web

Vu la diversité des fonctionnalités que le SIG Web doit assurer, la taille de l'application (27395 lignes de code) est relativement importante. Cela peut avoir des conséquences négatives lors du chargement de l'applet Java dans un navigateur Web.

Pour des raisons d'organisation du code, nous avons découpé l'application en quatre packages : Le package "Editor" pour les principales fonctionnalités de rendu 3D et de traitements d'objets et des couches spatiales, le package "Geometry" pour les traitements géométriques, le package "Topology" pour les traitements topologiques et le package "Shortest\_path" pour le calcul du plus court chemin. Quant au rendu 3D, nous avons utilisé l'API (Application Programming Interface) *OpenGL* et pour les requêtes graphiques, nous avons utilisé l'API *mxGraph*. La (Fig10.2) montre une illustration des dépendances du package "Editor" avec les autres packages de l'application.

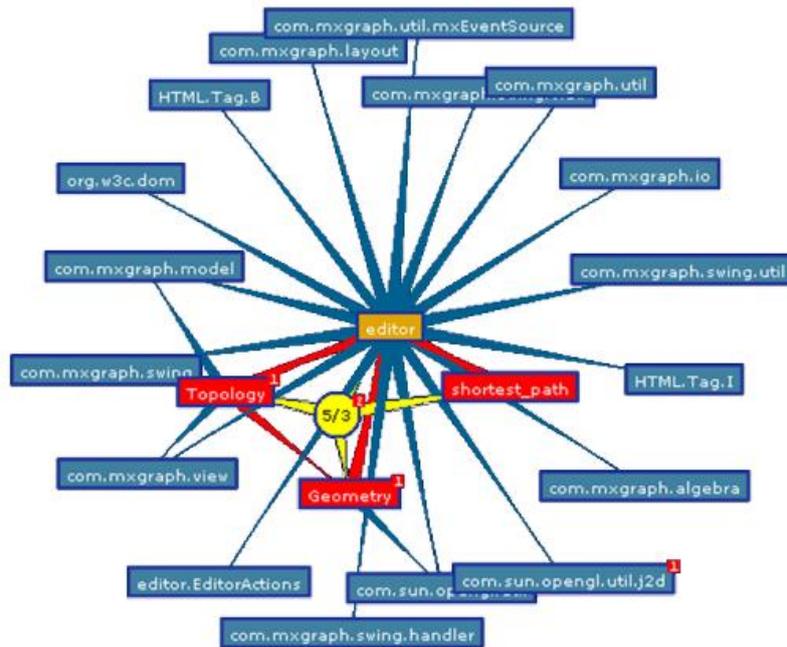


Fig10.2. Dépendances entre les packages du projet SIG Web

L'organisation de l'application en sous-modules nous a permis, d'une part, d'avoir un code plus structuré, plus lisible, maintenable, facilement réutilisable, simple à utiliser, etc. et, d'autre part, tirer profit des concepts de l'orienté-objet permettant de représenter facilement les propriétés ainsi que les comportements des objets spatiaux.

Nous nous intéressons maintenant aux performances du code Java lui-même. Pour ce faire, nous avons installé un plugin qui s'appelle "VisualVM" dans l'EDI (Environnement de Développement Intégré) Eclipse. Cet utilitaire permet de représenter, à travers des courbes et des statistiques, l'évolution dans le temps du taux d'occupation de la CPU, la taille de mémoire réelle et la mémoire utilisée, le nombre de classes chargées et le nombre de processus en exécution. Cet utilitaire détecte facilement la consommation de l'application en termes de mémoire utilisée et y apporte les corrections nécessaires telles que l'optimisation des processus, des classes, etc.

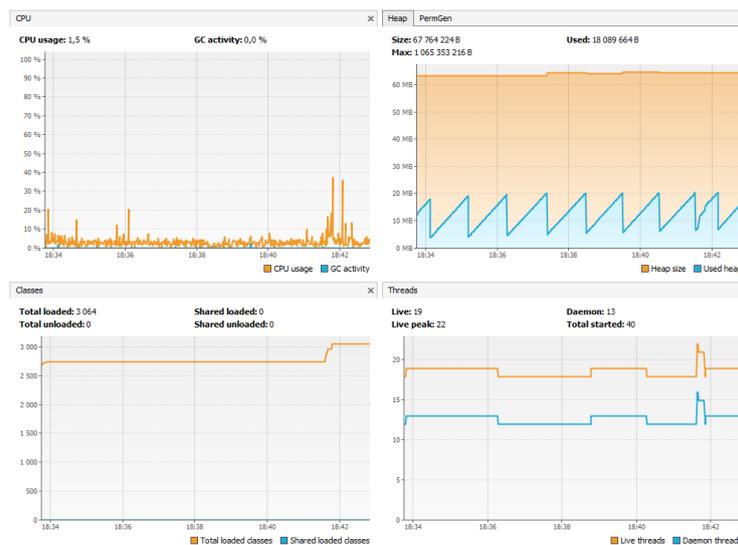


Fig10.3. Courbes décrivant la performance de l'application SIG Web

Pour faire ces différents tests, nous avons utilisé une machine ayant la configuration suivante :



**Fig10.4.** La configuration de la machine sur laquelle nous avons développé le SIG Web

D'après les courbes vues dans la (Fig10.3), nous remarquons que, lors de l'exécution du SIG, le système fonctionne normalement et les capacités de calcul restent élevées.

### 1.2.2. L'APPLICATION SIG

Quant à l'application SIG, nous discutons les fonctionnalités proposées par le SIG. En effet, nous distinguons deux types de fonctionnalités :

- Fonctionnalité d'IHM (Interaction Homme Machine) : il s'agit des outils proposés par le SIG pour interagir avec les modèles 3D rendus à l'écran tels que le déplacement, la rotation, etc.
- Fonctionnalités SIG : Ce sont les fonctionnalités proposées par les SIG pour le traitement des couches et objets spatiaux.

Le tableau (Tab10.1) suivant représente un petit aperçu de quelques fonctionnalités (IHM et SIG) pouvant être fournies par un SIG et justifie leur existence parmi celles proposées par notre SIG Web.

	Fonctionnalités	Oui	Non
Fonctionnalités IHM	Déplacement (vertical, horizontal, haut, bas, vers l'avant, vers l'arrière)	✓	
	Zoom (avant, arrière)	✓	
	Zoom sur une cible		✓
	Centrer sur une cible		✓
	Rotation axe X	✓	
	Rotation axe Y	✓	
	Rotation axe Z		✓
	Sélection graphique	✓	
	Sélection par le biais d'une requête	✓	
Fonctionnalités SIG	Représentation 2D		✓
	Représentation 3D	✓	
	Représentation vectorielle	✓	
	Représentation raster		✓
	Interrogation graphique	✓	
	Interrogation par les biais de requêtes	✓	
	Interrogation topologique	✓	
	Interrogation attributaires	✓	
	Recherche d'objets spatiaux	✓	
	Système à couches spatiales	✓	
	Fusion de couches spatiale	✓	
Saisie de données		✓	
Conversion de données		✓	

**Tab10.1** Aperçu de quelques fonctionnalités qui existent ou non dans le SIG Web

A l'issue de ce tableau, on peut constater que le SIG Web développé dans le cadre de cette thèse contient la plupart des fonctionnalités basiques que doit fournir tout SIG. Du côté IHM, plus d'attention doit être accordée surtout pour rendre le SIG Web plus interactif tels que l'ajout de quelques fonctionnalités pour rendre la navigation dans les modèles 3D plus souple. Du côté fonctionnalités SIG, la liste ne peut pas être exhaustive, mais cela n'empêche pas la nécessité d'avoir quelques fonctionnalités basiques telles que la représentation, 2D et/ou 3D, vectorielle et/ou raster des données spatiales, l'interrogation des modèles spatiaux pour en extraire des objets spatiaux, etc.

### 1.2.3. L'APPLICATION WEB

Après avoir développé toutes les fonctionnalités du SIG, l'étape finale consistait à diffuser telles fonctionnalités en ligne sur Internet. Nous avons alors employé certaines technologies pour rendre le SIG accessible via Internet.

Du côté technologies mises en place, nous avons utilisé des applets Java. La motivation de leur utilisation est due essentiellement à deux principales raisons : i) Les performances du langage Java en matière de développement d'application avancées, sa capacité d'utiliser des API telles qu'OpenGL pour la mise en place d'un moteur 3D, etc. ii) La simplicité et la facilité de son intégration dans une application Web. En intégrant une applet Java dans un site Web, il devient plus interactif. Cependant une applet Java peut présenter des problèmes de sécurité et de lourdeur de chargement.

Pour l'application Web, il y avait une nécessité de tester l'application SIG sur le réseau Internet. Cependant, pour des raisons de sécurité, nous n'avons eu la possibilité de tester l'application que dans un Intranet. Toutes les fonctionnalités du SIG Web 3D peuvent être exécutées dans un navigateur Web avec la même souplesse qu'en local.

## 2. PERSPECTIVES

Il n'existe pas de limites pour l'alimentation du SIG Web par des nouveaux services. Il est toujours possible d'améliorer l'application SIG car on peut avoir des nouvelles idées, on peut constater des limites ou des défaillances dans l'application, des nouvelles technologies apparaissent dans le domaine informatique, etc. Nous présentons, dans cette partie, les perspectives ou les travaux futurs dans le cadre du développement de notre SIG Web.

### 2.1.1. PERSPECTIVES TECHNIQUES/TECHNOLOGIQUES

Au début des nos travaux de thèse, nous avons étudié les technologies que nous pouvons mettre en place pour le développement des différents services du notre SIG Web. Notre attention était plus orientée vers des outils libres essentiellement pour minimiser au plus le coût du développement. Le défi consistait à savoir comment adapter et surtout comment intégrer ces différentes technologies pour la réalisation du SIG accessible sur Internet.

A l'instar de l'auto-évaluation technique décrite dans la partie 1.2 ci-dessous, on peut invoquer quelques perspectives pour l'amélioration des capacités du SIG Web. Les perspectives techniques portent essentiellement sur :

- **SGBDS et BDS :**

Le choix d'un SGBDS, en se basant sur des critères de choix fixés *a priori*, s'avère être une étape cruciale avant de commencer le développement du SIG. Nous étions intéressés aux données spatiales

en termes de modélisation et de stockage avant de procéder à la réalisation des différentes applications du SIG Web.

Comparé à Oracle Spatial, PostgreSQL/PostGIS présente quelques retards surtout en termes de rapidité d'interrogation et de capacités de stockage d'objets 3D. Nos travaux portaient, à un moment donné, sur l'amélioration de PostGIS en l'alimentant par des nouvelles fonctionnalités. Nous avons développé une fonction sur PostGIS qui permet de stocker des solides 3D, et pour calculer des opérations topologiques sur ces solides 3D, nous avons développé des fonctions dans le code Java. Cela peut être désavantageux car il faut absolument utiliser l'application Java pour mettre en place des calculs topologiques. Nous visons d'améliorer ces calculs topologiques entre des solides 3D en développant des fonctions sur PostGIS qui peuvent être exécutées directement dans le SGBD PostgreSQL ou dans le code Java. Cela peut apporter une valeur ajoutée en termes de rapidité de calcul ainsi que de flexibilité d'interrogation.

Pour alimenter le SIG, nous avons développé un algorithme pour l'intégration des données spatiales dans la BDS. Cependant, l'intégration se limite actuellement à des fichiers *shapefile*. On pourrait étendre la procédure d'intégration à d'autres types de fichiers tels que les fichiers CAO (*DXF* et *DWG*), fichiers ASCII, fichiers GML, etc. Lors du processus il s'agit de déchiffrer les représentations géométriques, proposées par les fichiers d'origine en des représentations lisibles pour pouvoir les utiliser dans nos programmes Java. Nous avons pensé alors à étendre les capacités de notre SIG pour qu'il puisse accéder à ces types de fichiers en mode natif. Il s'agit d'exploiter directement ces fichiers sans passer par une procédure d'intégration.

L'architecture à plusieurs niveaux de la BDS peut impliquer plusieurs tables. Cette architecture peut être avantageuse car la BDS peut contenir, outre les tables géométriques, un nombre illimité de tables attributaires. Néanmoins, les requêtes spatiales peuvent contenir des multiples jointures pouvant être des sources d'erreurs et de lourdeur d'interrogation. Une réflexion doit être menée pour optimiser cette architecture.

Les données spatiales peuvent être, de nos jours, stockées et exploitées dans des entrepôts et magasins de données spatiales. Ces derniers possèdent des structures de données facilitant les prises de décision grâce à des langages spécifiques tels que SOLAP (Spatial On-Line Analytical Processing). Cela constitue une perspective importante pour le stockage et l'analyse des données spatiales du SIG.

- **Code de l'application**

La programmation est la traduction de l'ensemble d'idées du programmeur en un ensemble d'instructions formelles selon les spécifications d'un langage de programmation donné. Nos orientations quant au choix d'un langage de programmation portaient essentiellement sur les langages orientés-objets. Pour ce faire, nous avons utilisé le langage de programmation Java.

A l'issue des tests, expliqués dans la section 1.3.1 de ce chapitre, que nous avons effectués sur l'application Java, nous pouvons constater que le fonctionnement ne pose pas des problèmes particuliers. Cependant, il est toujours possible d'optimiser le code en utilisant diverses techniques telles que la programmation parallèles en utilisant le concept des processus, inhiber les processus en état d'exécution mais qui ne sont pas utilisés, détruire les variables quand leur utilisation est terminée, gérer les tailles des variables, etc. Une procédure de nettoyage du code est envisagée pour l'optimiser.

Pour assurer une certaine sécurité lors de l'utilisation des applets Java dans une application Web, il est recommandé de signer tous les fichiers JAR (Java ARchive) utilisés par l'applet. La signature numérique des applets est simple, il suffit de taper quelques instructions dans un éditeur de lignes de commande. Mais l'inconvénient majeur des applets est qu'il faut renouveler les signatures numériques tous les six mois. Pour automatiser la création de telles signatures, nous comptons développer un module qui permet de les renouveler automatiquement une fois expirées.

Comme nous avons développé plusieurs algorithmes pour mettre en place les différentes fonctionnalités du SIG, leur optimisation est nécessaire pour assurer une qualité optimale des différents services du SIG et pour accélérer le temps de leur exécution. Pour ce faire, tous les algorithmes doivent être révisés pour les rendre plus légers et en gardant les mêmes performances.

- **Technologies employées**

Le choix des technologies à utiliser constituait une étape importante avant de commencer le développement du SIG Web. Comme il existe, de nos jours, une multitude de technologies, nos choix se basaient sur un certain nombre de critères que nous avons fixés au début. Les technologies retenues sont entièrement libres dans les différents niveaux de l'application SIG : PostgreSQL/PostGIS pour le stockage des données spatiales ainsi que les données attributaires associées, Applet Java pour la programmation des différentes applications du SIG et pour son intégration dans un site Web, Eclipse pour le développement Java, WAMP (Windows Apache MySQL PHP) en tant que serveur Web, WebExpert en tant qu'éditeur HTML, etc.

En utilisant ces diverses technologies, nous avons pu atteindre les principaux objectifs. Cependant, ces technologies peuvent être intégrées avec d'autres technologies pour améliorer la qualité du SIG Web.

O3D (Open 3D) est une technologie, proposée récemment par Google, permettant de déployer et de manipuler des modèles 3D très précis (textures, effets spéciaux, luminosité, etc.) dans un navigateur Web. Cette technologie se base sur le langage de scripts JavaScript et l'API OpenGL pour la création d'applications 3D interactives. Un SIG peut profiter des avancées de cette technologie pour déployer des fonctionnalités sophistiquées sur Internet.

Flex est une solution de développement et de déploiement des applications Internet riches (RIA). C'est une technologie très appropriée pour la création d'interfaces graphique très avancées et pouvant être intégrée dans une page Web. Cette technique peut être couplée avec une applet Java au sein d'une même application. En utilisant Flex et une applet Java on peut rendre le SIG Web plus ergonomique et plus interactif.

Oracle Spatial montre de plus en plus des capacités avancées tant au niveau du stockage d'objets spatiaux 3D qu'au niveau de traitement de tels objets spatiaux. Oracle spatial n'est pas libre, mais son utilisation dans un SIG peut apporter une valeur ajoutée en termes de fonctionnalités spatiales avancées, rapidité de calcul, etc.

### **2.1.2. PERSPECTIVES FONCTIONNELLES**

Malgré l'existence des fonctionnalités basiques que doit fournir tout SIG telles que la visualisation de données spatiales, leur stockage et interrogation, leur représentation d'une manière facilitant leurs analyse et interprétation, etc., il n'existe pas de limites pour enrichir un SIG par des nouvelles fonctionnalités.

- **Visualisation 3D**

La visualisation et la représentation 3D des données spatiales constituent des fonctionnalités cruciales, voire la raison d'être, de tout SIG. Pour ce faire, l'utilisateur doit pouvoir afficher et manipuler (déplacement, zoom, rotation, sélection, etc.) aisément un modèle 3D.

Comme nous avons choisi de développer intégralement les différentes applications du SIG Web sans faire recours à des outils SIG libres, nous étions amenés à développer un moteur 3D pour l'affichage et la manipulation d'objets spatiaux. Nous avons essayé alors d'optimiser les capacités de ce moteur 3D (textures automatiques, utilisation des quaternions pour les rotations, utiliser des techniques d'optimisation OpenGL, etc.). Pour la mise en place de ce moteur 3D, nous sommes passés par plusieurs étapes telles que la conception de l'interface graphique, le visionneur 3D, les textures, l'animation d'objets 3D, etc. Cependant, ce moteur 3D peut être amélioré. Pour ce faire, il est envisageable d'optimiser le fonctionnement de ce moteur en appliquant plusieurs techniques telles que bien gérer les boucles et les instructions conditionnelles, bien gérer les ressources, minimiser la taille et le nombre de variables et d'instructions dans les boucles infinies, etc.

Pour des raisons d'optimisation (rapidité d'affichage des modèles 3D, libération de la BDS, etc.), les données spatiales sont téléchargées dans une mémoire tampon avant d'être rendue à l'écran. A l'issue de plusieurs expériences que nous avons effectuées sur l'application, nous avons constaté que le rendu des modèles 3D à partir de la mémoire tampon est beaucoup plus souple, rapide et performant qu'à partir de la BDS. Malgré avoir effectué quelques procédures d'optimisation de la mémoire tampon telles que l'utilisation des "*ArrayList*" au lieu des "*Vector*" pour le stockage des modèles 3D, la gestion des connexions à la BDS lors du chargement des données, etc., nous avons remarqué qu'à partir de d'un certain nombre de polygones dans la mémoire tampon, les capacités du système commencent à se dégrader. Une réflexion sur l'optimisation de la mémoire tampon doit être mise en place pour améliorer l'affichage et la gestion des modèles 3D.

- **Fonctionnalités SIG**

Nous avons fixé l'ensemble de fonctionnalités que doit fournir le SIG dont les principales sont : l'intégration des données spatiales à partir des fichiers *shapefile* dans la BDS, le rendu 3D des modèles spatiaux, le calcul d'opérations topologiques entre des objets spatiaux, l'interrogation des modèles 3D pour en identifier des objets spatiaux particuliers, etc. Nous envisageons alors d'étendre le SIG Web par des nouvelles fonctionnalités.

Nous avons proposé, par le biais de ce SIG, une interface graphique pour l'interrogation des différents modèles 3D. L'interrogation est matérialisée par une requête SQL sur une ou plusieurs tables de la BDS. Une requête peut alors être simple ou complexe. Il est très important d'améliorer la procédure d'interrogation en créant des requêtes complexes à partir d'un ensemble de requêtes simples. Cela peut faciliter la tâche de l'utilisateur pour créer des requêtes plus sophistiquées.

Jusqu'à maintenant, une requête SQL ne peut porter que sur une seule BDS pour la création d'une couche spatiale. Comme perspective, nous pouvons améliorer l'interrogation pour qu'elle porte sur plusieurs BDS en même temps pour la création des couches spatiales plus personnalisées.

Parmi les apports d'un SIG est sa manière de représenter les données d'une manière facilitant les prises de décision. Le présent SIG permet d'utiliser des couleurs et des textures différentes pour

distinguer les couches spatiales ou les objets spatiaux après une requête d'interrogation. Il est très important d'alimenter le SIG par des nouvelles fonctionnalités pour qu'il puisse représenter les résultats d'une analyse spatiale par des graphiques, des tableaux, etc. ce qui pourra faciliter l'interprétation et la représentation de ces résultats d'analyse.

Au lieu de créer une IHM définie *a priori*, il est important de mettre en place une application simplifiée de création de composants graphiques et de les configurer selon les préférences de l'utilisateur. Cela laisse l'utilisateur libre de définir une IHM et de programmer son comportement de manière dynamique.

### 2.1.3. PERSPECTIVES D'UTILISATION

Le SIG doit pouvoir être utilisable aussi bien par des utilisateurs potentiels que par des utilisateurs occasionnels. L'utilisation dépend de plusieurs facteurs tels que le niveau d'expertise de l'utilisateur, la qualité des données spatiales, le contexte d'utilisation, etc.

- **Travail collaboratif**

Le SIG peut être utilisé individuellement ou collectivement. En effet, les mêmes lots de données peuvent être exploités par plusieurs utilisateurs pour des objectifs communs ou différents.

La diffusion des différents services du SIG sur Internet est due essentiellement à l'alimentation du SIG par des capacités de travail collaboratif. En effet, nous envisageons d'étendre ce travail par des fonctionnalités permettant à un groupe d'utilisateurs de travailler sur les mêmes lots de données pour des prises de décision collectives. Internet peut être utilisé alors comme un outil de communication pour les utilisateurs du SIG. Un utilisateur peut faire profiter les autres utilisateurs de son expérience sur des lots de données spatiales (retour d'expérience, interprétations, commentaires, etc.).

- **Interactivité**

L'interactivité caractérise les possibilités de communication entre l'utilisateur et le SIG Web. En effet, pour des raisons d'optimisation de l'IHM, nous avons essayé de limiter l'accès aux différentes fonctionnalités du SIG que via l'applet Java. Cela rend les possibilités de l'application Web relativement comparables à celles d'une application locale.

Plusieurs aspects doivent être pris en considération pour l'amélioration de l'interactivité de ce SIG Web. De point de vue ergonomique, le SIG Web doit être le plus simple possible de point de vue utilisation, les résultats d'une interrogation du SIG doivent être représentés d'une manière facilitant leurs perception et interprétation, l'utilisateur doit pouvoir à tout moment se repérer dans les services du SIG Web, etc. De point de vue assistance de l'utilisateur, il est très important de mettre en place une rubrique d'aide expliquant la manière de travail sur les différentes composantes du SIG Web et, en cas d'erreur, les messages d'erreur doivent être bien clairs pour bien orienter l'utilisateur.

### 2.1.4. CITYGML

CityGML est une spécification OGC (Open Geospatial Consortium) pour la représentation de modèles urbains 3D. Il propose une multitude de modèles géométriques, topologiques et sémantiques pour modéliser des objets urbains 3D dans les différents niveaux de détail. Parmi les objectifs de création de ce modèle est de considérer les modèles 3D urbains actuels comme étant purement graphiques ou géométriques et négligeant les aspects sémantiques et topologiques. Au début de nos travaux de thèse,

nous avons étudié la possibilité d'adopter CityGML comme modèle spatial pour notre SIG. Cependant, nous étions contraints par deux désavantages de CityGML : i) Il n'existait pas encore un intégrateur des données CityGML vers un SGBDS particulier tel que PostgreSQL/PostGIS. ii) Le format XML (textuel) de CityGML présente plusieurs désavantages du fait qu'un fichier CityGML peut atteindre rapidement un grand volume de données surtout pour des modèles urbains 3D volumineux ou détaillés au contraire des fichiers binaires des bases de données qui sont plus compactes, les parseurs des fichiers XML sont beaucoup plus longs que le langage SQL pour l'interrogation des Bases de Données, etc. Pour ces raisons, nous avons choisi de faire recours aux formats intégrables dans une BDS PostgreSQL/PostGIS. D'autre part, des logiciels tels que FME proposent la conversion et l'intégration du format CityGML vers d'autres BDS. Il ne nous paraissait pas nécessaire de redévelopper une nouvelle version de ce genre d'intégrateur. Le tableau (Tab10.2) montre une comparaison entre les fichiers sources de notre SIG et le format CityGML.

	<b>Fichiers sources (<i>shp, dwg, dxf</i>) de notre SIG</b>	<b>CityGML</b>
Format	<ul style="list-style-type: none"> <li>- <i>Shp</i> est le format binaire d'ArcGIS.</li> <li>- <i>DWG</i> est le format binaire natif d'Autocad.</li> <li>- <i>DXF</i> en est le format ASCII d'Autocad.</li> </ul>	<ul style="list-style-type: none"> <li>- CityGML est un format XML.</li> </ul>
Géométrie	<ul style="list-style-type: none"> <li>- Primitives points, lignes et surfaces.</li> </ul>	<ul style="list-style-type: none"> <li>- Primitives points, lignes, surfaces et solides.</li> <li>- Géométrie composées : multi-surfaces, multi-solides, etc.</li> </ul>
Topologie	<ul style="list-style-type: none"> <li>- La topologie se limite à des relations entre points, lignes et surfaces.</li> </ul>	<ul style="list-style-type: none"> <li>- Définition explicite de la topologie entre points, lignes, surfaces et solides.</li> </ul>
Attributs	<ul style="list-style-type: none"> <li>- Quelques attributs liés aux primitives géométriques.</li> </ul>	<ul style="list-style-type: none"> <li>- Des attributs structurés dans le domaine urbain.</li> </ul>
Volume	<ul style="list-style-type: none"> <li>- Selon les données.</li> </ul>	<ul style="list-style-type: none"> <li>- Volume important du aux données attributaires.</li> </ul>
Utilisation	<ul style="list-style-type: none"> <li>- Des formats très utilisés : <i>shp</i> (principalement dans des outils SIG) et <i>DWG</i> et <i>DXF</i> (principalement dans des applications DAO).</li> </ul>	<ul style="list-style-type: none"> <li>- Format récent et l'utilisation n'est pas encore généralisée pour des outils SIG ou des outils DAO.</li> </ul>

**Tab10.2** Tableau de comparaison entre les fichiers sources de notre SIG et CityGML

La richesse de ce modèle nous encourage à l'étudier davantage pour pouvoir exploiter des fichiers CityGML dans notre SIG. CityGML4J (CityGML for Java traduit en CityGML pour Java) est une API libre pour parser des fichiers CityGML (pour la lecture, la modification et l'écriture). Cela constitue une perspective très importante de nos travaux de thèse qui consiste à étudier et intégrer cette API dans l'application SIG pour pouvoir intégrer des fichiers CityGML dans la BDS.

## CONCLUSION

Un SIG est une application multidisciplinaire depuis qu'elle fait intervenir une multitude de technologies, d'approches, de concepts, etc. Pour faire un choix ou un autre, nous étions basés sur un ensemble de critères de choix tenant compte de plusieurs aspects. Pour estimer la qualité de nos choix, nous étions amenés à faire un bilan des nos travaux et sur la base de ce bilan nous avons définis un certain nombre de perspectives.

Le bilan portait sur la modélisation du SIG (modélisation des données spatiales, de la BDS et des traitements) et sur l'aspect technique qui portait, à son tour, sur le code ou les programmes développés

dans le cadre de ce travail, l'application SIG en termes de fonctionnalités et l'application Web en termes de technologies mises en œuvre.

L'évaluation a révélé plusieurs possibilités d'amélioration et d'enrichissement du SIG par des nouvelles fonctionnalités et technologies. En termes de fonctionnalités, il n'existe pas de limite pour ajouter des nouveaux services au SIG. Cela dépend essentiellement du contexte d'utilisation, du domaine d'application ainsi que des besoins attendus du SIG. Comme les nouvelles technologies ne cessent d'évoluer et de proposer de nouvelles solutions sur tous les plans (SGBDS, 3D, langages de programmation, outils de développement, etc.), un SIG peut en profiter pour assurer une meilleure performance. Pour ce faire, plusieurs technologies peuvent être utilisées par le SIG pour la mise en place des nouvelles fonctionnalités.



## CONCLUSION GENERALE

Certes les SIG, en particulier les SIG 3D, peuvent être qualifiés d'outils incontournables, voire les plus appropriés, pour la gestion et l'analyse des données géo-référencées. Ils montrent, jour après jour, des progrès considérables en termes de capacités et de fonctionnalités. Cela est dû essentiellement aux avancées réalisées notamment en matière d'outils, de technologies et d'approches mis en œuvre dans le processus de développement des SIG. Le choix de tels outils et de telles technologies dépend essentiellement des besoins et des attentes des utilisateurs du SIG.

Nos travaux de thèse s'inscrivaient dans le domaine des SIG. En effet, la problématique principale consistait à proposer un prototype de SIG 3D accessible via Internet pour la gestion des modèles spatiaux 3D, en particulier, des modèles urbains. Pour ce faire, nos travaux de recherche ainsi que nos contributions portaient sur plusieurs aspects relatifs aux SIG Web 3D. Malgré que nous ne nous basions pas sur un cahier de charges qui indique, *a priori*, précisément les tâches à effectuer, nous avons fixé, quand même, un certain nombre d'objectifs au début de nos travaux de thèse. Ces objectifs n'étaient pas définitifs car ils peuvent évoluer au fur et à mesure de l'avancement du développement du SIG.

Comme le SIG est un domaine multidisciplinaire, nos recherches portaient sur différents concepts dont nous jugeons pertinents dans tout SIG. Si on aborde le concept de données spatiales on doit forcément aborder la notion de modélisation spatiale. Et si on parle du SIG on doit forcément parler des notions de Bases de Données Spatiales (BDS), Système de Gestion de Bases de Données Spatiales (SGBDS), moteur 3D, etc. En effet, tous ces concepts se complètent pour constituer un système cohérent, intégré et surtout opérationnel.

Du côté modélisation spatiale, nous avons étudié une multitude de modèles spatiaux existants ainsi qu'un ensemble d'approches de modélisation 3D. Avant de choisir une approche de modélisation ou une autre, nous avons fixé un ensemble de critères de choix notamment en termes de possibilités de représentation des solides 3D, capacités de stockage, conformité aux normes, etc. Nos choix sont arrêtés sur le couplage, dans le même modèle spatial, de B-Rep (Boundary Representation) en tant qu'une approche surfacique et CSG (Constructive Solid Geometry) en tant qu'approche volumique. Ces choix sont motivés surtout par la simplicité et l'intuitivité d'implémentation de B-Rep et le niveau avancé de réalisme de CSG.

Du côté technologies utilisées, nos choix étaient orientés exclusivement vers les outils libres. Malgré que cela puisse restreindre nos choix, nous avons tenté d'adapter ces technologies aux besoins attendus du SIG Web 3D. Pour ce faire, nous avons choisi le SGBD libre PostgreSQL avec sa "cartouche" spatiale PostGIS en tant que SGBDS pour le SIG. Nous avons étendu les capacités de ce SGBDS pour qu'il puisse stocker des solides 3D, en particulier des bâtiments. Et pour le développement des différentes applications du SIG, nous avons programmé des applets Java. Ce choix est motivé par les capacités du langage Java pour le développement d'application 3D interactives, d'une part, et la facilité d'intégration des applets dans des applications Web, d'autre part.

Du côté fonctionnalités du SIG Web 3D, nous avons suivi une approche ascendante dans laquelle la mise en place des différentes fonctionnalités du SIG s'effectue au fur et à mesure de l'avancement du développement du SIG. En effet, les principales fonctionnalités sont matérialisées par l'intégration des fichiers SIG tels que les fichiers *shapefile* et des fichiers CAO (Conception Assistée par Ordinateur) tels que les fichiers *DXF* et *DWG* dans la BDS, la représentation des données spatiales à travers un système de couches, l'interrogation topologiques et attributaire des modèles spatiaux 3D, etc. Des requêtes graphiques et procédures d'assistance des utilisateurs lors de leur interrogation des BDS sont également mises en place pour faciliter la manipulation des différentes fonctionnalités du SIG. Il est à

noter aussi que toutes les fonctionnalités du SIG Web 3D sont développées intégralement dans le cadre de nos travaux de thèse sans recours à des outils existants.

Vu la diversité des tâches que nous devons accomplir tout au long de nos travaux de thèse, le travail ne peut pas être complet. De ce fait, plusieurs perspectives, notamment en termes d'amélioration de l'application existante ou de son enrichissement par de nouvelles capacités, sont envisageables. Le prototype réalisé peut constituer alors un premier noyau pour les travaux futurs de l'équipe TRIO-LSIIT dans le domaine des SIG 3D.

## REFERENCES BIBLIOGRAPHIQUES

Les références bibliographiques ci-dessous sont organisées en différentes rubriques (ouvrages, thèses de doctorat, journaux, dictionnaires, articles de recherche, tutoriaux, manuels techniques, sites Internet).

Lors de la rédaction, nous étions amenés parfois à enlever quelques parties du rapport de thèse contenant des références bibliographiques. De ce fait, on peut trouver des références bibliographiques mentionnées ci-dessous mais ne sont pas utilisées dans le rapport. Nous considérons que cela ne constitue pas un inconvénient car toutes les références bibliographiques peuvent être pertinentes pour le lecteur.

## 1 OUVRAGES

- [COLLET, 1992] COLLET, C (1992). *Systèmes d'Information Géographique en mode image*. Presses Polytechniques et Universitaires Romandes. ISBN 2-88074-239-0.
- [DENEGRE, *et al*, 1996] Denègre, J., SALGE, F. (1996). *Les systèmes d'information géographique*. Paris, PUF (Coll. « Que sais-je? », no 3122), 128 p. ISBN 2-13-047932-4.
- [DIDIER, *et al*, 1993] DIDIER, M., BOUVERON, C (1993). *Guide économique et méthodologique des SIG*. HERMES Sciences Publication. (21 Septembre 1993). ISBN-10: 2866013840.
- [GALATI, 2006] GALATI, S.R. (2006). *Geographic Information Systems*. Artech House publisher, Septembre, 2006. ISBN-10: 158053533X.
- [J.EGENHOFER, *et al*, 1990] J.EGENHOFER, M., R.HERRING, J (1990). *Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases*. NCGIA Technical Report, N° SBR-9204141.
- [LAURINI, *et al*, 1993] LAURIN, R., MILLERET-RAFFORT, F (1993). *Les bases de données en géomatique*. Hermes Sciences Publicat. (6 septembre 1993). ISBN-10: 2866013727.
- [OSTROMOUKHOV, 2002] OSTROMOUKHOV, V. (2002). *Techniques de rendu non-photoréaliste (NPR)*. François X. Sillion (ed.) Synthèse d'images géographiques, pp. 217-250, Editions Hermes Science, 2002. ISBN 2-7462-0450-9.
- [PANTAZIS, *et al*, 1996] PANTAZIS, D., DONNAY, J.P. (1996). *La conception de SIG, méthode et formalisme*. Hermes Sciences Publication (21 octobre 1996). ISBN-10: 2866015738.
- [REIX, *et al*, 2002] REIX, R., ROWE, F (2002). *La recherche en systèmes d'information : de l'histoire au concept*. Editions VUIBERT-Paris, Introduction de l'ouvrage : Faire de la recherche en systèmes d'information, Paris Juin 2002.
- [ROUET, 1993] ROUET, P. (1993). *Les données dans les Systèmes d'Information Géographiques*. Edition HERMES publication. Paris, 1993. ISBN- 9782866012953.
- [SERVIGNE, *et al*, 2006] SERVIGNE, S., LIBOUREL, T. (2006). *Fondements des bases de données spatiales*. ISBN 2-7462-1378-8. Lavoisier, 2006.
- [VAN DAM, *et al*, 1991] VAN DAM, F., HUGHES, F (1991). *Computer Graphics, Principles and Practice*. The systems programming series. ISBN 0-201-84840-6. Addison-Wesley, 1991.

## 2 THESES DE DOCTORAT

- [DE LA LOSA, 2000] DE LA LOSA, A. (2000) *Modélisation de la troisième dimension dans les bases de données géographiques*, Thèse de doctorat, Université Marne-La-Vallée, 175 pages.
- [HELALI, 2001] HELALI, H. (2001). *Design and implementation of a Web GIS for the city of tehran*. MSc thesis. Department of geodesy and geomatics Engineering K.N.Toosi University of technology, tehran, Iran.
- [KUFONIYI, 1995] KUFONIYI, O. (1995). *Spatial coincidence modelling, automated database updating and data consistency in vector GIS*. PhD thesis, ITC, The Netherlands, cited in Zlatanova, S, (2000), 3D GIS for Urban Development, ITC Dissertation Series No. 69.
- [PILOUK, 1996] PILOUK, M. (1996) *Integrated modelling for 3D GIS*. Ph.D. thesis, ITC, The Netherlands, 200 pages.
- [RAMOS, 2003] RAMOS, F. (2003). *Modélisation et validation d'un Système d'Information Géographique opérationnel*. Thèse de doctorat présentée le 5 Mai 2003. Université Marne-La-Vallée.
- [ZLATANOVA, 2000] Zlatanova, S. (2000). *3D GIS for Urban Development*. PhD Thesis, ITC, The Netherlands, 222 p.

## 3 JOURNAUX

- [DENEGRE, *et al*, 1997] DENEGRE, J., SALGE, F. (1997). *Les systèmes d'information géographique*. 2nde éd., Paris, Presses Universitaires de France, 128 p.

- [GOODCHILD, *et al*, 1997] GOODCHILD, M., JEANSOULIN, R. (1997) *Data quality in geographic information, From error to uncertainty*. Editions HERMES. 12-1997.
- [INFORMATICA, 2008] INFORMATICA (2008). *Des données de qualité. Exploitez le capital de votre organisation*. © Informatica Corporation, 2008.
- [PINET, *et al*, 2003] PINET, F., LBATH, A., VIGIER, F., SCHNEIDER, M. (2003). *Conception dse Systèmes d'Information à Référence Spatiale (SIRS)*. Communicants pour les agrosystèmes Vers une approche méthodologique adaptée aux spécificités et aux nouveaux besoins du secteur agricole. Ingénieries – Eau, Agriculture, Territoires, n° spécial Technologies pour les agrosystèmes durables, p. 143 – 155.
- [WANG, *et al*, 2001] WANG, D., CHENG, T. (2001). *A spatio-temporal data model for activity-based transport demand modeling*. International Journal of Geographical Information System 15(6): 561-585.
- [WORBOYS, *et al*, 1990] WORBOYS, M.F., HEARNshaw, H.M., MAGUIRE, D.J. (1990). *Object-Oriented Data Modelling for Spatial Databases*. International Journal of Geographical Information Systems, 1990, Vol. 4, No. 4, 369–383.

#### 4 DICTIONNAIRES

- [GIRAUD *et al*, 1974] GIRAUD, J., PAMART, P., RIVERAIN, J.R. (1974). *Nouveaux mots dans le vent*. LAROUSSE, Paris 1974.
- [LAROUSSE, 2008] LAROUSSE (2008). *Le petit Larousse illustré, dictionnaire de la langue française*. ISBN 978-2-03-582502-5.
- [ROBERT, 2008] ROBERT (2008). *Le nouveau Petit Robert de la langue française*. © Getty Images-Dennis Cody, 2008.
- [SENSAGENT, 2010] SENAGENT (2010). Dictionnaire et traducteur – définition, synonymes et encyclopédies. (en ligne) <http://dictionnaire.sensagent.com/moteur+de+rendu+3d/fr-fr>. (Page consultée le 9 novembre 2010).

#### 5 ARTICLES DE RECHERCHE

- [AL MAHMOUD, 2004] AL MAHMOUD, A.M.H. (2004). *Geographical Information Systems and digital models*. ISPRS 2004 International Society for Photogrammetry and Remote Sensing / Sociedad Internacional de Fotogrametría y Teledetección. Edición XX. ISPRS 2004, Estambul / Istanbul, Turquie. 12-23 Julio de 2004.
- [AVIGNON, 2004] AVIGNON, M. (2004). *Acquérir des données spatiales : quels types de données et de capteurs en réponse à quels types de besoins*. Actes du séminaire national - Les sciences de la vie et de la Terre au XXIème siècle : enjeux et implications 15 et 16 décembre 2004.
- [BEDARD, *et al*, 1989] BEDARD, Y., PAQUETTE, F. (1989). *Extending Entity/Relationship formalism for spatial information systems*. The International Symposium on Computer-Assisted Cartography, April 2 - 7, 1989, Baltimore, Maryland, USA.
- [BEDARD, *et al*, 1992] BEDARD, Y., PAGEAU, J., CARON, C. (1992). *Spatial Data Modeling: The Modul-R Formalism and CASE Technology*. ISPRS Symposium, 10p., August 1-14, 1992, Washington, USA.
- [BERNARD, *et al*, 2000] BERNARD, G., RAMOS, F., LEBARD, A. (2000). *Rapport de synthèse sur l'analyse des différents modèles 3D*. EADS S&DE, Geomatics, S3D/RPT, Juin, 2000.
- [BOHLEN, *et al*, 1998] BOHLEN, M., JENSEN, C.S. (1998). *Spatio-Temporal Database support for Legacy*. ACM Symposium on Applied Computing, Atlanta, Georgia, 1998.
- [BLINDENBERG, *et al*, 2007] BLINDENBERG, V., ROUHEN, C., MALHERBE, A., YOUNSI, N (2007). *La 3D résout le mystère de la Grande Pyramide*. 9<sup>èmes</sup> Rencontres Internationales de la Réalité Virtuelle du 18 au 22 avril 2007. Laval, France.
- Virtuelle du 18 au 22 avril 2007.
- [BONAN, *et al*, 2002] BONAN, L., GUORAY, C. (2002). *A general Object-Oriented approach spatial temporal data model*. Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002.
- [BRENNER, 2004] BRENNER, C. (2004). *Modeling 3D objects using weak CSG primitives*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences ISPRS 2004, Istanbul. Vol. 35.
- [BROOME, *et al*, 1990] BROOME, F., MEIXLER, D. *The TIGER Data Base Structure*. Cartography and Geographic Information Systems, Publié par Cartography and Geographic Information Society (Volume 17, N° 1). Janvier 1990.

- [CLEMENTINI, *et al*, 2008] CLEMENTINI, E., LAURINI, R. (2008). *Un cadre conceptuel pour modéliser les relations spatiales*. Revue des Nouvelles Technologies de l'Information RNTI-E-14():1-17, ISSN 978.2.85428.86, Novembre 2008.
- [COORS, 2002] COORS, V. (2002). *3D GIS in Networking environments*, pp. 345–357. Elsevier, Amsterdam (2003) (Special Issue 3D Cadastre).
- [DUBOIS *et al*, 2000] DUBOIS, E., NIGAY, L., TROCCAZ, J.e (2000). *Combinons le monde virtuel et le monde réel : Classification et principes de conception*. Acte des Rencontres Jeunes Chercheurs en IHM, Île de Berder, France, Mai 2000, p. 31-35.
- [ELLUL, *et al*, 2008] ELLUL, C., MORDECHAI, H. (2008). *Using a B-Rep Structure to Query 9-Intersection Topological Relationships in 3D GIS – Reviewing the Approach and Improving Performance*. Published in 3D Geo-Information Sciences (Lecture Notes in Geoinformation and Cartography) edited by Jiyeong Lee and Sisi Zlatanova, December 2008.
- [FLEMING, 1999] FLEMING, B. (1999). *3D modeling & surfacing*. Morgan Kaufmann Publisher. ISBN- 0-12-260490-3.
- [GERBE, 2002] GERBE, P. (2002). *Outils, données et applications de l'information géographique en 3D*. GeoEvennement, Paris, 9-11 Avril 2002.
- [GUERCKE, *et al*, 2009] GUERCKE, R., BRENNER, C. (2009). *A Framework for the Generalization of 3D City Models*. Proceedings of 12th AGILE Conference on GIScience, Hannover, Germany , 2009.
- [HADZILACOS, *et al*, 1997] HADZILACOS, T., TRYFONA, N. (1997). *An Extended Entity-Relationship Model for Geographic Applications*. SIGMOD Record, vol.26, N°3, 1997.
- [HERRING, 1987] J. HERRING (1987). *TGRIS: Topologically Integrated Geographic Information System*. Proceedings of the 7<sup>th</sup> International Symposium On computer Assisted Cartography (Auto-Carto 8), Baltimore, MD, 1987.
- [HOEL, *et al*, 2003] HOEL, E, MENON, S., MOREHOUSE, S. (2003). *Building a Robust Relational Implementation of Topology*. 8th International Symposium on Spatial and Temporal Databases – SSTD '03. Santorini island, Greece, 2003.
- [HUGH, 1996] HUGH, W.C. (1996). *Entity-Relationship modeling of spatial data for geographic information systems*. International Journal of Geographical Information Systems, January 1996.
- [JARROUSH, *et al*, 2004] JARROUSH, J., EVEN-TZUR, G. (2004). *Constructive Solid Geometry as the Basis of 3D Future Cadastre*. FIG Working Week 2004 Athens, Greece, May 22-27, 2004.
- [JIANG, *et al*, 2006] JIANG, B., ITZHAK, O. (2006). *Spatial Topology and its Structural Analysis based on the Concept of Simplicial Complex*. 9th AGILE Conference on Geographic Information Science, Visegrád, Hungary, 2006.
- [KALAY, 1982] KALAY, Y. E. (1982). *Determining the spatial containment of a point in general polyhedra*. *Comput. Graphics Image Process.* 19 (1982), 303-334.
- [KHUAN, *et al*, 2007] KHUAN, C.T., ABDUL-RAHMAN, A., ZLATANOVA, S. (2007). *New 3D data type and topological operations for Geo-DBMS*. In: (eds.) Coors, Zlatanova, Rumor, and Fendel. Annual UDMS, Taylor and Francis Group, London. ISBN 978-0-415-44059-2.
- [KHUAN, *et al*, 2008.a] KHUAN, C.T., ABDUL-RAHMAN, A., ZLATANOVA, S. (2008). *3D SPATIAL OPERATIONS FOR GEO-DBMS: GEOMETRY VS. TOPOLOGY*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008.
- [KHUAN, *et al*, 2008.b] KHUAN, C.T., ABDUL-RAHMAN, A. (2008). *Implementing 3D geometrical data type and spatial operations for postgis*. In The Proceedings of the 3rd International Workshop on 3D Geo-information, pages 17–35, Seoul, South Korea, 2008.
- [KIM, *et al*, 2009] KIM, J.S., KANG, H.Y., LEE, T.H., LI, K.J. (2009). *Topology of the Prism Model for 3D Indoor Spatial Objects*. Proceedings of the International Workshop on Indoor Spatial Awareness 2009, May 18, Taipei Taiwan.
- [KOONTZ, 2003] KOONTZ, L. D. (2003). *Geographic Information Systems: Challenges to Effective Data Sharing*. Washington, D.C.: General Accounting Office, Report GAO-03-874T. 2003. (En ligne) <http://www.gao.gov/new.items/d03874t.pdf>.
- [KOUSSA, *et al*, 2009] KOUSSA, C., KOEHL, M. (2009). *A simplified geometric and topological modeling of 3D buildings: combination of surface-based and solid-based representations*. ASPRS Annual Conference, Baltimore Maryland, United States of America. 9-13 Mars 2009.

- [LACHANCE, *et al*, 2006] LACHANCE, B., BEDARD, K., POULIOT, J., KIRKWOOD, D. (2006). *Design and experiment of a topological structure adapted to tetrahedral modeling*. Proceedings of the 26th gOcad Meeting, Nancy, France, 6-9 juin 2006. 15 pages.
- [LAFERTE, *et al*, 2009] LAFERTE, J.M., DAUSSIN, G., FLIFLA, J. (2009). *Modélisation de linéaires par des NURBS pour un simulateur de vols*. TAIMA'09: Traitement et Analyse de l'Information. Hammamet, Tunisie — 04-09 mai 2009.
- [LANE, *et al*, 1984] LANE, J., MAGEDSON, B., RARICK, M. (1984). *An efficient point in polyhedron algorithm*. *Comput. Vision Graphics Image Process.* 26 (1984), 118-125.
- [LARRIVEE, *et al*, 2006] LARRIVEE, S., BEDARD, Y., POULIOT, J (2006). *Fondement de la modélisation conceptuelle des bases de données géospatiales 3D*. *Revue internationale de géomatique*, Vol. 16, No. 1, pp. 9-28.
- [MAREDJ, *et al*, 2001] MAREDJ, A., SAIB, A., DJERMOUNI, A (2001). *Intégration des données multimédias dans les Systèmes d'Information Géographique*. RIST VOL. 11 N°02, ISSN-1111-0015. 2001.
- [MOLENAAR, 1990] MOLENAAR, M. (1990). *A formal data structure for 3D vector maps*. Proceedings of EGIS'90, Amsterdam, The Netherlands, pp.780-781.
- [MOREHOUSE, 1985] MOREHOUSE, S. (2003). *ARC/INFO: A Geo-Relational Model for Spatial Information*. Proceedings of the 7<sup>th</sup> International Symposium On computer Assisted Cartography (Auto-Carto 7), Washington, DC, Mars 1985.
- [PAUL, *et al*, 1997] PAUL, S., DUCHER, G., JOBARD, I., LATARCHE, C.H., LENCO, M. (1997). *Terminologie en télédétection et photogrammétrie* (Français-Anglais). CIFL, PUF, Paris, 455 p. ISBN 2-85319-270-9.
- [PFUND, 2001] PFUND, M. (2001). *Topologic data structure for a 3D GIS*. Proceedings of ISPRS, May 23-25, Bangkok, Thailand, Vol. 34, Part 2W2, pp. 233-237.
- [PINET, *et al*, 2003] PINET, F., LBATH, A., VIGIER, F., SCHNEIDER, M. (2003). *Conception de Systèmes d'Information à Référence Spatiale (SIRS)*. *Communicants pour les agrosystèmes Vers une approche méthodologique adaptée aux spécificités et aux nouveaux besoins du secteur agricole*. Ingénieries – Eau, Agriculture, Territoires, n° spécial Technologies pour les agrosystèmes durables, p. 143 – 155.
- [PLANCHER, *et al*, 2007] PLANCHER, G., NICOLAS, S., PIOLINO, P. (2007). *La Réalité Virtuelle pour la Mémoire Episodique*. 2<sup>ème</sup> journée de l'AFRV (Association Française de la Réalité Virtuelle, Augmentée, Mixte et d'Interaction 3D). 25, 26 Octobre 2007, Marseille.
- [QINGQUAN, *et al*, 2000] QINGQUAN, L., DEREN, L. (2000). *Algorithms for Tetrahedral Network (TEN) Generation*. *Geo-spatial Information Science* Vol.3, No. 1,p. 11~16, March 2000.
- [RAGHAVAN, *et al*, 2002] RAGHAVAN, V., SANTITAMONT, P., MASUMOTO, S., HONDA, K. (2002). *Implementing Web GIS Applications using Open Source Software*. *Map Asia 2002*. (En ligne) <http://www.gisdevelopment.net/technology/gis/techgi0062pf.htm>.
- [RAMOS, 2002] RAMOS, F. (2002). *A multi-level Approach for 3D modelling in Geographical Information System*. *Symposium Geospatial Theory, Processing and Applications*. Ottawa, 2002.
- [RENOLEN, 2000] RENOLEN, A. (2000). *Modeling the Real World: Conceptual Modeling in Spatiotemporal Information System design*. *Transactions in GIS*, Volume 4, Issue 1, pages 23–42, January 2000.
- [RUSSO, 2006] RUSSO, M. (2006). *Polygonal Modeling, Basic and Advanced Techniques*. Editeur: Jones & Bartlett Publishers, 2006. ISBN: 1598220071, 9781598220070.
- [SACCHI, *et al*, 1994] SACCHI, C., SBATELLA, L. (1994). *An object-Oriented approach to spatial databases*. Proceedings of the Fifth European Conference and Exhibition on Geographic Information Systems, EGIS '94. Utrecht: EGIS Foundation, 1994. 2:1204-1213.
- [SCHÖN, *et al*, 2008] SCHÖN, B., LAEFER, D.F., MORRISH, S.W., BERTOLOTTO, M. (2008). *Three-Dimensional Spatial Information Systems: State of the Art Review*. © 2009 Bentham Science Publishers Ltd. *Recent Patents on Computer Science*, 2009, Vol. 2, No. 1
- [SOOMRO, *et al*, 1999] SOOMRO, T. R., ZHENG, K., Pan, Y. (1999). *HTML and Multimedia Web GIS*. Proceedings of the 3rd International Conference on Computational Intelligence and Multimedia Applications, September 23-26, 1999, pp. 371-382.
- [TRYFONA, *et al*, 1999] TRYFONA, N., JENSEN, C.S. (1999). *Conceptual Data Modeling for Spatiotemporal Applications*. *Geoinformatica* Volume 3, Number 3, 245-268, DOI: 10.1023/A:1009801415799.
- [WEI, *et al*, 1998] WEI, G., PING, Z., JUN, C. (1998). *Topological Data Modelling for 3D GIS*. *The International Archives of Photogrammetry and Remote Sensing*, XXXII (4). D. Fritsch, M, English & M.

Sester, eds, 'IAPRS', Vol. 32/4, ISPRS Commission IV Symposium on GIS – Between Visions and Applications, Stuttgart, Germany, 1998.

- [WEN, *et al*, 2005] WEN, A.S., SHAMSUDDIN, H.J.S.M, SAMIAN, Y. (2005). *Ship Hull Fitting using NURBS*. Proceedings of the Computer Graphics, Imaging and Vision: New Trends (CGIV'05) 0-7695-2392-7/05 \$20.00 © 2005 IEEE.
- [XIANFENG, *et al*, 2004] XIANFENG, S., YASUYUKI, K., MAMORU, S. (2004). *The development of Web mapping application using open source GIS solution*. International Symposium on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences. Hanoi, Vietnam, 16-18 September 2004.
- [YANBING, *et al*, 2007] YANBING, W., LIXIN, W., WENZHONG, S., XIAOMENG, L. (2007). *On 3D GIS spatial modeling*. ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs 237. August 28-29, 2007 Urumchi, Xingjiang, China.
- [ZLATANOVA, *et al*, 1996] ZLATANOVA, S., PILOUK, M., TEMPFLI, K. (1996). *Building reconstruction from aerial images and creation of 3D topologic data structure*. Proceedings of IAPR / TC-7 Workshop, 2-3 September 1996, Graz, Austria.
- [ZLATANOVA, *et al*, 2002] ZLATANOVA, S., ABDUL-RAHMAN, A., Shi, W. (2002). *Topology for 3D spatial objects*. International Symposium and Exhibition on Geoinformation 2002, 22-24 October, Kuala Lumpur, Malaysia. Kluwer Academic Publishers. ISSN:1384-6175.

## 6 TUTORIAUX

- [BALESTAT, *et al*, 1996] BALESTAT, M.F., MARATREY, J.P. (1996). *Les systèmes de coordonnées*. (En ligne) [http://www.astrosurf.com/quasar95/exposes/systemes\\_de\\_coordonnees.pdf](http://www.astrosurf.com/quasar95/exposes/systemes_de_coordonnees.pdf) (Page consultée le 19 octobre 2010).
- [BOUCHER, 2008] BOUCHER, C. (2008). *Système de coordonnées-Systèmes de références*. 4e Ecole d'Été GRGS Forcalquier.
- [BRUSLE, 2008] BRUSLE, L. (2008). *Construction d'un SIG et champs d'application De la mise en forme des données géographiques aux divers champs de la géomatique, De la mise en forme des données géographiques aux divers champs de la géomatique*. (En ligne) [http://laeti.perrierbrusle.free.fr/SIG\\_2.pdf](http://laeti.perrierbrusle.free.fr/SIG_2.pdf) (Page consulté le 20 octobre 2010).
- [IOSIFESCU, *et al*, 2010] IOSIFESCU, I., HUGENTOBLE, M., HURNI, L. (2010). *GIScience 2010 Tutorial: Development of Web GIS and Web Mapping Applications with QGIS mapserver*. Sixth international conference on Geographic Information Science Zurich, Switzerland, 14th of September 2010.
- [THALMANN, *et al*, 2003] THALMANN, D., GOBRON, S. (2003). *Le rôle de l'informatique graphique et ses applications*. (En ligne) <http://vrlab.epfl.ch/~thalmann/infographie/part1.pdf>. (Page consultée le 01 Décembre 2010).
- [TUDOR, 2006] TUDOR, H. (2006). *Guide de conception orientée-objet*. (En ligne) [http://www.citi.tudor.lu/SI/Livvable.nsf/6664D938C3441E57C125702D00503A9D/\\$File/GU\\_Guide%20de%20Conception%2000\\_1.0.pdf](http://www.citi.tudor.lu/SI/Livvable.nsf/6664D938C3441E57C125702D00503A9D/$File/GU_Guide%20de%20Conception%2000_1.0.pdf) (Page consultée le 19 octobre 2010).

## 7 MANUELS TECHNIQUES

- [COLLADA, 2008] COLLADA (2008). *Digital Asset Schema Release 1.5.0 Specification*. April 2008. Editors: Mark Barnes and Ellen Levy Finch, Sony Computer Entertainment Inc.
- [E.KOLB, 1994] E.KOLB, C. (1994). *Rayshade User's Guide and Reference Manual*. 1994. (En ligne) <http://www.graphics.stanford.edu/~cek/rayshade/doc/guide/guide.html>.
- [ESRI, 1998] ESRI (1998). *ESRI Shapefile Technical Description*. An ESRI White Paper, July 1998.
- [ESRI, 2005] ESRI (2005) An ESRI white paper. Juillet 2005, Etats Unis.
- [GEUZAIN *et al*, 2009] GEUZAIN, C., REMACLE, J.F. (2009). *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. Gmsh Reference Manual. Version 2.4.2, November 11, 2009.
- [INSPIRE, 2008] INSPIRE Drafting Team "Data Specifications" (2008). *INSPIRE Infrastructure for Spatial Information in Europe. Definition of Annex Themes and Scope*. 18 Mars 2008.
- [ISO, 1999] ISO TC 211/WG 2 (1999). *Geographic Information – Spatial Schema. Technical Report, Second Draft of ISO 19107*. International Organization for Standardization, 1999.
- [ISO, 2003] ISO TC 211 (2003) *Geographic Information – Spatial Schema*.
- [OGC, 2007] OGC (2007). *Implementation Specification: Geographic information - Geography Markup Language Version 3.2.1*. Reference number of this document: OGC 07-061 Version: 3.2.1 Category:

OpenGIS® IS Revision Notes Editor: Clemens Portele, Project leader ISO 19136 Revision Notes for OpenGIS®, 2007.

- [OGC, 2008] OGC (2008). *OpenGIS® City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium Inc. Date: 2008-08-20 Reference number of this OGC® project document: OGC 08-007r1 Version: 1.0.0 Category: OpenGIS® Encoding Standard Editors: Gerhard Gröger, Thomas H. Kolbe, Angela Czerwinski, Claus Nagel.

## 8 SITES INTERNET

- [ABACOGROUP, 2009] ABACOGROUP (2009). (En ligne) <http://www.abacogroup.com> (Page consultée le 10 octobre 2009).
- [ANTOINE, et al, 2002] ANTOINE, A., DELLEA-WARIN, P., LESQUOY-FISCHER, K. (2002). *Le système documentaire*. (En ligne) <http://www.univ-nancy2.fr/Amphis/images/films/Gest-Qual-SystemeDocumentaire.pdf> (Page consultée le 14 octobre 2010).
- [APIA, 2009] APIA (2009). *Apia antica Project*. (En ligne) [http://www.appia.itabc.cnr.it/appia\\_3d.php](http://www.appia.itabc.cnr.it/appia_3d.php) (Page consultée le 10 octobre 2009).
- [BALDE, 2007] BALDE, I. (2007). *Mise en place d'une plateforme de cartographie dynamique*. (En ligne) [http://www.memoireonline.com/07/08/1287/m\\_mise-en-place-d-une-plate-forme-de-cartographie-dynamique8.html](http://www.memoireonline.com/07/08/1287/m_mise-en-place-d-une-plate-forme-de-cartographie-dynamique8.html) (Page consultée le 14 octobre 2010).
- [BERCYCOLLOC, 2010] BERCYCOLLOC (2010). *Optimiser les Systèmes d'Information*. (En ligne) [http://www.colloc.bercy.gouv.fr/colo\\_otherfiles\\_fina\\_loca/docs\\_som/axe5\\_1.pdf](http://www.colloc.bercy.gouv.fr/colo_otherfiles_fina_loca/docs_som/axe5_1.pdf) (Page consultée le 17 mai 2010).
- [BRANDEL, 2009] BRANDEL, S. (2009). *Modélisation géométrique*. (En ligne) <http://www710.univ-lyon1.fr/~sbrandel/enseignement/gamagora/supports/01-modeling.pdf> (Page consultée le 17 mai 2010).
- [CODEHAUS, 2009] CODEHAUS (2009). *GeoTools Users Guide, Point Set Theory and the DE-9IM Matrix*. [En ligne] <http://docs.codehaus.org/display/GEOTDOC/Point+Set+Theory+and+the+DE-9IM+Matrix> (Page consultée le 17 mai 2010).
- [CNRTL, 2010] CNRTL (2010). *Centre Nationale de Ressource Textuelle et Lexicale*. (En ligne) <http://www.cnrtl.fr> (Page consultée le 17 mai 2010).
- [DUBUISSON, 2007] DUBUISSON, S (2007). *Modélisation des objets solides Algorithmique Graphique et Modélisation*. (En ligne) [http://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master\\_esj\\_2006\\_2007/IMG/pdf/ModelisationSolides.pdf](http://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master_esj_2006_2007/IMG/pdf/ModelisationSolides.pdf) (Page consultée le 17 mai 2010).
- [ESRI, 2009] ESRI (2009). (En ligne). <http://gis.esri.com/library/userconf/proc99/proceed/papers/pap282/p282.htm> (Page consultée le 10 octobre 2009).
- [ESRIFRANCE, 2010] ESRIFRANCE (2010). *Les solutions SIG et cartographiques*. (En ligne) <http://www.esrifrance.fr> (Page consultée le 01 juin 2010).
- [FOURMIROUGE, 2010] FOURMIROUGE (2010). *Dossier Les outils de l'animation 2D/3D/Web : Témoignages sur le rendu Non Photo-Réaliste*. (En ligne) [http://www.fourmirouge.org/FP/px/PX59H\\_5.html](http://www.fourmirouge.org/FP/px/PX59H_5.html) (Page consultée le 17 mai 2010).
- [FREEGIS, 2010] FREEGIS (2010). *The FreeGIS project*. (En ligne) <http://freegis.org> (Page consultée le 26 mai 2010).
- [G3DGMV, 2010] G3DGMV (2010). *3D Graphical Map Viewer*. (En ligne) <http://g3dgmv.sourceforge.net> (Page consultée le 26 mai 2010).
- [GDF HANNOVER, 2005] GDF HANNOVER (2005). *GDF Hannover bR*. (En ligne) [http://www.gdf-hannover.de/lit\\_html/grass60\\_v1.2\\_fr/node74.html](http://www.gdf-hannover.de/lit_html/grass60_v1.2_fr/node74.html). (Page consultée le 18 septembre 2010).
- [GEOSERVER, 2010] GEOSERVER (2010). *GeoServer*. (En ligne) <http://geoserver.org> (Page consultée le 26 mai 2010).
- [GEOTOOLS, 2010] GEOTOOLS (2010). *GeoTools The Open Source Java GIS Toolkit*. (En ligne) <http://www.geotools.org> (Page consultée le 26 mai 2010).
- [GLOBALMAPPER, 2010] GLOBALMAPPER (2010). *US GIS Maps - 3D Map Software - Web GIS Software*. (En ligne) <http://www.globalmapper.com/product/screenshots.htm> (Page consultée le 01 juin 2010).
- [GMT, 2010] GMT (2010). *The general mapping tools*. (En ligne) <http://gmt.soest.hawaii.edu> (Page consultée le 26 mai 2010).
- [GOOGLEMAPS, 2010] GOOGLEMAPS (2010). (En ligne) <http://maps.google.fr> (Page consultée le 31 mai 2010).
- [GVSIG, 2010] GVSIG (2010). *Portal GvSIG*. (En ligne) <http://www.gvsig.org/web>
- [IGN, 2008] IGN (2008). *Systèmes de coordonnées*. (En ligne) [http://education.ign.fr/DISPLAY/000/527/312/5273120/geodesie\\_coordonnees.pdf](http://education.ign.fr/DISPLAY/000/527/312/5273120/geodesie_coordonnees.pdf) (Page consultée le 19 octobre 2010).

- [IGN, 2010] IGN (2010). *Et la géographie prend vie*. (En ligne) <http://www.ign.fr> (Page consultée le 26 mai 2010).
- [IRD, 2010] Institut de Recherche pour le Développement (2010). *Appui méthodologique, Procédures et standards*. (En ligne) [http://www.ird.fr/informatique-scientifique/methodo/standards/normes\\_iso\\_ogc/iso\\_geo](http://www.ird.fr/informatique-scientifique/methodo/standards/normes_iso_ogc/iso_geo). (Page consultée le 11 octobre 2010).
- [JUMP, 2010] JUMP (2010). *JUMP-Project.org*. (En ligne). <http://www.jump-project.org>
- [MAINSTREET, 2009] MAINSTREET (2010). *Geographic Information System- GIS software development and consulting*. (En ligne) <http://www.mainstreetgis.com> (Page consultée le 26 mai 2010).
- [MAPFISH, 2010] MAPFISH (2010). *Map Fish*. (En ligne) <http://www.mapfish.org> (Page consultée le 26 mai 2010).
- [MAPGUIDE, 2010] MAPGUIDE (2010). *MapGuide project home*. (En ligne) <http://mapguide.osgeo.org> (Page consultée le 31 mai 2010).
- [MAPSERVER, 2010] MAPSERVER (2010). *Open source Web mapping*. (En ligne) <http://mapserver.org> (Page consultée le 26 mai 2010).
- [MARANZANA, 2007] MARANZANA, R (2007). *Programmation des systèmes de CAO pour l'aéronautique, modélisation géométrique*. (En ligne) [http://www.gpa.etsmtl.ca/cours/mga802/Documents/BRep\\_Topology\\_6.pdf](http://www.gpa.etsmtl.ca/cours/mga802/Documents/BRep_Topology_6.pdf) (Page consultée le 17 mai 2010).
- [MASTERSIG, 2005] MASTERSIG (2005). *Grands types de solutions logicielles*. (En ligne) <http://mastersig.free.fr/wiki/doku.php?id=logiciels> (Page consultée le 21 octobre 2010).
- [MATHWORLD, 2010] MATHWORLD (2010). *Topology from Wolfram MathWorld*. (En ligne) <http://mathworld.wolfram.com/Topology.html> (Page consultée le 17 mai 2010).
- [MST, 2004] MST (2004). *Brief history of geographical information systems*. (En ligne) <http://web.mst.edu/~rogersda/gis/History%20of%20GIS.pdf> (Page consultée le 03 juin 2010).
- [MULTIMANIA, 2010] MULTIMANIA (2010). *La 3D : définition, notion de perceptive et histoire*. (En ligne) <http://membres.multimania.fr/tpe3d/def-fonc-hist.html> (Page consultée le 17 mai 2010).
- [NATIONALGEOGRAPHICS, 2010] NATIONALGEOGRAPHICS (2010). (En ligne) <http://maps.nationalgeographic.com> (Page consultée le 31 mai 2010).
- [OPEN3DGIS, 2009] OPEN3DGIS (2009). (En ligne) <http://www.open3dgis.org> (Page consultée le 10 octobre 2009).
- [OPENSOURCEGIS, 2008] OPENSOURCEGIS (2008). *Open Source GIS*. (En ligne) <http://opensourcegis.org> (Page consulté le 21 octobre 2010).
- [ORACLE, 2009] Oracle (2009). *Site officiel d'oracle*. (En ligne) <http://www.oracle.com> (Page consultée le 17 mai 2010).
- [PAGESJAUNES, 2010] PAGESJAUNES (2010). (En ligne) <http://www.pagesjaunes.fr>
- [PBMAPINFO, 2010] PBMAPINFO (2010). *Pitney Bowes MapInfo*. (En ligne) <http://www.pbmapinfo.eu> (Page consultée le 31 mai 2010).
- [PHPMYVISITES, 2010] PHPMYVISITES (2010). *Dico Info* (En ligne) <http://dictionnaire.phpmyvisites.net/definition-3d--6261.htm> (Page consultée le 17 mai 2010).
- [SHADOWS, 2010] SHADOWS (2010). *Histoire de la 3D*. (En ligne) <http://www.shadows.fr/3d/chronologie> (Page consultée le 17 mai 2010).
- [STAR-APIC, 2010] STAR-APIC (2010). (En ligne) <http://www.star-apic.com> (Page consultée le 17 mai 2010).
- [SUN, 2010] SUN (2010). *Sun Developer Network*. (En ligne) <http://java.sun.com/applets> (Page consultée le 15 septembre 2010).
- [TECHNOSCIENCE, 2010] TECHNOSCIENCE (2010). *Espace topologique*. (En ligne) <http://www.techno-science.net/?onglet=glossaire&definition=5231> (Page consultée le 17 mai 2010).
- [THEFREEDICTIONARY, 2010] THEFREEDICTIONARY (2010). *Geometric primitive*. (En ligne) [http://encyclopedia.thefreedictionary.com/Primitive+\(geometry\)](http://encyclopedia.thefreedictionary.com/Primitive+(geometry)) (Page consultée le 17 mai 2010).
- [VIAMICHELIN, 2010] VIAMICHELIN (2010). (En ligne) <http://www.viamichelin.fr> (Page consultée le 31 mai 2010).
- [VOLLE, 2004] VOLLE, M. (2004). *A propos de la modélisation*. (En Ligne) <http://www.volle.com/travaux/modelisation2.htm> (Page consultée le 17 mai 2010).
- [WALLONIE, 2010] WALLONIE (2010). *Aménagement et urbanisme-Géomatique, applications WebGIS*. (En ligne) <http://carto6.wallonie.be> (Page consultée le 10 octobre 2009).
- [WEBGIS.NET, 2010] WEBGIS.NET (2010). (En ligne) <http://www.webgis.net> (Page consultée le 21 octobre 2010).



## LISTE DES PUBLICATIONS

- KOEHL, M. and KOUSSA, C. (2008). *SIG 3D Et 3D dans les SIG : Application aux modèles patrimoniaux*. GéoÉvénement 2008. 13 May 2008. Paris, France.
- KOUSSA, C. and KOEHL, M. (2009). *A simplified geometric and topological modeling of 3D buildings enriched by semantic data: combination of surface-based and solid-based representations*. GIS Ostrava 2009. 25-28 Janvier 2009, Ostrava, République Tchèque.
  - ➔ Cet article a été retenu pour être publié dans le livre «*The Advances in Geoinformation Technologies 2009*», pp : 25-41. ISBN 978-80-248-2145-0.
- KOUSSA, C. and KOEHL, M. (2010). *Proposal of 3D GIS for spatial data visualization and analysis over Internet*. ASPRS Annual Conference. San Diego, California, United States of America. April 26-30, 2010.
- KOUSSA, C. and KOEHL, M. (2010). *Implementation of a 3D GIS in Internet environment*. ISPRS proceedings. 5<sup>th</sup> 3D GeoInfo Conference, pp 129-134. 3-5 November 2010, Berlin, Germany. ISSN 1682-1750.



## ANNEXES

## 1 ANNEXE 1 : DONNEE SPATIALES

## 1.1 NORMES ISO POUR L'INFORMATION SPATIALE

La (Fig1) montre un schéma illustratif des normes ISO (International Organisation of Standardisation) concernant les données géographiques [IRD, 2010].

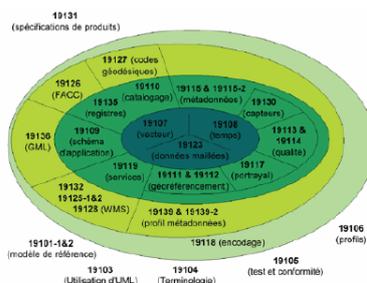


Fig 1. Schéma illustratif des normes ISO concernant les données géographiques

Ci-dessous, les différentes normes spatiales de l'ISO [IRD, 2010] sont représentées en termes d'identifiant ISO et description du rôle de la norme.

- ISO 6709 (Standard representation of geographic point location by coordinates): Représentation normalisée des latitudes, longitude et altitude pour la localisation des points géographiques
- ISO 19101 - ISO 19101-2 (Geographic information -- Reference model): Première norme de la série ISO 19100 : définition d'un cadre structurant pour la représentation de l'information géographique sous forme numérique la partie 2 de cette norme concerne l'imagerie.
- ISO 19103 (Geographic information -- Conceptual schema language): Définition des règles de modélisation des schémas conceptuels de données utilisés dans la série de normes ISO 19100. le langage de modélisation est UML(3).
- ISO 19105 (Geographic information -- Conformance and testing): Description des tests et des critères à remplir pour pouvoir prétendre être conforme avec la série des standards ISO/TC 211.
- ISO 19106 (Geographic information – Profiles): Description de méthodes pour créer des profils (restrictions et extensions) des schémas de données des normes ISO 19100 suivants les besoins des utilisateurs.
- ISO 19107 (Geographic information -- Spatial schema): Description des primitives géométriques (formes géométriques de base).
- ISO 19108 (Geographic information -- Temporal schema): Description des primitives temporelles.
- ISO 19109 (Geographic information -- Rules for application schema): Description d'un mécanisme générique d'association des objets géométriques et leurs métadonnées. cette norme est un méta-modèle géométrique qui a donné naissance au GML(4) (ISO 19136).
- ISO 19110 (Geographic information -- Methodology for feature cataloguing): Description complète du schéma de données.
- ISO 19111 (Geographic information -- Spatial referencing by coordinates): Schéma conceptuel pour la description des systèmes de références et de coordonnées.
- ISO 19112 (Geographic information -- Spatial referencing by geographic identifiers): Description des Systèmes de Références utilisant un IDentifiant géographique (SRID).
- ISO 19113 (Geographic information -- Quality principle): Description des principes sur la qualité de la donnée.
- ISO 19114 (Geographic information -- Quality evaluation procedures): Description de la procédure pour l'évaluation de la qualité d'un jeu de données
- ISO 19115 - ISO 19115-2 (Geographic information – Metadata): Description des métadonnées de l'information géographique. La norme ISO 19115-2 est une extension cette dernière pour les métadonnées concernat les images géographiques.
- ISO 19117 (Geographic information – Portrayal): Description des informations de représentation (légendes).
- ISO 19118 (Geographic information – Encoding) : Règles d'encodage pour l'échange d'information.

- ISO 19119 (Geographic information – Services): Description des services (métadonnées de services).
- ISO 19123 - Geographic information -- Schema for coverage geometry and functions: Description des couvertures géométriques.
- ISO 19125-1 - ISO 19125-2 (Geographic information -- Simple feature access): Définition de l'architecture des entités géographiques simples (Partie 1) et du SQL (Partie 2) pour accéder à ces entités. une entité géographique simple peut être une donnée géoréférencée de type vectoriel par exemple.
- ISO 19128 (Geographic information – Web Map server interface): Description d'un serveur cartographique.
- ISO 19130 (Geographic information -- Sensor and data models for imagery and gridded data): Modèle de description de propriétés physiques et géométriques des capteurs.
- ISO 19136 (Geographic information -- Geography Markup Language): GML(4) est un encodage XML(5) pour le transfert et le stockage de l'information géographique en conformité avec les règles d'encodage de la norme ISO 19118.
- ISO 19137 (Geographic information -- Generally used profiles of the spatial schema and of similar important other schemas): Description d'un jeu minimum d'éléments géométriques (issus de la norme ISO 19107) permettant la création de schémas d'application.
- ISO 19138 (Geographic information -- Data quality measures): Description de la qualité de la donnée
- ISO 19139 (Geographic information -- Metadata – XML schema implementation): Description d'un schéma d'encodage en XML(5) pour décrire, valider et échanger des métadonnées géographique conformes à la norme ISO 19115.

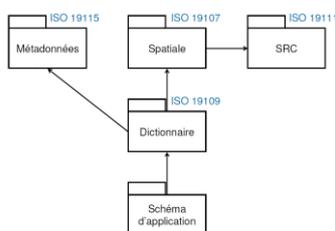


Fig 2. Quelques relations entre des modèles ISO

## 1.2 COMPARAISON DES MODELES SPATIAUX

Dans le tableau (Tab 1), [YANBING, *et al*, 2007] établissent une comparaison entre les modèles spatiaux en termes d'utilisation, primitives utilisées, bases des modèles, avantages et inconvénients.

Type	Nom	Auteur/Date	Utilisation	Primitives	Objets/entités	Base du modèle	Avantages	inconvénients
Modèle géographique	3D FDS	Molenaar 1990	Modèles urbains 3D	Nœud, arc, arête, face	Point, ligne, surface, volume	Single value map	Facile de mettre en place l'adjacence entre des objets spatiaux et non spatiaux	Difficile de définir des objets spatiaux complexes
	Modèle n-tuple	Pigot 1992	Géology, bâtiments complexes	0-3 cellules	0-3 tuple cell complexe	n-manifold	Expression topologique, maintenance facile	Difficile de définir des objets spatiaux complexes
	SSM	Zlatanova 2000	Orienté-Web, visualisation, requête	Noeuf, face planaire	Point, ligne, surface, volume	Simplex concept	Primitives abstraites faciles, stockage léger.	Mise à jour dynamique et modification complexes. Objets constructifs multi-valués
	UDM	Coors 2003	Visualisation urbaine	Nœud, triangle	Point, ligne, surface, volume	Triangulation	Primitives abstraites faciles, données légères, visualisation faciale	Mise à jour dynamique et modification complexes.

	OO3D	Shi Wenzhong, Yang bisheng 2002	Visualisation urbaine	Nœud, segment, triangle	Point, ligne, surface, volume	Modélisation OO	Visualisation d'objets complexes et LOD	Topologie indistinctes, peu de fonctions d'analyse spatiale 3D.
Modèle géologique	3D TIN	Abdul 2000	Géologie, terrain	Nœud, ligne, surface, solide	Point, ligne, surface, volume	Single value map/FDS	visualisation faciale	Pas de données attributaires associées aux objets
	GRID		terrain	Carré, rectangle		Surface divide	visualisation faciale	Données massives
	TEN	Pilouk 1996	Geo-information, pollution	Nœud, arc, triangle, tétraèdre	Point, ligne, surface, volume	Simplex divide	visualisation faciale, volumes et faces complexes	Visualisation d'objets complexes Difficile, stockage important
	Octree	Hunter 1978	CAD/CAM, geology, mer	Cube		Body divide	Structure simple, expression de transformation d'attributs spatiaux internes	Difficiles de définir les frontières géométriques d'objets géologiques
	Modèle solide	Simon Houlding 1994	Géologie, mine	Composant	Frontières des objets géologiques	3-couple	Structures complexes	Travail manuel important
	GTP	Wu LiXin 2004		Nœud, ligne (TIN-L, Side-L), face, (TIN-F, Side-F), GTP, diagonale	Point, ligne, surface, volume	Body divide	Topologie complète, Définition 3D de la géologie, basé sur des données contenant des trous	Visualisation d'objets géologiques complexes difficile
Modèle intégré	TIN+CSG	Li Qingquan, 1998	Modèle 3D urbains	Nœud, segment, triangle	Point, ligne, surface, volume, DEM	Surface divide	2 modèles intégrés dans la surface utilisateur.	Difficile de définir des objets géologiques complexes tels que les failles, les plis et les fissures
	TIN + Octree	Shi Wenzhong, 1996	Modèle 3D urbains, Génie géologique	Nœud, segment, triangle, octree		TIN express surface, Octree express inside	Expression de topologie, interrogation facile	Données octree changent avec les changements des données TIN,
	TEN + Octree	Li Qingquan, Li Deren	Génie géologique	Octree, TEN		Octree express whole, TEN express part	Objets précis, stockage léger.	Définition difficile de la topologie des objets spatiaux

**Tab 1.** Comparaison des différents modèles spatiaux [YANBING, *et al*, 2007]

Et dans le tableau (Tab 2) ci-dessous, [PINET, *et al*, 2003] décrivent les différentes méthodes de conception et formalismes pour la géomatique.

	Année de création	Version précédente	Démarche méthodo.	Spécialités	AGL
MECOSIG	1994	-	Oui	Regroupement des classes par thèmes	Non
OMEGA	1995	-	Oui	Adapté pour les contacts avec les utilisateurs finaux	AIGLE
GeoOOA	1995	-	Partielle	Description des réseaux (canalisation, réseau hydraulique,...)	REGIS
MADS	1997	-	Partielle	Représentation multi-résolution	COBALT
OMT-G	1999	Geo-OMT	Oui	Description des réseaux, données spatiales continues	Non
GeoFrame	1999	GMOD	Partielle	Description des données spatiales continues	Non
Geo-UML	1999	MODUL-R	Adaptation possible de la démarche de MODUL-R	Modélisation précise des historiques de données	Perceptory
T-OMEGA	2002	OMEGA	Oui	Méthode intégrée pour la spécification des applications géomatiques mais aussi télégéomatiques	T-AIGLE
Profil SIG	En cours	-	En cours	Méthode basée sur des patrons de conception et développée par le Cemagref	Extension de Objectteering

**Tab 2.** Méthodes de conception et formalismes pour la géomatique [PINET, *et al*, 2003]

### 1.3 EXEMPLES D'OUTILS SPATIAUX

Il existe une multitude d'outils spatiaux, libres ou payants, pour mettre en place différentes applications spatiales. Un échantillon de tels outils est décrit ci-dessous.

- Global mapper : Outre la visualisation des données vectorielles et rasters, Global mapper est un outil de conversion, édition, impression, acquisition des données GPS, etc. Le site officiel de Global mapper est : <http://www.globalmapper.com>.
- GeoTiff : C'est un standard permettant d'ajouter des informations de géoréférencement à des images TIFF. Le site officiel de GeoTiff est : <http://trac.osgeo.org/geotiff>.
- Geoserver : Geoserver est un logiciel libre écrit en Java permettant le partage et l'édition des données spatiales. Désigné pour être interopérable, Geoserver est capable de publier la plus part des sources de données spatiales en s'appuyant sur les standards. Le site officiel de Geoserver est : <http://geoserver.org>.
- Opendegeo : Opendegeo propose toute une gamme de solutions logicielles dans des domaines spatiaux. Le site officiel d'Opendegeo est <http://opendegeo.org>.
- GeoWebCache : GeoWebCache peut être considéré comme un proxy entre un client comme Google Maps et un serveur tel que Geoserver pour optimiser l'interrogation des cartes géographique. Le site officiel de GeoWebCache est <http://geowebcache.org>.
- GeoEXT : GeoExt se base sur le langage de programmation Web JavaScript pour alimenter un site Web par des fonctionnalités SIG. Le site officiel de GeoEXT est <http://www.geoext.org>.
- GeoTools : GeoTools développé en Java qui implémente les spécifications de Open Geospatial Consortium et permet de déployer des cartes géographiques via le Web. Le site officiel de GeoTools est <http://www.geotools.org>.

- GDAL : GDAL (Geospatial Data Abstraction Library) est une librairie de translation de données raster. Le site officiel de GDAL est <http://www.gdal.org>.

## 1.4 EXEMPLES DE SIG LIBRES

Il existe des SIG Libres pour assurer différentes fonctionnalités spatiales. Un petit échantillon est expliqué ci-dessous.

- DIVAGIS : C'est un outil SIG libre pour la cartographie et l'analyse de données spatiales. Le site officiel de DIVAGIS est <http://www.diva-gis.org>.
- MIGRATIO : C'est un logiciel de cartographie statistique automatique permettant de créer des cartes géographiques statistiques en plages de couleurs ou choroplèthes. de MIGRATIO est <http://www.migratio.fr>.
- GVSIG : C'est une application SIG libre développée en Java permettant d'accéder à des données en fichiers SIG (GML, SHP) ou CAD (DWG, DXF, DGN), à des bases de données spatiales (PostGIS, MySQL, Oracle) ou serveur (WMS, WCS, WFS, Catalogue ou Gazzeteet), pour Linux, Windows et Mac OS X. Le site officiel de GV SIG est <http://www.gvsig.org>.
- SAGA GIS: SAGA (System for Automated Geoscientific Analyses) est un SIG libre proposant des fonctionnalités différentes pour la gestion et l'analyse de cartes géographiques 2D. Le site officiel de SAGA GIS est <http://www.saga-gis.org/en/index.html>.

## 2 ANNEXE 2 : METHODES DE CONCEPTION DES SYSTEMES D'INFORMATION

Les méthodes de conception des systèmes d'information les plus répandues dans le monde sont la méthode Entité-Relation (ER), Orienté-Objet (OO) et Objet-relationnelle. Nous présentons, ci-dessous, une brève définition pour chaque méthode.

### 2.1 ENTITE-RELATION (ER)

La méthode Entités-Relations est aussi appelée Entité-Association. Une entité (occurrence d'entité ou instance d'entité) est définie comme tout concept concret (une voiture) ou abstrait (département Informatique) que l'on reconnaît comme individualisable. Toutes les entités du "monde perçu" sont distinctes. Le regroupement d'entités de même nature ayant les mêmes propriétés structurelles (attributs) constitue un niveau générique appelé Classe d'entités ou Type d'entités. On admet que les entités sont en association les unes avec les autres. Dans le modèle de base une association est définie comme un groupe de deux ou plusieurs entités, chacune jouant un rôle défini dans ce groupe. Une classe d'association (ou type d'association) constitue le niveau générique de toute association présentant les mêmes caractéristiques (entités associées et rôles). Le nombre de classes d'entités associées est appelé dimension ou arité ou degré de la classe d'association : on parle de type d'association binaire, ternaire, ... n-aire. On étend le modèle de base, en autorisant la définition d'attributs (ou propriétés) au niveau d'un type d'association.

#### Représentation graphique

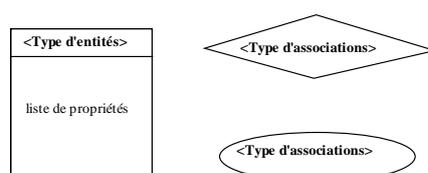


Fig 3. Concepts fondamentaux d'un modèle conceptuel entité-relation

Diverses représentations graphiques sont utilisées, de manière générale un type d'entités est représenté par un rectangle, un type d'association par un losange ou un ovale.

## 2.2 ORIENTE OBJET (OO)

Une règle, déjà proposée dans les règles de bases, est particulièrement importante ici, il s'agit de la *séparation de la structure et du comportement*. La structure est stable car (quasi)indépendante du temps, lorsque le comportement est la chose la plus variable. Le comportement consiste à introduire le temps dans la structure. Sans compréhension préalable de la structure, aucune modélisation du comportement n'est possible (business process, ...). Or l'informatique est la discipline des systèmes dynamiques, c'est-à-dire comportementaux, par excellence. L'algorithme a donc longtemps été le concept de prédilection des développeurs de logiciels. Cet aspect est bien entendu prépondérant, mais ne se modélise valablement qu'une fois la structure maîtrisée.

Dans la pratique, la séparation structure-comportement n'est pas temporelle : on commence par une partie de la structure, puis on introduit, sur des schémas, voire des modèles à part le comportement qui agit sur cette structure. On obtient ainsi une méthode de création itérative :

- modélisation de la structure (principalement sur schéma statique)
- modélisation du comportement (sur des schémas dynamiques)
- validation du niveau de qualité obtenu

Ces itérations se répètent jusqu'à obtention de la qualité et du niveau de précision souhaités [TUDOR, 2006].

UML (Unified Modeling Language) est une méthode très connue pour la modélisation orientée-objet grâce à un ensemble de modèle graphiques permettant de décrire des la structure et le comportement d'un objet donné.

Par exemple dans [OGC, 2008], un diagramme de classe peut être représenté selon la représentation graphique ci-dessous.

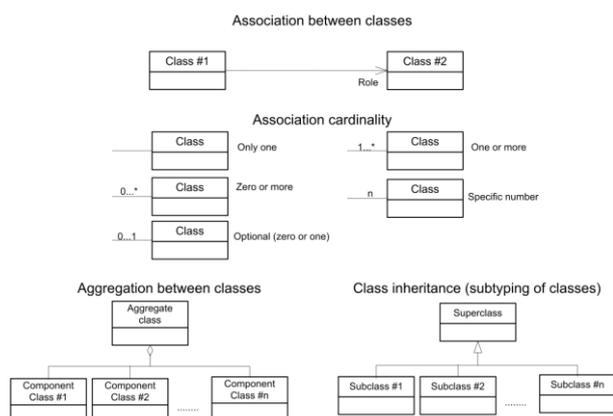


Fig 4. Concepts fondamentaux d'un diagramme de classe UML

## 2.3 OBJET-RELATIONNEL

La modélisation objet-relationnelle, appelée aussi relationnel étendu, permet de combiner les avantages des deux méthodes ER et OO en évitant au mieux leurs inconvénients.

### 3 ANNEXE 3 : LE CODE DE L'APPLICATION

#### 3.1 FONCTION DE CREATION DE SOLIDES

```

CREATE OR REPLACE FUNCTION solid_creation(
table_name character varying[],
tables_number numeric,
attribute_name character varying,
attribute_value numeric)
RETURNS numeric[] AS
$BODY$
DECLARE id_surface numeric; result character varying; counter numeric;
myquery character varying = ''; surfaces_cursor refcursor;
BEGIN
result = ''; counter=1; id_surface=0;
FOR i IN 1..$2 LOOP
IF i=1 THEN
myquery = myquery || ' (SELECT oid' || ' FROM ' || $1[i] || ' WHERE ' || $3 || ' = '
|| $4 || ' ) ';
ELSE
myquery = myquery || ' UNION (SELECT oid' || ' FROM ' || $1[i] || ' WHERE ' || $3 || '
= ' || $4 || ' )';
END IF;
END LOOP;
OPEN surfaces_cursor FOR EXECUTE myquery;
LOOP FETCH surfaces_cursor INTO id_surface; EXIT WHEN NOT FOUND;
IF counter = 1 THEN result = result || id_surface;
ELSE result = result || ',' || id_surface;
END IF;
counter = counter+1;
END LOOP;
CLOSE surfaces_cursor;
RETURN '{' || result || '}';
END;
$BODY$
LANGUAGE 'plpgsql' VOLATILE

```

**Fig 5.** Fonction PostGIS pour la création d'un solide

## 3.2 CLASSE JAVA QUATERNION

```

public class Quaternion
{
//-----
    /**
     * Class variables declaration
     */
    float X;
    float Y;
    float Z;
    float W;
    float Deg2Radian=0.01745329f;
//-----
    /**
     * The class construtor
     */
    public Quaternion(float x, float y,
float z, float w)
    {
        X=x;
        Y=y;
        Z=z;
        W=w;
    }
//-----
    /**
     * get x
     */
    public float getX()
    {
        return X;
    }
//-----
    /**
     * get y
     */
    public float getY()
    {
        return Y;
    }
//-----
    /**
     * get z
     */
    public float getZ()
    {
        return Z;
    }
//-----
    /**
     * get w
     */
    public float getW()
    {
        return W;
    }
//-----
    /**
     * the set methods
     */
//-----
    /**
     *
     */
    public void setX(float x)
    {
        X=x;
    }
//-----
    /**
     *
     */
    public void setY(float y)
    {
        Y=y;
    }
}

```

```

//-----
    /**
     *
     */
    public void setZ(float z)
    {
        Z=z;
    }
//-----
    /**
     *
     */
    public void setW(float w)
    {
        W=w;
    }
//-----
    /**
     * We need to get the inverse of a
     quaternion to properly apply a quaternion-
     rotation to a vector
     * The conjugate of a quaternion is the
     same as the inverse, as long as the
     quaternion is unit-length
     */
    public Quaternion getConjugate()
    {
        Quaternion quaternion=new
Quaternion(-X, -Y, -Z, W);
        return quaternion;
    }
//-----
    /**
     * Normalize a quaternion
     */
    public void normalizeQuaternion()
    {
        float norme= (float)
Math.sqrt(Math.pow(X, 2)+ Math.pow(Y, 2)+
Math.pow(Z, 2)+ Math.pow(W, 2));
        if(norme!=0)
        {
            X/=norme;
            Y/=norme;
            Z/=norme;
            W/=norme;
        }
    }
//-----
    /**
     * Multiplying two quaternions
     */
    public Quaternion
multiplyQuaternion(Quaternion quaternion)
    {
        Quaternion resultQuaternion=new
Quaternion(
        W*quaternion.getX() +
X*quaternion.getW() + Y*quaternion.getZ() -
Z*quaternion.getY(),
        W*quaternion.getY() -
X*quaternion.getZ() + Y*quaternion.getW() +
Z*quaternion.getX(),
        W*quaternion.getZ() +
X*quaternion.getY() - Y*quaternion.getX() +
Z*quaternion.getW(),
        W*quaternion.getW() -
X*quaternion.getX() - Y*quaternion.getY() -
Z*quaternion.getZ());
        return resultQuaternion;
    }
//-----
    /**
     * Quaternion rotation
     */
}

```

```

public void rotateQuaternion(float
angle, float UpX, float UpY, float UpZ)
{
    float cosAngle= (float)
Math.cos((angle*Deg2Radian)/2);
    float sinAngle= (float)
Math.sin((angle*Deg2Radian)/2);

    Quaternion quaternionResult=new
Quaternion
(
    UpX*sinAngle,
    UpY*sinAngle,
    UpZ*sinAngle,
    cosAngle
);

    Quaternion quaternionMultiplied=
quaternionResult.multiplyQuaternion(this);
    Quaternion quaternionConjugate =
quaternionResult.getConjugate();

quaternionMultiplied=quaternionMultiplied.m
ultiplyQuaternion(quaternionConjugate);

this.setX(quaternionMultiplied.getX());
this.setY(quaternionMultiplied.getY());
this.setZ(quaternionMultiplied.getZ());
this.setW(quaternionMultiplied.getW());
}
// -----
/**
 * Quaternion to rotation matrix
 */
public float[] toRotationMatrix()
{
    float mat[]=new float[16];

    float xx = X * X;
    float xy = X * Y;
    float xz = X * Z;
    float xw = X * W;

    float yy = Y * Y;
    float yz = Y * Z;
    float yw = Y * W;

    float zz = Z * Z;
    float zw = Z * W;

    mat[0] = 1 - 2 * ( yy + zz );
    mat[1] = 2 * ( xy - zw );
    mat[2] = 2 * ( xz + yw );

    mat[4] = 2 * ( xy + zw );
    mat[5] = 1 - 2 * ( xx + zz );
    mat[6] = 2 * ( yz - xw );

    mat[8] = 2 * ( xz - yw );
    mat[9] = 2 * ( yz + xw );
    mat[10] = 1 - 2 * ( xx + yy );

    mat[3] = mat[7] = mat[11] =
mat[12] = mat[13] = mat[14] = 0;
    mat[15] = 1;
    return mat;
}
// -----
/**
 * Quaternion to rotation matrix
 */
public void matrixToQuaternion(float[]

```

```

mat)
{
    float T= 1 + mat[0] + mat[5] +
mat[10];
    float S;

    /*If the trace of the matrix is greater
than zero, then
perform an "instant" calculation.
Important note wrt. rounding errors:*/

    if ( T > 0.00000001 ) //to avoid large
distortions!
    {
        S = (float) (Math.sqrt(T) * 2);
        this.setX(( mat[9] - mat[6] ) / S);
        this.setY(( mat[2] - mat[8] ) / S);
        this.setZ(( mat[4] - mat[1] ) / S);
        this.setW(0.25f * S);
    }

    /*If the trace of the matrix is equal to
zero then identify
which major diagonal element has the
greatest value.
Depending on this, calculate the
following:*/

    if ( mat[0] > mat[5] && mat[0] > mat[10] )
    {
        // Column 0:
        S = (float) (Math.sqrt( 1.0 + mat[0] -
mat[5] - mat[10] ) * 2);
        this.setX(0.25f * S);
        this.setY((mat[4] + mat[1] ) / S);
        this.setZ((mat[2] + mat[8] ) / S);
        this.setW((mat[9] - mat[6] ) / S);
    }
    else if ( mat[5] > mat[10] )
    {
        // Column 1:
        S = (float) (Math.sqrt( 1.0 + mat[5] -
mat[0] - mat[10] ) * 2);
        this.setX(X = (mat[4] + mat[1] ) / S);
        this.setY(0.25f * S);
        this.setZ(Z = (mat[9] + mat[6] ) / S);
        this.setW((mat[2] - mat[8] ) / S);
    } else
    {
        // Column 2:
        S = (float) (Math.sqrt( 1.0 + mat[10] -
mat[0] - mat[5] ) * 2);
        this.setX((mat[2] + mat[8] ) / S);
        this.setY((mat[9] + mat[6] ) / S);
        this.setZ(0.25f * S);
        this.setW((mat[4] - mat[1] ) / S);
    }
}
// -----
/**
 * Multiplying two matrix
 */
public float[] multiplyTwoMatrixes(float[]
matrix1, float[]matrix2)
{
    float matrix[]=new float[16];

    matrix[0]=matrix1[0]*matrix2[0]+
matrix1[1]*matrix2[4]+
matrix1[2]*matrix2[8]+
matrix1[3]*matrix2[12];

    matrix[1]=matrix1[0]*matrix2[1]+matr
ix1[1]*matrix2[5]+matrix1[2]*matrix2[9]+mat
rix1[3]*matrix2[13];

    matrix[2]=matrix1[0]*matrix2[2]+matr
ix1[1]*matrix2[6]+matrix1[2]*matrix2[10]+ma
trix1[3]*matrix2[14];

```

```
matrix[3]=matrix1[0]*matrix2[3]+matrix1[1]*matrix2[7]+matrix1[2]*matrix2[11]+matrix1[3]*matrix2[15];

matrix[0]=matrix1[4]*matrix2[0]+matrix1[5]*matrix2[4]+matrix1[6]*matrix2[8]+matrix1[7]*matrix2[12];

matrix[1]=matrix1[4]*matrix2[1]+matrix1[5]*matrix2[5]+matrix1[6]*matrix2[9]+matrix1[7]*matrix2[13];

matrix[2]=matrix1[4]*matrix2[2]+matrix1[5]*matrix2[6]+matrix1[6]*matrix2[10]+matrix1[7]*matrix2[14];

matrix[3]=matrix1[4]*matrix2[3]+matrix1[5]*matrix2[7]+matrix1[6]*matrix2[11]+matrix1[7]*matrix2[15];

matrix[0]=matrix1[8]*matrix2[0]+matrix1[9]*matrix2[4]+matrix1[10]*matrix2[8]+matrix1[11]*matrix2[12];

matrix[1]=matrix1[8]*matrix2[1]+matrix1[9]*matrix2[5]+matrix1[10]*matrix2[9]+matrix1[11]*matrix2[13];

matrix[2]=matrix1[8]*matrix2[2]+matrix1[9]*matrix2[6]+matrix1[10]*matrix2[10]+matrix1[11]*matrix2[14];

matrix[3]=matrix1[8]*matrix2[3]+matrix1[9]*matrix2[7]+matrix1[10]*matrix2[11]+matrix1[11]*matrix2[15];

matrix[0]=matrix1[12]*matrix2[0]+matrix1[13]*matrix2[4]+matrix1[14]*matrix2[8]+matrix1[15]*matrix2[12];

matrix[1]=matrix1[12]*matrix2[1]+matrix1[13]*matrix2[5]+matrix1[14]*matrix2[9]+matrix1[15]*matrix2[13];

matrix[2]=matrix1[12]*matrix2[2]+matrix1[13]*matrix2[6]+matrix1[14]*matrix2[10]+matrix1[15]*matrix2[14];

matrix[3]=matrix1[12]*matrix2[3]+matrix1[13]*matrix2[7]+matrix1[14]*matrix2[11]+matrix1[15]*matrix2[15];

return matrix;
}
```