

Table des matières

Résumé.....	ii
Abstract.....	iii
Avant-Propos.....	iii
Table des matières.....	v
Liste des tableaux.....	vii
Liste des figures.....	viii
Chapitre 1 Introduction.....	9
1.1 Mise en contexte.....	9
1.2 Problématique.....	15
1.3 Hypothèse.....	19
1.4 Objectifs de la recherche.....	19
1.5 Méthodologie.....	19
1.6 Cadre expérimental, tests et validation.....	20
Chapitre 2. De la donnée 3D à un modèle 3D : Revue des concepts et des technologies 22	
2.1 Introduction.....	22
2.2 La construction des modèles géométriques 3D.....	22
2.2.1 Structures géométriques 3D.....	23
2.2.2 La standardisation des structures géométriques.....	31
2.3 La gestion des données et des modèles 3D.....	34
2.4 L'analyse spatiale 3D.....	38
2.4.1 Rappel des concepts.....	38
2.4.2 Standardisation des opérateurs spatiaux.....	40
2.5 Le partage des modèles 3D.....	43
2.5.1 Les services Web.....	44
2.5.2 Limites et solutions pour la diffusion en 3D.....	45
2.5.3 Le Geography Markup Language.....	46
2.5.4 Les spécifications de services Web standards de l'OGC.....	47
2.6 Inventaire des géotechnologies 3D.....	50
2.6.1 Les outils CAO.....	52
2.6.2 Les DAO.....	53
2.6.3 Les SIG.....	53
2.6.4 Les visualisateurs de globes virtuels.....	54
2.6.5 Les SGBD spatiaux.....	55
2.6.6 Recensement des projets de recherche sur le SIG 3D.....	55
2.7 Conclusion du chapitre.....	58
Chapitre 3. Proposition d'une architecture de SIG 3D orientée service Web.....	60
3.1 Les composantes fondamentales d'un SIG 3D.....	60
3.2 Stratégies de développement d'une solution SIG 3D.....	61
3.3 Choix d'une stratégie de développement.....	65
3.4 Architecture du SIG 3D proposé.....	68
3.4.1 Le serveur.....	69
3.4.2 Le Web Feature Service (WFS) 3D.....	70
3.4.3 Le client.....	72

3.4.4	Le Datastore.....	76
3.4.5	L'analyse spatiale	81
Chapitre 4	Implémentation et validation dans un contexte de modélisation géologique	83
4.1	Besoins liés à la modélisation géologique 3D	83
4.2	Architecture du Web Geological Feature Server (WGFS)	86
4.3	Choix des technologies	87
4.3.1	Le client	87
4.3.2	Le serveur	89
4.3.3	La base de données	90
4.4	Développement côté client.....	90
4.4.1	Le plug-in Gocad	90
4.4.2	Le traducteur de données	91
4.4.3	La structure Gocad versus GML.....	92
4.5	Développement de l'application serveur	94
4.6	Implémentation de la base de données	97
4.7	Le WGFS en action.....	99
4.7.1	Fonctionnalité " <i>Voir la liste des modèles 3D disponibles</i> "	100
4.7.2	Fonctionnalité " <i>Importer un modèle</i> "	101
4.7.3	Fonctionnalité " <i>Requêter les propriétés descriptives d'un modèle</i> "	105
4.7.4	Fonctionnalité " <i>Exporter un modèle vers la base de données</i> "	106
4.8	Évaluation du WGFS	108
4.8.1	Description des tests réalisés	108
4.8.2	Résultats.....	110
4.9	Limites du système et travaux futurs	113
Chapitre 5	Conclusion	118
5.1	Bilan et apports de l'étude	118
5.2	Travaux futurs.....	122
Bibliographie	124
Annexe A	Diagrammes de classe du schéma spatial ISO 19107.....	133
Annexe B	Représentation d'objets volumiques sous format GML	135
Annexe C	Détails des spécifications de service Web de L'OGC	137
Annexe D	Inventaires des géotechnologies : Détails techniques des logiciels.....	141
Outils CAO	: Autocad, Microstation et Gocad.....	142
Outils DAO	: 3D Studio Max, Maya et Blender.....	143
SIG	: ArcGIS, MapInfo et Geomedia	143
Visualisateur de globes virtuels:	Google Earth, Virtual Earth et ArcGIS Explorer ...	145
SGBD Spatiaux:	Oracle Spatial, SQL Server et PostGIS.....	146
Autres technologies 3D:	Plateformes et langages de développement Web.....	147
Annexe E	Inventaire des géotechnologies : Justification détaillée du tableau comparatif....	148
Annexe F	Extrait du code source des classes ajoutées ou modifiés pour gérer le 3D dans	
Deegree	151

Liste des tableaux

Tableau 1: Les types de relations spatiales du SFS for SQL extraits de OGC, 1999.	41
Tableau 2: Les opérateurs spatiaux standards du SFS for SQL extraits de OGC, 1999.	42
Tableau 3: Résumé comparatif des capacités des géotechnologies en termes de 3D.	52
Tableau 4 : Correspondance entre les objets Gocad et les éléments GML.	94
Tableau 5 : Les principales primitives géométriques implémentées par Deegree et leur correspondance avec les primitives de la Java Topology Suite (JTS).	95
Tableau 6: Liste des classes du schéma spatial ISO 19107 implémentées dans Deegree dans le cadre de ce projet.	95
Tableau 7 : Validation du prototype WGFS par rapport à la définition théorique du SIG 3D.	110
Tableau 8 : Sommaire des caractéristiques des machines cliente et serveur utilisées lors des tests du WGFS.	111
Tableau 9 : Temps d'importation dans Gocad des différents solides du modèle Porcupine.	112

Liste des figures

Figure 1: Les niveaux d'agrégation d'un modèle tétraédrique.	34
Figure 2: Chevauchement entre une surface et un solide.	42
Figure 3 : Architecture multi-tiers du SIG 3D proposé.	69
Figure 4: Exemple de primitives géométriques représentées sous format WKT.	74
Figure 5: Exemple de représentation de solides sous format WKT 3D.....	75
Figure 6: Schéma de traduction du format XML/GML vers la structure de données cliente	76
Figure 7 : Exemple de stockage de la géométrie d'objets spatiaux en GML et de leurs propriétés descriptives dans une table relationnelle.	79
Figure 8: Modèle tétraédrique de la région Porcupine-Destor Nord-ouest du Québec (modifié de Fallara <i>et al.</i> , 2004) tel que visualisé dans l'interface Gocad.	85
Figure 9: Architecture du Web Geological Feature Server proposée pour le stockage, le requêtage et la diffusion de modèles géologiques 3D	87
Figure 10 : Schéma détaillé du passage des données entre le document XML et l'application cliente Gocad.	90
Figure 11: Extrait de la base de données MySQL montrant la table et les données stockées du modèle géologique 3D de Pocupine-Destor.	98
Figure 12 : Extrait de la boîte de dialogue WFS du plug-in Gocad.....	99
Figure 13 : Schéma de la transaction entre le client et le serveur lors d'une requête <i>GetCapabilities</i>	100
Figure 14 : Liste déroulante de la boîte de dialogue du plug-in Gocad montrant la liste des jeux de données disponibles sur le serveur.	101
Figure 15 : Schématisation de la transaction entre client et serveur lors d'une requête <i>GetFeature</i>	101
Figure 16 : Exemple de requête WFS de type <i>GetFeature</i>	102
Figure 17 : Exemple d'un filtre OGC montrant les paramètres de l'opérateur BBOX.	102
Figure 18 : Extrait de document XML montrant le modèle 3D de Porcupine-Destor et ses propriétés descriptives sous format GML tel que généré par le WGFS.	104
Figure 19 : Partie du modèle de Porcupine Destor tel qu'affiché dans Gocad après l'importation à partir de la base de données.	105
Figure 20 : Exemple de fenêtre du plug-in Gocad montrant les propriétés descriptives d'un solide.	106
Figure 21 : Schématisation de la transaction entre client et serveur lors d'une requête WFS <i>Transaction</i> de type <i>Insert</i> , <i>Update</i> ou <i>Delete</i>	106
Figure 22 : Exemple de fenêtre du plug-in Gocad montrant la liste des propriétés descriptives d'un modèle suite à une requête <i>GetFeatureType</i>	107
Figure 23 : Schéma simplifié d'intégration entre SIGEOM et le WGFS.	117
Figure 24 : Diagramme de classe de la géométrie de base ISO 19107.	133
Figure 25 : Diagramme de classe des primitives topologiques ISO 19107.	134
Figure 26 : Exemple de représentation en format GML d'une unité géologique.	135
Figure 27 : Exemple de schéma XML provenant d'un WFS résultant d'une requête <i>DescribeFeatureType</i>	139

Chapitre 1 Introduction

1.1 Mise en contexte

Depuis toujours on cherche à représenter, à modéliser la réalité du monde qui nous entoure. À l'origine de ce désir, on retrouve plusieurs grandes motivations dont celles de vouloir mieux comprendre cette réalité, d'en dégager diverses simulations, prévisions, de définir diverses règles pour en régir son utilisation ou tout simplement pour prendre des décisions plus éclairées à partir de ces modèles. La modélisation d'une réalité s'effectue par rapport aux propriétés de l'espace, du temps, et/ou de thématiques d'intérêt tel l'écoulement de l'eau, les changements climatiques, etc. En mécanique classique, l'espace est modélisé comme un espace euclidien de dimension 3 dont tous les points peuvent être représentés par des coordonnées. La représentation spatiale des entités et phénomènes de notre réalité à l'intérieur d'un tel espace fait appel aux notions de géométrie. En sciences géomatiques, un grand intérêt est porté à la modélisation géométrique afin de représenter les caractéristiques spatiales de la réalité territoriale. La modélisation géométrique traite plus spécifiquement des propriétés géométriques de l'objet soit la forme, la taille, la position et l'orientation (OQLF, 1997). Par exemple, dans l'espace vectoriel¹, l'utilisation de points, de lignes et de polygones pour représenter des objets de la réalité tels des puits, des routes et des bâtiments, fait partie du processus de modélisation géométrique. Cette représentation des objets se fait à l'intérieur d'un espace défini qui devient l'univers de modélisation. Le positionnement des objets à l'intérieur d'un tel univers est basé sur un système de référence qui permet la gestion de la composante spatiale généralement à l'aide de coordonnées planimétriques (x,y,z) ou angulaires (latitude, longitude, altitude).

Pour le moment, les modèles géométriques les plus utilisés sont généralement basés sur la représentation des objets dans un univers 2D tel un plan ou une carte constituée de points, de lignes et/ou de polygones. Toutefois, le monde qui nous entoure étant lui-même 3D, l'utilisation de la 3^e dimension dans les modèles représentant cette réalité offre de nombreux avantages. La notion de 3^e dimension peut s'appliquer au modèle lui-même (i.e. que la représentation ne soit plus planimétrique mais qu'elle exploite des axes X, Y, et Z)

¹ Représentation mathématique, à l'aide d'opérations sur des vecteurs, de l'ensemble des états physiques possibles d'un système quantique OQLF, 2007.

ou encore à la dimension de l'objet (objet volumique dans ce cas). Tout en ajoutant une dimension supplémentaire à la représentation, le modèle 3D procure une « sensation » de réalité, on a plus l'impression d'être dans cette réalité. La perception et la reconnaissance des objets sont donc plus faciles, plus intuitives. La 3^e dimension permet aussi de représenter certains objets, phénomènes et relations entre eux qui seraient difficiles de modéliser en 2D. Par exemple, la représentation d'une surface géologique qui se replie sur elle-même à plusieurs reprises est complexe, voire impossible à montrer sur une carte (Massé, 2003). C'est cette 3^e dimension qui permet de montrer plus facilement toute la complexité du chevauchement des structures géologiques, connaissances essentielles pour comprendre ces systèmes. Prenons aussi l'exemple du cadastre vertical, ou les cartes cadastrales 2D ne permettent pas de représenter efficacement les propriétés situées au-dessus et au-dessous les unes des autres (Stoter *et al.*, 2003). L'intégration de la 3^e dimension géométrique dans les processus de modélisation/simulation est ainsi à la base de diverses activités de recherche (Zlatanova *et al.*, 2002; Arens *et al.*, 2003; Stoter *et al.*, 2003; Apel, 2004; Pouliot *et al.* 2005; Ledoux et Gold, 2008; Kolbe, 2009). Prenons par exemple la planification urbaine, la représentation du cadastre, la gestion de l'environnement, les télécommunications, les opérations de secours, l'aménagement paysager, la gestion du transport, l'immobilier, les activités hydrographiques, les services publics, les applications militaires, autant d'applications où l'intégration de la 3^e dimension dans les processus de représentation et d'analyse représente une plus value indéniable. Nombreuses sont donc les applications potentielles de la modélisation 3D et la popularité de certaines technologies montrent déjà un engouement certain pour le 3D. Par exemple, les applications comme Google Earth ou Virtual Earth (maintenant Bing Maps) qui présentent une terre en 3D avec son relief et des modèles 3D urbains ont récemment fait leur apparition et sont téléchargeables gratuitement sur Internet. La popularité de ces applications est phénoménale comme le laisse entendre Michael Jones, Google Earth chief technology officer, qui mentionne le chiffre de 400 millions de téléchargements récemment atteint par leur populaire produit.

Si la 3^e dimension dans les modèles fait relativement consensus quant à son utilité, il persiste cependant quelques incertitudes sur les éléments constitutifs d'un modèle 3D, des méthodes et procédures entourant sa construction et sa diffusion et des outils informatiques

utiles à effectuer adéquatement ces tâches. D'abord, les différents systèmes d'acquisition des données 3D nous montrent que la donnée 3D peut tout simplement correspondre à une coordonnée x,y,z (les points issus d'un balayeur laser, un relevé GPS seraient des exemples de ce type d'acquisition), comme elle pourrait correspondre à une coordonnée x,y accompagnée d'une profondeur ou d'une hauteur (ex., des données de forage, un relevé de nivellement) ou encore des matrices montrant les profils d'un objet en particulier (par exemple, la tomographie² est basée sur cette technique pour reconstruire le volume des objets). De cette donnée 3D peuvent par la suite être dérivés différents modèles. Par exemple, un bâtiment 3D pourrait être modélisé à partir d'un polygone (2D) représentant sa base et par une technique d'extrusion qui permettra de montrer la hauteur de ce dernier (cette information de hauteur étant ici stockée dans un attribut de l'objet et non pas comme une coordonnée). Ce bâtiment pourrait aussi être modélisé en assemblant les coordonnées x,y,z de tous les coins et arêtes formant celui-ci (si nous les possédons bien sûr). On pourrait également représenter ce bâtiment en montrant uniquement ses faces, donnant ainsi l'impression du volume, même si celui-ci est bien vide à l'intérieur (on va parler de *meshes*). Ce même bâtiment pourrait aussi être représenté par une combinaison, à l'aide d'opérateurs géométriques booléens, de formes volumiques élémentaires tels le cube, le cylindre, la sphère, le cône. Il conviendra donc de bien connaître ces divers modèles de représentation car chacun d'eux présente des avantages et des limites quant aux efforts à déployer pour construire et valider ces objets 3D et du type d'analyse que nous pourrons en déduire (par exemple, le calcul du volume d'un objet représenté par un modèle qui n'est pas constitué de solides exigera diverses manipulations pour y arriver).

Pour construire, stocker, analyser et visualiser ces modèles, différents outils informatiques existent tels les systèmes d'information géographique (SIG), les outils de conception (CAO³) et de dessins assistés par ordinateurs (DAO). Il est donc intéressant, voire nécessaire, de vérifier les capacités de ces systèmes en termes de gestion de données 3D et du fait même, de s'assurer de ce que nous entendons par 3D.

² La tomographie est une technique qui consiste à reconstruire le volume d'un objet (le corps humain dans le cas de l'imagerie médicale, une structure géologique dans le cas de la géophysique) à partir d'une série de mesures déportées à l'extérieur de l'objet (OQLF, 2003).

³ En anglais on utilise le terme CAD pour *Computer Aided Design* ou encore CAM pour *Computer Aided Manufacturing*.

Depuis le début des années 90, les SIG comme ArcInfo (maintenant ArcGIS), Geomedia, MapInfo, Manifold, Grass, etc. sont devenues de puissants outils pour maintenir et analyser l'information spatiale et descriptive (Longley *et al.*, 2001; Chrisman, 2002). Les fonctionnalités typiques qu'on leur attribue sont souvent présentées comme l'acquisition, la structuration, la manipulation, l'analyse et la diffusion de données spatiales (Raper and Maguire, 1992). Les SIG gèrent des données vectorielles et matricielles. D'une manière générale, les données vectorielles peuvent être des points (0D), des lignes (1D), des polygones (2D) ou des volumes (3D) et ces primitives peuvent utiliser deux ou trois coordonnées pour enregistrer leur position. Cependant, le concept de volume dans les systèmes de type SIG comporte certaines limites. Même si leurs systèmes de coordonnées permettent le stockage de trois axes (X,Y,Z), les objets eux-mêmes sont généralement planimétriques, les modèles en réseaux de triangles irréguliers (TIN) étant un exemple classique de ce genre de données où ceux-ci proposent un assemblage de points, de lignes et de polygones stockés par des coordonnées x,y,z. Il existe cependant la possibilité de montrer le pourtour volumique d'objets en extrudant ceux-ci. La troisième dimension provient alors d'un attribut qui est attaché à une primitive surfacique 2D ce qui est communément appelé le 2.5D.

Les données matricielles quant à elles représentent des maillages continus (réguliers ou non), des tessellations de pixels en 2D ou voxels en 3D. Chaque cellule d'une tessellation peut être positionnée de manière relative en x,y (pixels) ou x,y,z (voxels) par rapport au point d'origine géoréférencé de la grille. Encore une fois, la majorité des SIG ne supportent pas les objets volumiques construits à base de voxels. Ils permettent toutefois la création de modèles numériques de terrain (MNT) présentant par exemple le relief du territoire par l'attribution d'une hauteur à chaque pixel, soit à la manière du 2.5D. Au niveau de l'analyse spatiale, les SIG sont généralement très bien outillés et permettent l'analyse spatiale métrique et topologique des données 2D. Ces analyses sont réalisées par le biais de divers opérateurs implémentant des algorithmes permettant de mesurer et de déduire les relations entre les objets et phénomènes spatiaux des modèles vectoriels et matriciels. Dans la grande majorité des cas, ces algorithmes dans les SIG ne tiennent pas compte de la hauteur ou de la coordonnée z lors des calculs ce qui serait nécessaire pour effectuer de l'analyse spatiale 3D. Pour ce qui est du partage des modèles 3D, les SIG permettent de les exporter sous

format de fichier (format vectoriel, image, pdf, etc...) ou de les diffuser sur le Web par le biais de modules reposant sur le concept de service Web. Ces modules étant plutôt destinés à la cartographie en ligne, ils ne permettent que la diffusion de modèles 2D et 2.5D. En effet, les modules de *Web Mapping* tels que ArcIMS, ArcGIS Server, MapXtreme, MapX ou MapGuide offerts par les fournisseurs de logiciels SIG comme Esri, MapInfo, Geomedia ou Autodesk, sont destinés à la publication des données géométriques produites au sein de leur logiciel SIG respectif donc limités aux données 2D et 2.5D.

De leur côté, les logiciels CAO (Autocad, Micro-Station, Catia, SolidWorks, Earth Vision, Gocad, Google Sketchup, etc.) et DAO (3D Studio Max, Maya, Softimage, etc.) permettent la création, l'édition et la visualisation de modèles 3D. Contrairement aux SIG, ces applications implémentent des structures géométriques spécialement conçues pour la modélisation 3D (B-REP, CSG, Octree...) et permettent donc de modéliser des volumes. Toutefois, ils n'offrent pas les mêmes capacités que les SIG au niveau de l'analyse spatiale puisqu'ils ne mettent généralement à disposition que quelques opérateurs spatiaux de type métriques sans permettre d'analyse spatiale topologique (Pouliot *et al.*, 2003). Ces applications ne permettent pas non plus la gestion des systèmes de coordonnées, ce qui les limite à la modélisation à grande échelle i.e. des modèles couvrant peu de territoire. Du côté gestion et stockage des données, les outils CAO et DAO n'exploitent pas non plus de bases de données, stockant l'information au sein de fichiers. Les outils CAO sont principalement utilisés dans les domaines de l'ingénierie, l'architecture et la construction comparativement aux outils DAO qui sont plutôt utilisés par les infographistes et sont essentiellement orientés vers l'animation cinématographique, les jeux vidéo 3D et la création d'environnements de réalité virtuelle. Les DAO sont avant tout utilisés pour obtenir un environnement virtuel 3D réaliste du point de vue de la visualisation. Ils sont en général basés sur le concept de manipulation de formes de base comme les cubes, les sphères ou des cônes, mais aussi des courbes de Bézier ou des NURBS⁴ qui peuvent être par la suite « modélées » pour obtenir des formes plus complexes, comme par exemple une voiture ou un personnage. Pour ce qui est de leur capacité à distribuer des modèles 3D, il n'existe pas

⁴ *Non Uniform Rational B-Spline*

au sein des outils CAO et DAO de fonctionnalités ou de modules permettant la diffusion Web de ces modèles.

Peu importe le type de logiciels de modélisation utilisés, tous ont recours à une méthode de stockage spécifique. Les SIG et les outils CAO et DAO utilisent les fichiers (*flat-files*) pour le stockage des modèles géométriques. L'accès aux données de ce type de stockage est relativement lent puisque le fichier doit être parcouru en entier pour trouver l'élément recherché. On lui reproche aussi certains problèmes de redondance, de maintenance et d'intégrité des données (Date, 2000). En plus du stockage par fichier, les SIG ont généralement la capacité de stocker les modèles au sein de bases de données relationnelles par le biais d'un système de gestion de base de données (SGBD) comme Oracle, DB2, MySQL, PostgreSQL ou SQL Server. Principalement, les SGBD sont utilisés pour assurer la gestion automatique d'une base de données en permettant la création, la modification, l'utilisation et la protection des données (OLF, 2001). De nombreux avantages liés à l'utilisation des SGBD pour la gestion des données, comparativement à l'écriture à l'intérieur de fichiers, ont été mis de l'avant (Date, 2000; Bruenig *et al.*, 2004; Apel, 2004). Parmi ces avantages on retrouve :

- la possibilité pour plusieurs usagers de travailler en même temps sur des données partagées
- le recouvrement des données en cas de panne
- les "lock" automatiques sur les objets de la base de données lors des transactions
- les mécanismes permettant d'éviter les pertes de données
- la sécurité des données
- l'intégrité des données (la validité et la cohérence des données géométriques peuvent être vérifiées lors de l'insertion des données et lors de l'exécution des requêtes)
- les opérations permettant de faire des requêtes, de l'insertion et de la mise à jour de données

Ces fonctionnalités très intéressantes des SGBD ont d'abord été conçues pour optimiser la manipulation des données descriptives de type texte, entier, date, etc. Cependant, assez rapidement des SGBD spatiaux sont apparus sur le marché (Arens *et al.*, 2003). Les SGBD

spatiaux mettent à disposition, en plus des données descriptives, des données de type primitive géométrique (point, ligne, polygone) et des opérateurs spatiaux facilitant le stockage et la manipulation des modèles géométriques. Ainsi, en stockant les données descriptives et géométriques dans un SGBD spatial, l'ensemble des données d'une organisation peut être centralisé et géré dans un même environnement tout en bénéficiant des avantages des SGBD. Toutefois, tout comme les SIG, ces SGBD spatiaux sont limités aux primitives géométriques ponctuelles, linéaires et surfaciques ce qui signifie qu'ils n'intègrent pas le concept de volume.

Au moment d'écrire ces lignes, Oracle Spatial présentait plusieurs innovations importantes en ce qui a trait au stockage et à la diffusion de modèles 3D. En effet, dans sa version 11.1g, Oracle offre pour la première fois la gestion d'une primitive spatiale volumique, des fonctions de validation de la géométrie (étanchéité et connexion entre surfaces formant le solide), un index spatial 3D de type R-Tree ainsi que des opérateurs spatiaux 3D de type métrique. Au niveau de la diffusion des données, Oracle 11.1g implémente une interface standard de service web pour le partage de données spatiales 3D sur Internet. Ces innovations sont certes très encourageantes pour les acteurs du domaine 3D et démontrent qu'il y a un intérêt réel pour le stockage et la diffusion de données 3D.

Alors que les domaines des SIG et des SGBD orientent leurs développements vers le 3D, les organismes de standardisations font de même et ont commencé à réviser leurs normes et spécifications portant sur la représentation, le stockage et le partage d'entités géométriques pour y introduire une meilleure gestion des objets 3D tels que les objets volumiques. Ainsi, l'International Standard Organisation (ISO) et l'Open Geospatial Consortium (OGC) offrent aujourd'hui des normes et spécifications qui incluent une gestion explicite des objets volumiques telles que le Geography Markup Language 3 (GML 3) et le schéma spatial ISO 19107 destinées à la représentation de données géométriques au sein d'un système.

1.2 Problématique

L'expérience démontre que les SIG et leurs homologues les SGBD spatiaux sont des outils forts intéressants pour la gestion et le stockage des données spatiales. Cependant, la

littérature montre aussi qu'ils ne sont pas encore matures vis-à-vis du stockage, la manipulation et l'analyse du 3D et que dans ce sens ils n'ont pas encore fait leurs preuves quant à la gestion de données volumiques (Bruenig *et al.*, 2004; Stoter *et al.*, 2003; Zlatanova *et al.*, 2002; Pouliot *et al.*, 2003). Ainsi, la majorité des applications actuelles dans le domaine de l'information géographique sont toujours 2D (ou 2.5D) et le passage du 2D vers le 3D n'est pas toujours aussi simple qu'il n'en paraît. Selon les chiffres de Daratech inc. (Directions-Magazine, 2003), les principaux joueurs du marché des systèmes d'information géographique présentaient des revenus de 1,75 milliard US\$ pour l'année 2003 en hausse de 8% par rapport à l'année précédente. De ce montant, 91% provient de la vente de logiciels et de services des compagnies tels qu'Esri, Intergraph Corporation, Autodesk, Inc., GE Network Solutions, Leica Geosystems GIS and Mapping Division, MapInfo Corporation, etc. Les solutions SIG proposées par ces compagnies sont en grande partie limitées aux données spatiales 2D ou 2.5D.

Les SIG et les SGBD spatiaux supportent les données géométriques à coordonnées tridimensionnelles, mais à géométries surfaciques uniquement. Ils ne permettent pas de gérer les volumes. Le problème est qu'il n'existe pas dans ces systèmes, de primitives géométriques volumiques qui permettraient de gérer et stocker fidèlement les modèles 3D qui contiennent des solides. Arens *et al.* (2003) mentionnent que l'absence d'une vraie primitive 3D dans les SGBD spatiaux entraîne entre autres, le mauvais fonctionnement de certains opérateurs spatiaux (ex : le calcul d'aire d'une surface orientée verticalement donne un résultat de 0 puisque sa projection en 2D est une ligne), la redondance dans le stockage des coordonnées et l'impossibilité d'avoir une relation 1:1 entre le nombre d'enregistrements de la table représentant l'objet et la réalité. Du côté des SIG, l'inexistence d'une primitive volumique ne leur permet pas de modéliser, manipuler et analyser efficacement les données 3D au-delà de la visualisation (Zlatanova *et al.*, 2002; Pouliot *et al.*, 2003). En effet, sans primitives volumiques, les SIG se limitent à la visualisation de certains types de modèles 3D représentés par des surfaces importés de sources externes et à la création de modèles 3D obtenus par extrusion de primitives 2D. Ce type de modélisation ne permet pas de représenter les structures géométriques complexes mises en œuvre dans plusieurs domaines d'application et limite grandement les possibilités d'analyse spatiale 3D ce qui restreint donc leur utilisation à la visualisation de certains

modèles simples. Les SIG sont donc généralement limités à la visualisation de modèles représentant le relief du territoire et les bâtiments à sa surface ainsi qu'au calcul d'orientation, de pente ou d'intervisibilité entre points. Les SIG 2D étant très limités quant aux possibilités de modélisation et d'analyse spatiale 3D, le développement de SIG 3D doit être envisagé. Toutefois, le développement d'un tel système devra faire l'objet d'une évaluation consciencieuse afin de rester ouvert, non-propriétaire et interopérable. En effet, les tentatives passées de développer des SIG 3D qui supportent le couplage avec des bases de données ont résulté en des systèmes adaptés uniquement pour des SGBD spécifiques en plus de ne respecter aucune interface standard (par exemple, le GeoToolkit exploite CORBA et repose sur la base de données ObjectStore ne permettant le stockage au sein d'aucun autre SGBD) (Apel, 2004).

Du côté des outils de modélisation de type CAO, ils ont atteint un niveau de maturité suffisant pour la création de modèle 3D de qualité, mais sont encore limités au niveau de l'analyse spatiale et du stockage des données par rapport aux SIG (Zlatanova *et al.*, 2002b; Pouliot *et al.*, 2003). En effet, les outils CAO utilisent des structures permettant la gestion de données volumiques (CSG, B-REP, ...) pour représenter les objets et offrent des outils d'édition et de manipulation adaptés à la modélisation tridimensionnelle. Par contre, ils ont certaines lacunes côté gestion des données puisqu'ils ne gèrent que les modèles 3D stockés sous format fichier et ne permettent généralement pas d'être liés à des bases de données. Certains produits plus récents comme Autocad Map 3D ou Bentley Map, permettent de se connecter à des bases de données. Toutefois, dans le cas précis de ces produits, ils implémentent tant de concepts dérivés des SIG qu'il faut se demander s'ils peuvent encore être considérés comme des outils CAO ou encore de la pertinence de cette distinction SIG-CAD. Ce genre de produit hybride représente bien cette tendance à proposer des produits intégrant les forces de gestion des SIG et les forces de modélisation 3D des CAD. Cependant nous jugeons qu'au moment où ces travaux ont débuté, et encore aujourd'hui cette distinction existe toujours et elle mérite d'être mise de l'avant pour bien comprendre les forces et faiblesses de chacune de ces catégories de logiciels. Aussi, le nombre d'opérateurs spatiaux des outils CAO (et ceux hybrides) est souvent limité à l'analyse spatiale métrique. Par exemple, ils permettent de réaliser des opérations de calculs de distance ou l'application de corridor autour d'objets 3D mais il est souvent difficile ou

impossible de vérifier l'intersection entre des objets 3D ou la présence d'un objet à l'intérieur d'un autre objet 3D.

Si on examine plus particulièrement les capacités de diffusion des modèles 3D, les outils de type SIG et CAO offrent la possibilité de publier sur le Web des données spatiales par le biais de modules et API tels que ArcIMS, MapXtreme, MapX, MapGuide, MapServer, etc. Toutefois, ces produits se limitent à la publication de données 2D et n'offrent pas de solutions pour la diffusion de modèles volumiques. Il existe d'autres types de solution très populaires pour la visualisation 3D sur le Web tels que Google Earth, Nasa World Wind, ArcGIS Explorer ou Virtual Earth de Microsoft. Ces applications montrent le relief du territoire à la manière des MNT (2.5D) ainsi que les bâtiments par extrusion d'entités polygonales 2D. Sans offrir de fonctionnalité de modélisation géométrique 3D, certaines de ces applications comme par exemple Google Earth, permettent d'importer des modèles 3D construits à partir d'applications externes (CAO ou DAO). Ils ne permettent toutefois pas de réaliser d'analyse spatiale 3D à part peut-être le calcul de distance suivant la courbure de la Terre.

La plupart de ces technologies de diffusion Web de données spatiales implémentent le concept de service Web i.e. interface reposant sur les protocoles de communication comme le HTTP et sur le langage XML. Le problème avec ces services Web est que chaque solution implémente sa propre interface, ce qui complique le développement des applications et ne favorise pas l'interopérabilité. Le recours à des interfaces de services Web standards telles que le Web Map Service (WMS), le Web Feature Service (WFS) ou le Web Terrain Service présenté par l'OGC devra être envisagé.

Ainsi, chacun des deux mondes (SIG et CAO-DAO) possèdent leurs avantages et leurs défauts, mais aucun des deux n'est suffisamment complet pour répondre aux besoins imminents d'aujourd'hui, c'est-à-dire la gestion de modèles 3D complexes (pas seulement les MNT), leur stockage au sein de bases de données, leur analyse spatiale métrique et topologique 3D ainsi que leur diffusion Web. Alors que la force des SIG repose sur leur capacité d'analyse spatiale, leur intégration aux bases de données et leur diffusion Web, seuls les outils CAO maîtrisent de façon satisfaisante la gestion des volumes. La solution

semble donc se trouver du côté d'un hybride qui intégrerait les atouts principaux de ces deux types d'applications.

1.3 Hypothèse

Dans un contexte de modélisation d'une réalité territoriale à partir de données à référence spatiale, nous prétendons que la 3^e dimension géométrique, celle des solides, est requise. Actuellement, les architectures de systèmes de type SIG ou CAO, même s'ils présentent des fonctionnalités intéressantes, n'intègrent que partiellement certains aspects et ils ne sont donc pas parfaitement adaptés à la création, à l'interrogation (spatiale ou non), à l'analyse (spatiale ou non) et à la diffusion de modèles volumiques représentant un territoire.

1.4 Objectifs de la recherche

L'objectif principal de cette recherche est de contribuer à l'établissement de concepts nécessaires à la définition et au développement d'une architecture logicielle qui permettrait le stockage et l'interrogation spécifique de modèle 3D tout en favorisant leur interopérabilité. Nous pourrions nommer cette architecture un SIG 3D, car les fonctionnalités recherchées par celle-ci devraient coïncider avec celles fournies par les systèmes d'information géographique. Plus spécifiquement, on veut :

- Évaluer les aptitudes et les limites des (géo) technologies actuelles permettant la gestion et/ou le traitement de données géométriques 3D
- Explorer et mettre à profit les normes et les approches de développement open source pour la mise en place d'un SIG 3D
- Démontrer l'utilité et la faisabilité d'implantation d'une architecture logicielle de type SIG 3D (application au cas de la modélisation géologique 3D)

1.5 Méthodologie

La méthodologie consiste d'abord à effectuer une série de lectures sur les activités de recherche et de développement liées aux thématiques de la modélisation géométrique 3D, de la normalisation et du développement informatique. Ces lectures sont accompagnées de divers exercices de modélisation 3D, nous permettant de mieux comprendre les tenants et aboutissants d'un tel processus, et de découvrir et comparer les géotechnologies 3D

actuellement à disposition. De là, nous avons effectué différentes synthèses nous permettant de préciser les caractéristiques fondamentales d'un SIG 3D. Différents choix stratégiques et technologiques ont alors été effectués, ceux-ci ayant aidé à concevoir une toute nouvelle architecture logicielle. Ces choix ont également été effectués en lien avec les besoins et contraintes d'un client potentiel (Géologie Québec). L'architecture a été implantée et testée pour une application spécifique de modélisation 3D de notre client, celle de la modélisation géologique. Les tests devaient montrer l'utilité, la faisabilité d'implantation et la performance d'une telle solution géomatique 3D, en fonction des besoins et des contraintes énoncés.

1.6 Cadre expérimental, tests et validation

Ces travaux de maîtrise ont été réalisés dans le cadre d'un projet de recherche financé par le Réseau de Centres d'excellence GEOIDE⁵, qui s'intitule "*Development of a 3D predictive modeling platform for exploration, assessment and efficient management of mineral, petroleum and groundwater resources*"⁶. Ce projet GEOIDE vise à développer des outils/méthodes géomatiques pour aider les experts à concevoir des modèles géologiques et hydrogéologiques, et ainsi progresser vers la mise en valeur et l'exploitation équitable des ressources naturelles canadiennes telles, l'eau, les minerais et le pétrole. L'intégration de la 3^e dimension géométrique des données spatiales exploitées dans les modèles 3D est tout particulièrement visée par ce projet GEOIDE. Le présent projet de maîtrise vient compléter deux initiatives qui cherchaient à aider la mise en place d'un SIG 3D, soit celle de Lachance (2005) qui a proposé une structure topologique de données 3D pour l'analyse de modèles géologiques et celle de Bédard (2006) qui proposait un guide procédural pour la construction de modèles géologiques 3D.

Le client, partenaire au projet GEOIDE, était le Ministère de Ressources naturelles et de la Faune du Québec (MRNF), secteur géologie (Géologie Québec). C'est donc avec eux que différents échanges ont eu lieu sur l'établissement des besoins liés à la mise en place d'une telle solution géomatique 3D. Nous remercions le MNRF d'avoir mis à notre disposition différentes données dont un modèle géologique 3D complexe de la région Porcupine-

⁵ GEOMatique pour des Interventions et des DEcisions éclairées, www.geoide.ulaval.ca

⁶ <http://geotopo3d.scg.ulaval.ca/>

Destor (Nord-Ouest du Québec, Canada) contenant des surfaces et des solides construits avec Gocad (Fallara *et al.*, 2004; Fallara *et al.*, 2006; Bédard, 2006).

Rapport-Gratuit.com

Chapitre 2. De la donnée 3D à un modèle 3D : Revue des concepts et des technologies

2.1 Introduction

Pour bien comprendre les défis méthodologiques et technologiques à relever lorsque vient le temps de passer de la donnée 3D à un modèle 3D, nous allons dans cette section présenter les concepts de base nécessaires au processus de modélisation 3D (i.e. la construction du modèle 3D) ainsi qu'aux fonctionnalités connexes telles que la gestion, l'analyse et le partage des modèles 3D. Nous ferons aussi le point sur les normes en vigueur et sur les géotechnologies qui permettent le stockage et le traitement de données géométriques (0D à 3D).

2.2 La construction des modèles géométriques 3D

Comme nous l'avons mentionné au chapitre 1, un processus de modélisation consiste à construire un modèle, c'est-à-dire, une représentation simplifiée d'une réalité. Cette modélisation peut s'effectuer par la représentation des propriétés spatiales, thématiques et/ou temporelles. Le processus qui consiste à représenter les propriétés géométriques dont la position, la forme, la taille et l'orientation des objets d'une réalité se nomme modélisation géométrique (OQLF). En modélisation géométrique 3D, on s'intéresse donc à représenter plus spécifiquement les propriétés géométriques volumiques d'objets spatiaux 3D. Selon ISO 19107, (2002) un objet géométrique est un objet utilisé pour représenter les caractéristiques spatiales d'une abstraction d'un phénomène du monde réel. Dans cet ouvrage, nous utiliserons le terme objet spatial en référence à l'objet du monde réel (le bâtiment) et objet géométrique pour référer par exemple au polygone servant à sa représentation. Larrivée *et al.*, (2006) proposent de bien distinguer la notion de dimension associée à l'objet lui-même et la dimension de l'univers dans lequel cet objet est positionné. Par exemple, la représentation d'un bâtiment pourrait être réalisée par un point associé à son centre géométrique (objet géométrique ponctuel), par une ligne associée à sa façade (objet géométrique linéaire), par un polygone associé à son assise (objet géométrique surfacique) et par un polyèdre associé à son volume (objet géométrique volumique). Ces différentes représentations pourraient être placées dans un univers 1D, 2D et 3D. Par exemple, le polygone associé à son assise (objet surfacique) pourrait être positionné sur une

carte topographique (univers 2D). On pourrait aussi positionner ce même objet surfacique dans un univers 3D comme un modèle 3D de la ville dans lequel le bâtiment se situe. Étant donné cette situation potentiellement problématique, nous avons convenu de parler d'objets 3D ponctuels, linéaires, surfaciques ou encore volumiques pour faire référence à la représentation de ces objets positionnés dans un univers 3D (i.e. un modèle 3D). Dans le cas d'objets spatiaux 2D, on ferait alors référence à des objets ponctuels, linéaires et surfaciques positionnés dans un univers 2D comme une carte par exemple.

2.2.1 Structures géométriques 3D

Pour bien comprendre le processus de construction des modèles 3D et les éléments constitutifs des différentes structures de modélisation géométrique 3D, nous allons d'abord présenter les principales structures géométriques. En cartographie, une structure géométrique correspond à la disposition relative de fins éléments graphiques identiques, simples ou complexes, régulièrement répétés (OQLF, 1970). Elle est donc en quelque sorte le schéma ou le plan décrivant l'organisation logique des données géométriques (vectorielles ou matricielles) au sein d'une application. Cette structure géométrique est en général étroitement liée à un système de stockage particulier nécessaire à la persistance des données.

En géomatique, on distingue généralement deux catégories de structure géométrique, l'une dite vectorielle et l'autre matricielle (Longley *et al.*, 2001; Rigaud *et al.*, 2002). L'approche vectorielle (dite orientée-objets) implique des primitives géométriques de type point (0D), ligne (1D), polygone (2D) ou polyèdre (3D). Les objets de la réalité peuvent alors être représentés à partir des coordonnées des points et arêtes décrivant leur position et leur forme. Des polygones ou des volumes représentant les limites des objets peuvent aussi être construits à partir d'arêtes formant une boucle fermée. De son côté, l'approche matricielle (dite orientée-espace) représente la réalité de manière continue et implique des structures où le concept d'objet n'est pas explicite, mais accessible à travers le regroupement des mailles discrétisant l'espace géographique. On nomme raster, grid ou image, le fichier contenant des données matricielles. Un fichier raster peut être vu comme un tableau rectangulaire de cellules carrées identiques appelées pixel dont chacun d'eux se voit associer une valeur numérique ou nulle. Une fois un raster géoréférencé, la position d'un pixel dans le tableau

permet de lui attribuer une coordonnée x,y et la valeur numérique contenue dans le pixel peut être utilisée pour le stockage de la hauteur (valeur z) ou toute autre valeur comme par exemple la mesure du rayonnement électromagnétique acquise par télédétection. En 3D, on parlera plutôt de cellules volumiques (par exemple cubiques) appelées voxels géoréférencées en x,y,z .

En plus de cette distinction entre structures vectorielle et matricielle, un autre concept fondamental, celui de la topologie, doit être pris en compte. La topologie peut être définie comme la branche des mathématiques traitant des relations de voisinage qui s'établissent entre des figures géométriques, et qui ne sont pas altérées par la déformation des figures (OQLF, 2005). Elle fait référence aux relations de voisinage entre les primitives géométriques formant un objet d'un modèle et entre les objets eux-mêmes (micro et macro-topologie). Son application permet d'assurer la cohérence des données géométriques et facilite leur utilisation et leur stockage en éliminant la redondance au niveau des primitives géométriques. Elle vient aussi faciliter l'analyse spatiale puisque celle-ci fait généralement intervenir les relations entre les objets et les primitives spatiales d'un modèle (ex : est-ce que l'objet A se trouve à l'intérieur de l'objet B?) et ce sont spécifiquement ces relations qui sont stockées dans une structure dite topologique. Lorsque la topologie n'est pas explicitement stockée, elle doit être déduite et calculée à partir de la géométrie lors des traitements.

En 3D, la classification vectorielle/matricielle est toujours présente même si plusieurs auteurs ont proposé d'autres catégorisations (Billen, 2002; Breunig, 2001; De La Losa, 2000; Li, 1994; Pouliot *et al*, 2006a; Ramos, 2003; Raper, 1992 ; Shapiro, 1997; Van Den Bergen, 2004). Dans le cadre de l'application de type SIG 3D qui nous intéresse, il sera important de distinguer le type de primitives utilisées pour la représentation de volumes (objets 3D volumiques). C'est pourquoi nous proposons de regrouper et de présenter les structures selon trois niveaux de complexité basés sur le type de primitives utilisés pour construire les objets :

1. les structures basées sur des points et des lignes (fil de fer),
2. les structures basées sur des surfaces (B-Rep, TIN, ...),
3. les structures basées sur des solides (CSG, Voxel, ...)

La description que nous en ferons prend pour hypothèse que les objets sont des manifolds puisqu'il existe tout un monde de modélisation pour le non-manifold. Un manifold est un espace topologique localement euclidien i.e. autour de chaque point, il existe un voisinage qui est topologiquement le même que la sphère unitaire ouverte dans \mathbb{R}^n (<http://mathworld.wolfram.com>). Les objets non Manifold ne respectent pas cette règle.

Structures de modélisation basées sur des points et des lignes

La structure de modélisation basée sur des points et des lignes, appelée fil de fer, consiste à créer des objets à l'aide de nœuds et d'arrêtes formant le squelette de leurs frontières. Ce type de modèle est, parmi ceux de nos trois niveaux de complexité, le plus simple et le moins gourmand côté espace de stockage. Par contre, cette simplicité n'a pas que de bons côtés puisqu'elle ne lui permet pas de représenter sans ambiguïté n'importe quels objets. En effet, le même objet peut être interprété différemment selon que l'on considère fermés ou ouverts les plans implicitement créés par les segments composant les limites de l'objet. Cette méthode n'est donc pas très efficace pour la représentation de volumes puisque la surface des objets n'est pas fermée et leur contenu est vide. Cela a aussi pour effet de compliquer les calculs d'analyse spatiale puisqu'il n'y a pas de topologie entre les différents objets (seulement au niveau micro i.e. les segments adjacents partagent les mêmes points) et le calcul de volume doit être effectué à partir des coordonnées des primitives formant le squelette de la frontière des objets. Au niveau de sa capacité à représenter l'irrégularité des formes, cette structure peut être qualifiée de performante puisqu'il suffit d'introduire plus de nœuds au niveau des lignes pour représenter plus précisément les objets et les phénomènes.

Structures de modélisation basées sur des surfaces

Le deuxième niveau de complexité implique l'utilisation de surfaces pour former les limites des objets ce qui permet de régler le problème d'ambiguïté du modèle fils de fer. Toutefois, le contenu des objets surfaciques ainsi créés reste vide, ce qui complique encore une fois le calcul du volume qui doit être obtenu à partir de la géométrie des surfaces le composant.

Les surfaces d'un modèle surfacique peuvent être planes ou courbes. Les modèles composés de surfaces planes sont souvent appelés « *polygon mesh* » et sont généralement composés de faces triangulaires (TIN), quadrilatères ou d'autres polygones convexes

simples afin de simplifier le rendu graphique. Ces faces peuvent être de formes régulières ou irrégulières et sont composées de vertex (points) et de *edges* (lignes). Le stockage d'un *polygon mesh* consiste minimalement en le stockage des coordonnées de chaque vertex du *mesh* et de la liste de ses vertex adjacents. Afin de faciliter la représentation des faces lors du rendu graphique, on préfère généralement stocker explicitement les faces en tant que liste des vertex qui les composent. Afin d'offrir une plus grande flexibilité lorsque des changements sont apportés dynamiquement au modèle, on peut aussi stocker les *mesh* suivant la méthode appelée *Winged-edge* (Baumgart, 1975). Cette méthode consiste à enregistrer explicitement pour chaque surface, ses vertex, ses faces, mais aussi chaque *edge*. Pour chacun de ces *edges*, ses deux vertex, les deux faces qui le bordent de chaque côté ainsi que les deux *edges* sens horaires et sens anti-horaires les plus près de chaque extrémité sont aussi stockés.

De leur côté, les surfaces courbes sont obtenues par l'application d'algorithmes comme les *Splines*, le plus souvent des NURBS, sur les *edges* composants les faces des surfaces permettant ainsi d'obtenir des courbes lisses et harmonieuses. Pour créer une telle surface on peut partir d'une surface simple (ex : surface plane) laquelle on déforme librement en déplaçant des points de contrôle se trouvant le long des *edges* composant la surface. On peut aussi partir d'une courbe lissée à l'aide d'une *Spline* à laquelle on applique une technique comme le balayage. Le balayage consiste en le déplacement d'un objet 2D comme une courbe ou une surface par exemple, le long d'une trajectoire pour créer un objet. Cette méthode permet de facilement stocker les objets puisque seulement l'objet 2D ainsi que la formule de sa trajectoire doivent être enregistrés. Cette méthode n'est toutefois pas adaptée pour la modélisation d'objet de forme irrégulière puisque le déplacement d'une forme 2D le long d'une trajectoire implique obligatoirement une certaine régularité.

Qu'on traite de surfaces planes ou courbes, l'utilisation de « *mesh* » est souvent suffisante pour la visualisation de ces modèles surfaciques. Toutefois, pour certains modèles plus complexes représentant par exemple des objets surfaciques 3D troués ou contenant d'autres objets 3D, il devient parfois nécessaire de stocker l'orientation de chacune des faces composant sa coquille (*shell*) afin de pouvoir distinguer l'intérieur de l'extérieur de l'objet.

Boundary Representation (B-Rep)

La structure de modélisation basée sur des surfaces la plus connue est appelée B-Rep (Boundary Representation). Cette technique largement implémentée au niveau des outils CAO et DAO est l'une des deux techniques de modélisation géométrique 3D, avec le CSG, les plus répandues. Un B-Rep permet de représenter des objets volumiques 3D seulement par l'assemblage des faces formant sa surface. Un B-Rep est essentiellement une structure en *Graph* dont les nœuds correspondent aux faces, aux *edges* et au vertex et dont l'assemblage forme les frontières d'un solide. Les liens entre les nœuds représentent la connectivité entre les primitives assurant la consistance topologique et les coordonnées liées à chaque vertex correspondent à l'information géométrique. Dans un B-Rep, une face est une portion d'une surface fermée et orientée contenant sa frontière représentée par des *edges*. Un *edge* est une partie d'une courbe contenant sa frontière (des vertex) et un vertex correspond à un point. Pour créer, manipuler et éditer les faces, *edges* et vertex d'un B-Rep, on utilise généralement les opérateurs d'Euler. Les opérateurs d'Euler sont au B-Rep ce que les opérations booléennes sont au CSG. Ils permettent d'assurer l'intégrité des frontières du modèle (fermeture, étanchéité, présence d'*undershoot* et d'*overshoot*) et offrent un mécanisme pour en vérifier la validité. Le principal avantage du B-Rep est sa grande capacité à représenter des solides de forme irrégulière. Il est aussi relativement simple de transformer un B-Rep en une représentation fil de fer, souvent utilisée pour des raisons de simplicité lors de l'affichage graphique. Par contre, le B-Rep a tendance à prendre beaucoup d'espace pour le stockage spécialement dans le cas des modèles complexes. Aussi, même s'il permet la visualisation d'objet 3D comme s'il s'agissait de solide, le B-Rep ne gère pas réellement des solides, mais seulement leur frontière. Pour cette raison, le calcul de certaines analyses spatiales topologiques pourrait s'avérer complexe. Toutefois, lorsque le B-Rep est orienté, i.e. l'intérieur et l'extérieur des objets sont distingués, le B-Rep se rapproche beaucoup de la gestion d'objets volumiques. Il est aussi relativement facile de calculer le volume d'un B-Rep à partir des faces composant sa surface lorsque celles-ci sont planes (De La Losa, 2000).

Structures de modélisation basées sur des solides

La structure de modélisation basée sur des solides consiste en la représentation d'un solide par ses frontières, mais aussi par son intérieur. En effet, cette approche implique la création d'objets par un assemblage de primitives volumiques simples de manière à remplir

l'intérieur des objets en plus de représenter leur frontière. De cette façon, il n'existe plus d'ambiguïté tant au niveau de l'interprétation visuelle qu'au niveau de la connaissance de l'intérieur et de l'extérieur des objets. Dans une telle structure, la représentation des solides doit être suffisamment complète pour faciliter le calcul automatique et non ambigu des propriétés géométriques bien connues comme le volume ou le centre de masse (Requisha et Voelcker, 1982).

Constructive Solid Geometry (CSG)

Il existe plusieurs structures géométriques basées sur des solides. La plus populaire, le CSG, permet la construction de modèles par l'application d'opération d'union, d'intersection et de différence sur des solides élémentaire paramétrable tels que des cubes, sphères, cylindres, cônes, etc. Ces solides élémentaires sont générés à partir de formes de base (cercles, rectangles, carrés...) et de certains paramètres de construction (diamètre, largeur, hauteur...) puis sont déplacés dans l'espace par des translations et des rotations. Les solides élémentaires utilisés pour la construction d'un modèle, leurs paramètres, leurs transformations ainsi que la séquence des opérations appliquées sont généralement stockés sous la forme d'un *Graph* ou d'un arbre, soit une structure de données hiérarchique composée de nœuds (les primitives géométriques) et de relations entre eux (les opérations).

Le CSG est une structure relativement simple à implémenter. L'étanchéité ou la fermeture du solide final créé est aussi facilement atteinte lorsque l'ensemble des primitives le composant le sont elles aussi. Par contre, étant donné la géométrie des solides de base mis à disposition, il peut être très difficile de représenter des objets irréguliers comme par exemple le relief du territoire. Le CSG est bien adaptée lorsque l'on veut offrir à l'utilisateur, la possibilité de construire son modèle de façon intuitive par combinaison et assemblage de primitives volumiques de base. Toutefois, même si plusieurs systèmes permettent de construire des solides en se basant sur le concept du CSG, cette approche seule n'est souvent pas suffisante puisque la forme des objets à représenter est complexe et ne peut pas alors se limiter à un assemblage de primitives de base (Ramos, 2003; Haala *et al.*, 2006; Tsuzuki *et al.*, 2007). En effet, la représentation d'un objet 3D suivant cette structure ne correspond pas à la frontière explicite de l'objet, aucune information sur la connectivité entre les primitives ainsi assemblées n'est présente, et sur l'existence même du solide ainsi

créé en tant qu'entité à part entière puisque la définition de l'objet ne repose uniquement que sur son *Graph*. Pour ces raisons, il est souvent nécessaire dans les systèmes actuels de combiner la structure CSG à celle du B-Rep et diverses méthodes sont présentées pour calculer les frontières d'un solide CSG (Requicha, 1996; Benouamer et Michelucci, 1997; Tsuzuki *et al.*, 2007; Hoffmann, 1992; Ramos, 2003).

Primitive Instancing

Une autre structure géométrique, appelée Primitive Instancing, adopte aussi une approche constructive comme le CSG, mais utilise plutôt des primitives complexes paramétrables pour représenter des objets. Ce type de modélisation consiste à instancier la primitive générique complexe de base et d'adapter cette instanciation en faisant varier des paramètres prédéfinis. Par exemple, dans un ensemble d'engrenage, chacune des pièces sera créée à partir d'une roue de base en spécifiant sa largeur, son rayon et le nombre de dents. Cette approche peut être avantageuse lorsque plusieurs objets du même type doivent être créés. Par contre, si les objets à modéliser sont tous très différents, l'utilisation de cette approche n'est plus justifiée puisque chaque objet demande la création d'une nouvelle primitive (de la Losa, 2000).

Voxel et Octree

La structure géométrique basée sur des solides appelée Voxel consiste en le découpage de façon régulière de la totalité de l'espace à l'aide de cubes élémentaires nommés voxels ("*volumetric elements*") d'une manière semblable au modèle raster et au pixel 2D. Chacun de ces voxels composant l'espace de modélisation, se retrouve soit occupé ou inoccupé par une partie du solide. Le solide ainsi représenté consiste donc en l'ensemble des voxels occupés de l'espace tridimensionnel. Ce modèle permet difficilement de représenter exactement la forme des objets, mais plutôt des approximations qui sont dépendantes de la taille et de la quantité de voxels mise en œuvre. Une grande quantité de voxels plus petits permettra de représenter plus exactement les limites des objets, mais générera un volume de données plus grand. C'est pourquoi cette structure géométrique est généralement considérée comme très gourmande en espace disque et en mémoire. L'analyse spatiale ne peut pas non plus être très élaborée puisqu'elle ne peut compter que sur l'adjacence des

voxels. Les calculs de volumes sont par contre très simples et peuvent être réalisés en faisant la somme des voxels contenus dans les objets.

Afin de diminuer la quantité de données d'un modèle de voxels, on opte souvent pour la structure appelée Octree qui permet de regrouper des voxels adjacents faisant partie de la même entité pour former des cubes plus gros. L'espace se retrouve donc découpé en cubes de tailles variables qui sont stockés dans un arbre possédant une racine (représentant l'espace total), des nœuds (cubes découpés en huit) et des feuilles stockant l'information « espace occupé » ou non. Ceci permet donc une certaine forme de compression tout en facilitant l'accès aux données proches en se basant sur le modèle du Quadtree en 2D (de la Losa, 2000).

Ensemble de solides

Finalement, viennent les modèles représentant des volumes par assemblage de solides irréguliers i.e. de polyèdres. Dans ce genre de modèle, l'assemblage de solides peut se faire au niveau de l'objet tout comme il peut obliger la discrétisation de l'ensemble de l'espace de modélisation comme dans le cas du Tetrahedral Network (TEN) (Pilouk, 1996). Les polyèdres sont assemblés de manière à partager des frontières communes et peuvent être de tailles fixes ou irrégulières. Le tétraèdre est un exemple de ce type de cellule simple qui peut être utilisé dans ce modèle. Les volumes sont alors peuplés de tétraèdres qui eux sont construits à partir d'un ensemble de points qui ne sont pas nécessairement associés sémantiquement à la géométrie de l'objet (Pouliot *et al.*, 2006). En modifiant la densité de ces points, on peut donc affiner la forme des objets sans devoir ajouter une grande quantité de cellules comme c'est le cas lors de l'utilisation de cubes de taille régulière. Cette méthode implique de devoir différencier les faces des polyèdres situées aux frontières des solides afin de pouvoir distinguer les différents volumes entre eux et entre l'espace de modélisation ne faisant pas partie d'un volume.

Ce type de modèle offre comme avantage la possibilité de rendre explicite les relations de voisinage entre les objets (car les frontières font partie intégrante du procédé de modélisation). Cela a pour effet de rendre très intéressant ce type de modèle pour effectuer de l'analyse spatiale topologique en plus de permettre de calculer facilement le volume des objets. Toutefois, remplir les solides de polyèdres ajoute à la complexité de la modélisation

en impliquant une plus grande quantité de calculs et d'information à gérer ce qui peut également nuire à la performance de ces modèles. Il existe plusieurs propositions pour la modélisation de solides irréguliers tels que le 3D Formal Data Structure (3D FDS) (Molenaar, 1990), le Simplified Spatial Model (SSM) (Zlatanova, 2000b) le Urban Data Model UDM (Coors, 2003), le Object Oriented 3D (OO3D) (Shi *et al.*, 2003) ou le Geological TEN (GeoTEN) plus spécifique au domaine géologique (Lachance, 2005). Une particularité importante de ces propositions est qu'elles définissent tous des structures topologiques 3D.

2.2.2 La standardisation des structures géométriques

L'Open Geospatial Consortium (OGC) et le comité ISO/TC 211 de l'International Standard Organisation (ISO), sont les deux principaux organismes responsables de la standardisation dans le domaine de l'information géographique. En 1997, l'OpenGIS Consortium (maintenant OGC) publiait la spécification "*OpenGIS Simple Features Specification*" (SFS). Cette spécification décrit une interface standard permettant à divers systèmes de communiquer en termes de "*Simple Features*" i.e. d'entités géométriques simples 2D. Ces entités sont les points, lignes, "*linestrings*", courbes et polygones définis de manière standard et dont on associe à un système de référence géographique. Les recommandations de cette spécification ont été implémentées dans bon nombre d'applications spatiales comme PostGIS, FME, ArcGIS, etc... et encore aujourd'hui, le SFS représente un des modèles géométriques les plus répandus au niveau des applications gérant des données géométriques 2D.

Plus récemment, l'ISO et l'OGC ont présenté un schéma spatial conceptuel qui décrit d'une manière beaucoup plus complète, les caractéristiques spatiales des entités géométriques et opérations spatiales pouvant s'appliquer sur celles-ci (ISO 19107, 2002). Ce standard international appelé schéma spatial ISO 19107 est équivalent au "*Topic 1: Feature Geometry*" de la spécification "*The OpenGIS Abstract Specification*" de l'OGC. Avec cet effort commun, l'ISO et l'OGC mettent à disposition un modèle standard pouvant servir de guide lors de l'implémentation de la plupart des structures géométriques vectorielles dans une application spatiale. En effet, alors que le SFS se consacrait aux entités simples 2D, le schéma spatial de l'ISO présente un inventaire beaucoup plus complet d'entités

géométriques prenant en charge, en outre, la topologie et le 3D. Pour simplifier sa compréhension, son implémentation et aider à faire le pont avec la spécification SFS encore largement répandue, le schéma spatial ISO 19107 a été décomposé en différents profils. L'un de ces profils, le "*Simple Features*" peut être considéré comme le successeur du SFS et contient un sous-ensemble des entités du schéma spatial ISO 19107 portant sur les entités géométriques 2D non-topologiques. Il est donc important de faire attention au terme "conforme ISO 19107" dans la description de certaines applications, car cela ne signifie pas nécessairement qu'elles permettent la gestion de modèles 3D ou de modèles topologiques. La majorité des systèmes actuels qui implémentent ce schéma spatial sont effectivement conforme qu'en partie puisqu'ils ne gèrent entre autres que les primitives ponctuelles, linéaires et surfaciques 2D, soit le profil "*Simple Features*" (ex : GeoServer, Deegree, GeOxygene, RedSpider). En annexe A, on présente le schéma conceptuel du schéma spatial ISO 19107. Pour ce qui est des structures géométriques de type matriciel (orientées-espace), l'ISO propose aussi une solution pour leur standardisation dans la spécification : "*Schema for coverage geometry and functions*" sous la bannière ISO 19123.

Sans présenter en détail l'ensemble de la norme ISO 19107, les paragraphes suivants présentent les éléments importants qui touchent plus particulièrement le 3D et qui sont susceptibles d'entrer dans la composition des structures géométriques.

La norme ISO 19107 propose la définition des primitives de base point (GM_Point), courbe (GM_Curve), surface (GM_Surface) et solide (GM_Solid) sur lesquelles repose la construction des modèles géométriques. En langage ISO, une primitive est un objet qui n'est pas décomposable en d'autres primitives dans le système. Les courbes, surfaces et solides sont donc considérés comme des primitives au sein de cette norme même si théoriquement ils sont décomposables en primitive plus simple (ce sont des sous-classes de GM_Primitive). Le GM_Solid est l'élément de base de la géométrie tridimensionnelle de l'ISO. La norme ISO nous montre qu'un GM_Solid est construit à partir de ses frontières, soit un GM_SolidBoundary, définit lui-même par une enveloppe intérieure et extérieure à la manière du B-Rep. Les enveloppes intérieures et extérieures d'un GM_SolidBoundary sont représentées par des coquilles fermées (GM_Shell). De son côté, un GM_Shell consiste en un certain nombre de références à des surfaces orientées

(GM_OrientableSurface) connectées entre elles dans un cycle topologique i.e. un objet fermé et sans frontière. Chacune des GM_OrientableSurface qui composent un GM_Shell est formée d'une surface (GM_Surface) et de son orientation (+, -). Avec ces éléments, la norme de l'ISO permet donc de représenter de manière standard l'ensemble des structures géométriques basées sur des points et des lignes et celles basées sur des surfaces.

Pour ce qui est des structures géométriques basées sur des solides, la norme ISO met aussi à disposition les éléments nécessaires à leur représentation. Par exemple, la norme présente plusieurs primitives volumiques telles que GM_Sphere, GM_Cylinder ou le GM_Cone qui peuvent être utilisées pour représenter de manière standard les primitives entrant dans la composition des modèles constructifs comme le CSG ou le Primitive Instancing. Si une primitive volumique n'existe pas dans la norme, elle peut être construite par l'assemblage d'éléments de dimension inférieure pour former un GM_Solid (ex : un cube peut être créé par l'assemblage de surfaces) ou à l'aide des éléments complexes de la norme, soit les composites et les agrégats. En effet, les primitives géométriques de la norme ISO sont des éléments de base qui peuvent être assemblés pour former des modèles géométriques ou des primitives plus complexes telles qu'un GM_Composite ou un GM_Aggregate. Un GM_Composite est un élément complexe dont la géométrie sous-jacente peut être traitée comme une simple primitive. Cet assemblage est aussi restreint à un seul type de géométrie. Par exemple, un GM_CompositeSurface est une collection de surfaces orientées dont leurs frontières communes reposent sur les mêmes *curves* et qui lorsque considérée comme un tout forme une simple surface. D'une manière semblable, un GM_CompositeSolid est une collection de solides assemblés en paires à partir de surfaces communes composant leurs frontières. Cette collection de solides est un élément complexe qui possède les mêmes caractéristiques qu'une primitive de type solide. De leur côté, les GM_Aggregate (GM_MultiPoint, GM_MultiCurve, GM_MultiSurface, GM_MultiSolid) sont aussi des assemblages de primitives de même type, mais dont les frontières peuvent être disjointes. En utilisant ces concepts, des modèles du type Ensemble de solides tels que le TEN ou le GeoTEN pourraient relativement facilement être représentés d'une manière standard à l'aide de la norme ISO. Bien qu'il n'existe pas de primitives volumiques de base telles que le tétraèdre nécessaire à la construction de ces modèles, il est possible à l'aide de cette norme de représenter de tels modèles en tirant profit des composites. En effet, un

modèle tétraédrique est typiquement composé d'un ensemble de solides formés eux-mêmes par un assemblage de tétraèdres. Un tel modèle pourrait être représenté selon les concepts de l'ISO par une collection de GM_CompositeSolid dont chacun d'eux est formé par une collection de solides soit les tétraèdres qui sont eux-mêmes construits à la manière du B-Rep par l'assemblage de quatre surfaces triangulaires. La figure suivante montre les trois niveaux d'agrégation d'un modèle tétraédrique soit le modèle lui-même, les solides composites qui le composent et finalement les tétraèdres qui forment ces solides.

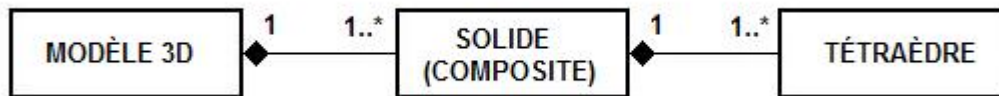


Figure 1: Les niveaux d'agrégation d'un modèle tétraédrique.

La présence d'une primitive volumique comme le GM_Solid, des classes connexes pour la construction de solides à la manière du B-Rep (GM_SolidBoundary, GM_Shell...) ainsi que la possibilité d'assembler ces primitives en modèle plus complexe (GM_Composite, GM_Aggregate, GM_CompositeSolid, GM_MultiSolid...), font du schéma spatial de l'ISO 19107 une norme générique bien adaptée pour la représentation des différentes structures géométriques 3D et pour la modélisation géométrique 3D en général.

2.3 La gestion des données et des modèles 3D

La gestion des modèles géométriques 3D au sein d'une application spatiale implique deux choses, le stockage du modèle et l'ensemble des processus destinés à l'administration des données et des traitements au sein du système. La persistance des modèles sur support numérique passe par l'implémentation d'une structure de stockage qui viendra définir les bases de l'organisation physique des données. Cette structure (à ne pas confondre avec la structure géométrique) peut être implémentée au sein d'un des différents médias de stockage disponible tel qu'une base de données ou un fichier. Pour sa part, la représentation et la manipulation du modèle au sein de l'application passent par une structure géométrique comme celles présentées plus tôt dans ce chapitre. L'application doit donc pouvoir accéder à l'information (le modèle) stockée sur support numérique et la traduire en une structure

géométrique compréhensible et supportée par le programme dans le but de la manipuler et d'éventuellement l'analyser et la diffuser. Ainsi, la gestion des données géométriques inclut l'ensemble des fonctionnalités chargées de créer, maintenir, traiter et interroger une collection de données géométriques que ce soit au niveau du programme lui-même ou de la base de données. La structure géométrique implémentée au niveau du programme et la structure de stockage du modèle sont toutefois deux niveaux structurels distincts dont un système qui gère des données géométriques devrait prendre en charge. Pour bien faire la différence entre les deux, on peut prendre l'exemple d'un logiciel comme Autocad et de sa structure de stockage i.e. le format de fichier d'extension .DWG. Comme Autocad permet entre autres de créer des modèles selon la structure géométrique du CSG, toute information nécessaire pour recréer le modèle CSG au sein de ce logiciel se retrouve codifiée, structurée et stockée au sein d'un fichier .DWG lors d'une sauvegarde. Lorsqu'un usager d'Autocad ouvre ce fichier pour importer son modèle, l'information contenue dans le fichier est interprétée et traduite en un modèle dans la structure géométrique du CSG au sein du programme soit généralement sous la forme d'un arbre binaire.

La persistance des données géométriques implique la codification selon un format prédéfini des propriétés géométriques des primitives formant le modèle. Par exemple, le stockage d'un modèle par balayage pourrait impliquer l'enregistrement des coordonnées formant les primitives 2D ponctuelles, linéaires ou surfaciques avec pour chacune d'elles, une valeur correspondant à la hauteur d'extrusion. Le stockage de cette information dans une application spatiale peut se faire principalement de deux façons soit à l'intérieur d'un fichier ou d'une base de données interne ou externe à l'application.

Le stockage par fichier est la méthode qui vient par défaut avec la plupart des logiciels SIG. Bien que très simple, elle nécessite de parcourir en entier le fichier lors de recherche pour trouver les éléments désirés. On lui reproche aussi certains problèmes de redondance, de maintenance et d'intégrité des données. De plus, un fichier étant accessible en écriture que par une seule personne à la fois, le stockage par fichier est généralement plus approprié au travail individuel et dans les petites organisations.

De son côté, la gestion par base de données permet d'éliminer ces contraintes et permet de porter à un niveau supérieur la gestion des données à l'intérieur des SIG en la faisant passer

du niveau simple usager à celui institutionnel. En effet, la gestion d'une base de données est généralement prise en charge par un SGBD tel Oracle, DB2, PostgreSQL, SQL Server, etc. et présente de nombreux avantages tels que l'accès multi-usagers, la sécurité et le maintien de l'intégrité des données. Afin de permettre aux usagers de SIG de bénéficier de ces avantages, les fournisseurs de SIG et de SGBD ont tous deux fait évoluer leurs produits dans un effort commun d'intégration. Les fournisseurs de SIG ont développé des fonctionnalités, des plug-ins ou des produits intermédiaires (*middlewares*) pour se brancher aux principaux SGBD alors que ceux-ci ont été « spatialisés » par l'ajout de la gestion de primitives géométriques et d'opérateurs spatiaux.

D'une manière générale, la façon dont sont stockées les données dans un SGBD dépend du type de SGBD utilisé. Dans un SGBD relationnel, les données sont stockées au sein de lignes et de colonnes de tables relationnelles. Chaque colonne d'une telle table se voit attribuer un type de donnée tel que entier, texte, date, etc., et chaque table possède sa clé primaire lui permettant d'identifier de manière unique les objets qu'elle contient et de faire le lien avec les autres tables du système. L'information composant un modèle géométrique tel que les primitives, les liens entre elles et les coordonnées x,y,z qui les composent, peut donc être stockée de manière ordonnée au sein d'un tel système. Typiquement, une « couche » spatiale est représentée par une table dont chaque enregistrement (ligne) représente un objet spatial dont les propriétés géométriques et descriptives incluant les coordonnées sont stockées au sein de colonnes. Afin d'améliorer la performance lors de l'accès aux données, les coordonnées de chaque objet spatial sont indexées. L'indexation est la technique qui permet de retrouver plus rapidement certains éléments d'une table à l'aide d'un index calculé à partir des valeurs d'une ou de plusieurs colonnes. Au niveau spatial, cette indexation réalisée par le biais de techniques comme le R-Tree, le Quadtree ou l'Octree, permet d'optimiser les requêtes spatiales et de rechercher plus rapidement à travers les éléments d'un modèle en tenant compte des propriétés géométriques telles que la distance entre ces éléments ou les coordonnées de leur rectangle ou boîte englobante (Samet, 1995).

Dans un SGBD orienté-objet tel qu'ObjectStore, Poet FastObjects, Versant Developer Suite ou Objectivity, les propriétés géométriques telles que les coordonnées ne sont plus stockées

comme des valeurs numériques mais peuvent être stockées comme des objets complexes créés par le biais d'un des langages de programmation orienté-objet supporté par le SGBD. Ainsi, des objets de type point, ligne ou polygone peuvent être créés pour encapsuler les coordonnées décrivant la primitive géométrique ainsi que d'autres informations telles que le système de référence géographique.

Sur le marché, la grande majorité des SGBD dit spatiaux sont des solutions plutôt hybrides. En effet, les produits tels qu'Oracle Spatial, DB2 Spatial Extender, PostGIS, SQL Server Katmai, etc. sont des modules venant se greffer à un SGBD relationnel lui conférant des capacités de gestion des objets géométriques à la manière orientée-objet. Ces SGBD spatiaux permettent donc d'ajouter au sein de tables relationnelles une colonne de type « géométrie » venant encapsuler et stocker les propriétés géométriques d'un point, d'une ligne ou d'un polygone au même titre que les types entier, texte, date, etc. Ces SGBD implémentent aussi des index spatiaux ainsi que plusieurs opérateurs spatiaux leur permettant ainsi d'interroger et de manipuler spécifiquement ces objets

Cette « spatialisation » des SGBD a été réalisée dans la majorité des cas en implémentant une spécification dérivée du SFS soit la spécification "*OpenGIS Simple Features Implementation Specification for SQL*". Cette spécification décrit un schéma SQL standard destiné au SGBD, pour le stockage, la récupération, le requêtage et la mise à jour des entités géométriques simples 2D (vectoriels). Le stockage des modèles matriciels (pixels) étant fondamentalement différent, les fournisseurs de SGBD spatiaux ont aussi développé des techniques spécifiques pour leur gestion. D'une manière semblable au modèle vectoriel, des types géométries propres au modèle matriciel ont été créés pour leur stockage, leur indexation et leur requêtage au sein des SGBD spatiaux (par exemple GeoRaster dans Oracle Spatial).

Jusqu'à tout récemment (depuis Oracle spatial dans sa version 11.1g), les SGBD spatiaux n'offraient pas encore de primitives volumiques au sein de leur système. Le stockage des modèles 3D s'effectue donc encore aujourd'hui presque entièrement au sein de fichiers. Nous verrons toutefois plus loin dans ce chapitre que des études ont été réalisées et qu'il existe des méthodes pour contourner cette lacune.

2.4 L'analyse spatiale 3D

La 3^e composante fondamentale identifiée pour constituer un SIG correspond à ses capacités d'analyse spatiale.

2.4.1 Rappel des concepts

D'une manière générale, l'analyse spatiale peut être définie comme un raisonnement qui permet de déduire les caractéristiques d'un phénomène en faisant intervenir des données géographiques (OQLF, 2004). Les aspects géométriques et topologiques des représentations spatiales permettent de distinguer l'analyse spatiale, d'une analyse conventionnelle (mathématique, statistique ou logique par exemple). Plusieurs publications ont été faites sur ce sujet largement couvert dans le monde du SIG (Anselin 1995; Fotheringham *et al.*, 2000; Goodchild 2000; Longley *et al.*, 1999).

L'analyse spatiale peut d'abord se réaliser visuellement, à partir de représentations, d'univers 2D (une carte par exemple) ou 3D (un modèle 3D par exemple). En exploitant ces vues, l'analyse spatiale permet d'examiner les objets, les patrons et certains de leurs aspects visuels. Même si la visualisation des données spatiales peut être utile pour obtenir certaines informations et pour effectuer de l'analyse qualitative, pour la plupart des applications, une analyse quantitative est nécessaire (Verbree *et al.*, 2005). Ainsi, à partir de traitements spécifiques appliqués sur des données géométriques, l'analyse spatiale peut permettre d'identifier et de calculer de nouvelles informations qui seraient difficiles ou impossibles de déduire sans ces opérations. On parle alors de l'application d'opérateurs métriques, topologiques, arithmétiques ou logiques. Les opérateurs métriques permettent le calcul de la position, de la forme, de l'aire, du périmètre, du volume ou de la distance des objets géométriques alors que les opérateurs topologiques font plutôt références aux relations spatiales (ou de voisinages) entre ceux-ci et peuvent être de cinq types : proximité, intersection, inclusion, égalité et adjacence (Cromley, 1992). Par opérateurs logiques on entend les fonctions classiques du type VRAI, FAUX, OUI, NON, ET, OU, >, <, =. Les opérateurs arithmétiques sont ceux utilisés pour modéliser les opérations de calculs mathématiques telles que l'addition (+), la soustraction (-), la multiplication (*) et la division (÷). Ces différents opérateurs sont appliqués sur les modèles géométriques et l'implémentation des algorithmes sous-jacents dépend directement de la structure

géométrique. Par exemple, le calcul d'adjacence entre deux objets géométriques sera effectué différemment dans un modèle de type B-Rep que dans un modèle Octree.

On peut ainsi distinguer l'analyse spatiale selon qu'elle s'effectue dans un univers 2D (système de coordonnées planimétriques X,Y) ou 3D (système de coordonnées X,Y,Z), selon la dimension des objets examinés (0D à 2D, axiomes de la géométrie plane) et 3D (axiomes de la géométrie des solides) ou encore selon le type d'analyse topologique effectué (2D ou 3D). Notons que l'analyse spatiale 3D ne peut se réaliser qu'en exploitant des systèmes informatiques gérant de manière explicite les coordonnées tridimensionnelles (ex. x,y,z) associées aux objets à représenter. Les objets localisés dans ces univers pourront posséder différentes dimensions (0D à 3D). La catégorie d'opérateurs à appliquer sera ici importante, dépendamment de la dimension de ces objets et c'est pourquoi nous discuterons plus en détail cet aspect particulier.

Le recensement des opérateurs spatiaux dans les SIG actuels montre que ceux-ci sont essentiellement 2D (Pilouk, 1996; Zlatanova, 2000b; Rahman, 2000; Zlatanova *et al.*, 2002; De La Losa, 2000). Ceci est dû au fait qu'ils soient basés sur une géométrie plane et que seules les coordonnées x et y sont prises en compte lors des calculs. Au mieux, ils permettent d'effectuer certaines analyses spatiales 3D comme des calculs d'intervisibilité entre deux points et des calculs d'orientation ou de pentes lorsque l'altitude des points x,y est connue (lorsqu'on dispose d'un modèle 2.5D). Toutefois, la majorité des opérateurs spatiaux courants tels que l'intersection, l'analyse de proximité ou d'inclusion, les calculs d'aire et de distance sont destinés à être appliqués aux modèles 2D (planaires) traditionnels des SIG et ont donc été conçus en conséquence i.e. sans la prise en compte de la 3^e dimension.

Pour ces raisons, et à moins d'une programmation spécifique supplémentaire, une opération spatiale dite 2D appliquée sur un modèle 3D ne fournira pas le résultat attendu puisque l'algorithme ne tiendra pas compte de la troisième dimension dans les calculs, seules les coordonnées x,y seront traitées. Par exemple, l'aire calculée par un opérateur 2D d'une surface plane 3D perpendiculaire au plan horizontal serait nulle puisque la dite surface, en projection plane (celle utilisée par l'opérateur 2D) sera perçue comme une ligne et non pas une surface. Ainsi, l'analyse spatiale est dite 3D lorsqu'elle tiendra compte des trois

dimensions des données géométriques lors des calculs. Dans le cas de l'analyse spatiale vectorielle 3D, on parlera d'analyse portant sur des ensembles discrets de points, lignes, polygones, mais aussi de volumes tandis que du côté matriciel cette analyse porterait sur des grilles continues de voxels.

2.4.2 Standardisation des opérateurs spatiaux

Il existe très peu de normes aidant à la standardisation des opérateurs d'analyse spatiale (mise en œuvre et application). Le standard y portant une attention particulière est la spécification "*OpenGIS Simple Features Implementation Specification for SQL*" de l'OGC et vise le traitement des données 2D au sein d'une base de données. Cette spécification définit huit méthodes pour tester les relations spatiales topologiques entre objets spatiaux simples basées sur le modèle des neuf intersections d'Egenhofer (Egenhofer *et al.*, 1990) et sept opérateurs spatiaux tels que le calcul de distance, de l'union ou de la différence entre objets spatiaux. D'autres méthodes comme le calcul de l'aire ou du centroïde d'un polygone sont aussi abordées dans cette spécification. L'approche amenée par Egenhofer pour évaluer la relation entre deux géométries consiste à tester l'intersection de l'intérieur, de la limite et de l'extérieur des géométries pour ensuite classifier la relation selon la matrice d'intersection résultante. Théoriquement, le modèle d'Egenhofer montre qu'il existe 512 (2^9) types de relations spatiales, mais seulement un petit groupe de celles-ci peut être apprécié dans la réalité. L'OGC en définit cinq comme étant les relations spatiales de base avec leurs matrices d'intersection correspondantes (Disjoint, Touches, Crosses, Within, Overlaps) et en ajoute trois autres à des fins d'accommodement (Contain, Intersects, Equal). Le tableau suivant résume l'ensemble des méthodes pour tester les relations spatiales telles que définies par l'OGC :

Relation	Description
Equals	spatially equal to: $a=b$
Disjoint	spatial disjoint: equivalent to $a \cap b = \emptyset$
Intersects	spatially intersects: $[a \cap b]$ is equivalent to $[\text{not a disjoint}(b)]$
Touches	spatially touches: equivalent to $[a \cap b = \emptyset \text{ and } (a \cap b) \neq \emptyset]$; does not apply if a and b are points
Crosses	spatially crosses: equivalent to $[\dim(I(a) \cap I(b)) < \max\{\dim(I(a)), \dim(I(b))\}]$ and $a \cap b \neq \emptyset$ and $a \cap b \neq b$

Relation	Description
Within	spatially within: within(b) is equivalent to [$a \cap b = b$ and $I(a) \supseteq I(b)$]
Contains	spatially contains: [a contains(b)] is equivalent to [b within(a)]
Overlaps	spatially overlaps: equivalent to [$\dim(I(a) \cap I(b)) = \dim(I(a)) = \dim(I(b))$ and $a \cap b \neq a$ and $a \cap b \neq b$]
Equals	spatially equal to: $a=b$

Tableau 1: Les types de relations spatiales du SFS for SQL extraits de OGC, 1999.

Ces méthodes pour tester les relations spatiales entre objets spatiaux simples telles que présentées et standardisées par l'OGC sont appliquées à des objets spatiaux ponctuels, linéaires et surfaciques dans un univers 2D. Plusieurs projets de recherche se sont attardés sur la clarification des relations entre objets spatiaux dans un univers 2D mais très peu se sont penchés sur les relations spatiales dans un univers 3D (Zlatanova, 2000). Quelques projets de recherche ont montré que le modèle d'intersection d'Egenhofer peut être utilisé pour déterminer les relations spatiales entre objets spatiaux ponctuels, linéaires, surfaciques, mais aussi volumiques dans un univers 3D (Bric, 1993; Egenhofer, 1995; Zlatanova, 2000). L'introduction de la 3^e dimension change la donne au niveau des cas possibles de relation spatiale entre les objets spatiaux et les différents auteurs ne s'entendent pas tous sur leur quantité. Dans Zlatanova, 2000, le modèle des neuf intersections d'Egenhofer est appliqué pour déterminer les relations spatiales possibles entre objets spatiaux 3D et 25 contraintes sont introduites afin d'en éliminer les cas impossibles. En appliquant ces contraintes, 69 types de relations spatiales seraient possibles en 3D et devraient donc être considérés lors de l'implémentation d'une application offrant des capacités d'analyse spatiale 3D.

Ces relations possibles peuvent théoriquement être regroupées sous l'une ou l'autre des huit appellations standards définies par l'OGC (Disjoint, Touches, Crosses, Within, Overlaps, Contain, Intersects, Equal) bien que dans quelques situations, ces noms établis pour les relations en 2D peuvent paraître inappropriés pour certaines relations entre objets 3D (Zlatanova, 2000). Par exemple, dans la figure 3, il n'est pas nécessairement évident de distinguer qu'une surface pénétrant partiellement à l'intérieur d'un solide est une relation

de chevauchement. Pourtant, nous sommes bien en présence d'un chevauchement puisque nous avons une intersection partielle entre l'intérieur des deux objets.

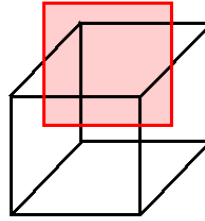


Figure 2: Chevauchement entre une surface et un solide.

Un SIG 3D pourrait donc implémenter ces huit méthodes standards à l'aide d'algorithmes validant l'ensemble des relations possibles entre objets ponctuels, linéaires, surfaciques et volumiques dans un univers 3D.

En plus de ces huit méthodes, l'OGC définit aussi sept opérateurs spatiaux dans la spécification SFS qui permettent de retourner une distance ou une nouvelle géométrie suite à un calcul. Le tableau suivant montre et décrit ces sept opérateurs :

Opérateur	Description
Distance	the shortest distance between any two points in the two geometries as calculated in the spatial reference system of this geometry
Buffer	all points whose distance from this geometry is less than or equal to a specified distance value
Convex Hull	the convex hull of this geometry
Intersection	the point set intersection of the current geometry with another selected geometry
Union	the point set union of the current geometry with another selected geometry
Difference	the point set difference of the current geometry with another selected geometry
Symmetric difference	the point set symmetric difference of the current geometry with another selected geometry (logical XOR)

Tableau 2: Les opérateurs spatiaux standards du SFS for SQL extraits de OGC, 1999.

Encore une fois, la spécification SFS suggère l'implémentation de ces opérateurs avec comme objectif d'être appliqués sur des objets spatiaux ponctuels, linéaires et surfaciques dans un univers 2D. Toutefois, l'ensemble de ces opérateurs pourrait théoriquement être implémenté pour traiter aussi des points, des lignes, des polygones et des volumes dans un

univers 3D. Par exemple, l'opérateur Intersection pourrait très bien être utilisé pour calculer l'intersection d'un solide avec une surface ou un autre solide.

Avec cette spécification, l'OGC définit les méthodes et opérateurs de base permettant de couvrir la grande majorité des principaux traitements d'analyses spatiales vectorielles que l'on retrouve dans les SIG. Nous pensons donc qu'un SIG 3D implémentant ces méthodes et opérateurs, avec la logique 3D sous-jacente qui s'impose i.e. les algorithmes prenant en charge la 3^e dimension, serait très bien outillé pour répondre à la majorité des demandes des usagers en terme d'analyse spatiale tridimensionnelle. Toutefois, l'implémentation des différents algorithmes 3D de ces opérateurs spatiaux représente un travail colossal d'autant plus qu'il n'existe pas de bibliothèque ou d'API complète d'algorithmes d'opérateurs spatiaux 3D standards comme c'est le cas en 2D avec des produits comme la Java Topology Suite (JTS) ou les bibliothèques propriétaires des différents logiciels SIG.

2.5 Le partage des modèles 3D

Dans un objectif de partage de connaissance et d'échange d'information, il est important d'examiner les options qui s'offrent à nous pour faciliter l'accès aux modèles 3D i.e. les rendre disponible pour leur visualisation ou leur importation. Nous allons donc passer en revue les différentes technologies et stratégies de partage des données géométriques 2D présentement employées afin de vérifier si elles sont prêtes à supporter les modèles 3D. Nous accorderons une attention particulière aux technologies orientées vers Internet, média de communication par excellence, et aux standards entrants dans le processus de diffusion afin de vérifier où ils en sont par rapport au 3D.

Il existe différentes façons de transmettre des modèles géométriques. La plus simple, la méthode manuelle consiste à produire un document (papier ou numérique) ou sauvegarder l'information sur un support numérique comme un CD ou un DVD et le faire parvenir à destination. Avant l'ère des nouvelles technologies, c'était souvent la seule méthode disponible. Avec Internet et les réseaux locaux, il est maintenant beaucoup plus simple de transmettre l'information, et ce, de manière plus rapide et à un plus grand nombre d'usagers. L'internet est le moyen de communication par excellence et nous offre plusieurs solutions pour diffuser et distribuer des modèles que ce soit par courriel, par le biais d'une

page Web ou de manière plus dynamique par l'intermédiaire d'une application en ligne. En effet, le Web regorge maintenant de sites Web offrant la diffusion d'information cartographique dynamique par le biais d'applications ou d'API de cartographie en ligne telles que ArcIms, MapGuide (Enterprise ou Open Source), MapXtreme, MapX, Google Map, Google Earth, ArcGIS Explorer, Virtual Earth, Nasa World Wind etc. Le dynamisme est certainement l'élément responsable du succès de ces applications et ce qui les rend particulièrement intéressantes. Une vue statique d'un modèle géométrique comme par exemple une carte peut être suffisante pour un certain nombre d'applications, mais cette vue n'offre certes pas la richesse d'une navigation dynamique et libre au travers d'un modèle. Le dynamisme de ces technologies est dû en grande partie au concept de service web et au principe du client-serveur.

2.5.1 Les services Web

Un service Web peut se définir comme un composant applicatif accessible sur le Web, par l'entremise d'une interface standard ou non, qui peut interagir dynamiquement avec d'autres applications en utilisant des protocoles de communication basés sur le XML, et cela, indépendamment du système d'exploitation et des langages de programmation utilisés (OQLF, 2004). Dans le cadre d'un service de diffusion cartographique en ligne, le service Web permet de recevoir sur demande suite à une requête paramétrée, une carte sous forme d'image créée à partir d'un serveur hébergeant le modèle géométrique. Le client (par exemple un navigateur internet) est l'application à partir de laquelle un usager construit et envoie sa requête au serveur par le biais du service Web. Le serveur est l'ordinateur contenant l'application permettant de recevoir les requêtes des clients, d'effectuer les traitements nécessaires afin de générer l'extrait offert par le service Web et transmettre la réponse au client. Les services web sont utilisés en général pour rendre accessible sur le web des applications ou des données dans le cadre d'un contrat d'échange entre entreprise et client (Business to Consumer) ou entre entreprise et entreprise (Business to Business). Il existe présentement une grande quantité de services Web en tout genre implémentés pour un besoin bien spécifique entre deux organisations ou offerts au grand public par le biais d'annuaires en ligne (<http://www.webservicelist.com>, <http://www.wsindex.org>). Par exemple, deux organisations peuvent implémenter un service web pour copier sur demande, le contenu d'une base de données vers une autre dans un contrat d'échange d'information.

Une autre organisation peut offrir un service public de validation automatique de numéros de téléphone ou de codes postaux pour assurer aux clients de ce service une qualité et une intégrité d'information avant de la stocker.

2.5.2 Limites et solutions pour la diffusion en 3D

Ainsi, les services Web de cartographie en ligne sont des outils de prédilection pour les organisations désireuses de rendre accessible en temps réel de l'information sous forme de carte. Toutefois, les principaux produits offrant de telles capacités ne supportent pas le 3D, étant généralement basés sur des modèles stockés sous l'un des formats des principaux logiciels SIG eux-mêmes reposants sur des structures géométriques incompatibles au 3D. Au mieux, ils permettent la diffusion de modèle 2.5D comme ceux montrant le relief du territoire et les bâtiments. En fait, faute de moyens, on diffuse encore beaucoup sur le Web les modèles 3D sous forme d'image statique. La distribution de ces modèles repose encore aussi en grande partie sur le partage de fichiers. En effet, le recours au CD ou au DVD comme support numérique pour donner accès aux modèles 3D généralement très lourds demeure encore souvent le seul moyen à disposition pour partager cette information. Il existe toutefois quelques solutions intéressantes de diffusion de modèle 3D sur Internet. En effet, on retrouve certains langages comme le Virtual Reality Modeling Language (VRML) et le X3D, son successeur, ayant fait leur apparition il y a déjà quelques années, qui permettent la représentation (géométrique et graphique) et la manipulation de modèles 3D au sein d'un navigateur Web. Les spécifications de ces deux formats ont d'ailleurs été acceptées comme standards ISO. Plusieurs projets de recherche ont utilisé ces langages, couplés à des bases de données, CAO ou SIG, afin de diffuser sur Internet des modèles 3D (Coors et Jung, 1998; Zlatanova et Tempfli, 2000c; Aren *et al.*, 2003; Vries et Stoter 2003). Une alternative intéressante est aussi apparue plus récemment pour la diffusion des modèles 3D soit les fichiers PDF 3D. Des entreprises comme Visual Technology Services ou Adobe offrent des produits permettant de visualiser et manipuler des modèles 3D au sein d'un fichier PDF. Déjà, les développeurs d'outils CAO comme Bentley et GeoSoft commencent à exploiter cette technologie en permettant d'exporter des modèles 3D directement en PDF à partir de leurs produits. Une autre alternative très populaire serait le Keyhole Markup Language (KML), une grammaire XML utilisée pour coder conjointement les données géométriques avec leurs propriétés graphiques dans les visualisateurs de globe

virtuel en ligne tels que Google Earth, Microsoft Virtual Earth, etc. Ce format propriétaire de Google a récemment été adopté comme standard ouvert par l'OGC. Le KML fournit la syntaxe pour la représentation de primitives géométriques vectorielles ponctuelles, linéaires et surfaciques en permettant pour chacune d'elles le stockage des coordonnées x, y et z. Il permet aussi de stocker les propriétés d'images stockées localement ou sur un serveur distant afin de permettre leur drapage sur le relief du globe virtuel.

Bien qu'assez limitées, des solutions de diffusion de modèles 3D existent et ce domaine semble être en pleine croissance. D'ailleurs, du côté des standards de partage de l'information géométrique plus spécifique aux sciences géographiques, on commence aussi à évoluer vers le 3D. En effet, l'OGC propose de standardiser les interfaces de services Web destinés au partage de l'information géographique dans l'optique de favoriser l'interopérabilité entre les systèmes. Bien qu'au départ leur usage se limitait aux données 2D, leurs capacités ont depuis été élargies afin de supporter les données 3D. Ces standards et spécifications tels que le GML, WMS, WFS, WTS, etc. seront présentés plus en détail en mettant l'emphase sur leurs possibilités face au 3D. Nous présenterons aussi un aperçu du Web 3D Service (W3DS), le tout dernier des services Web de l'OGC, encore à l'étape de discussion, portant spécifiquement sur le partage de modèles 3D. Nous verrons ce que ce nouveau standard, à terme, pourrait apporter au 3D.

2.5.3 Le Geography Markup Language

Nous ne pouvons parler de partage de données sans aborder la notion de langage de communication. La tendance actuelle est d'exploiter des langages génériques de balise extensibles de type XML (eXtensible Markup Language) ou GML (Geography Markup Language). XML (eXtensible Markup Language) est un langage informatique de balisage générique supporté par le World Wide Web Consortium (W3C). Son objectif initial était de faciliter l'échange de données entre systèmes d'informations hétérogènes, dont principalement sur Internet (<http://www.w3.org>). Le XML est utilisé pour exprimer des langages de balisages spécifiques comme le XHTML, le SVG, ou le Geography Markup Language (GML) par exemple. Le GML est la grammaire XML, définie par l'OGC pour représenter d'une manière standard des entités géographiques. Le GML sert de langage de modélisation et format de stockage pour les systèmes d'information géographique et est

aussi utilisé comme format d'échange de données spatiales sur Internet (<http://www.opengeospatial.org>). C'est un standard ouvert, indépendant des technologies et non propriétaire. Le langage GML a été adopté comme standard international par l'ISO (sous la bannière ISO 19136) en 2007. Le GML n'est pas le seul langage utilisé pour décrire de l'information géographique, mais il est le premier à être accepté par la majorité des intervenants de la communauté des sciences géographiques et surtout, il est le seul à proposer une description des aspects sémantiques associés aux objets à modéliser. Étant bâti sur le XML, le GML profite de l'ensemble des technologies XML disponibles à ce jour comme par exemple, le schéma XML, le XLink, et le XPointer (Lake, 2004). De plus, par sa syntaxe très flexible, le GML permet de combiner facilement les données spatiales avec des données non spatiales pour ainsi rattacher de l'information aux entités géométriques. En effet, le GML permet de décrire la position, la forme et l'étendu d'un objet spatial mais aussi ses propriétés descriptives comme par exemple le type de sol, le type de roche, la densité ou toute autre information sémantique.

Depuis la version 3, le GML est maintenant adapté à la modélisation 3D et permet de représenter entre autres, des objets volumiques. Il permet aussi de représenter les objets topologiques. Un des points forts du langage GML est sa concordance avec le schéma spatial ISO 19107. En effet, on retrouve dans le GML les éléments de syntaxe nécessaires à la représentation de la majorité des entités géométriques et leur comportement définis dans le schéma spatial de l'ISO. Pour plus de détail sur la représentation des objets volumiques en format GML, consultez l'annexe B.

2.5.4 Les spécifications de services Web standards de l'OGC

L'OGC propose de standardiser les interfaces de services Web destinées au partage et à la diffusion de l'information géographique dans l'optique de favoriser l'interopérabilité entre les systèmes. En effet, la production de données géométriques à jour et de qualité est très coûteuse. L'échange et le partage de données entre organisations sont souvent les seules façons pour les décideurs et autres utilisateurs de données d'obtenir toute l'information désirée (de Vries *et al.*, 2004). Les possibilités d'Internet et des services Web côté diffusion de l'information font de ces technologies des avenues de premier plan pour les organisations qui désirent systématiser les processus de transfert de l'information

géographique. Par contre, la nature hétérogène des données spatiales autant sur le plan technique que sémantique cause problème. En effet, les nombreux formats de données utilisés pour stocker l'information ainsi que les différentes structures géométriques utilisées rendent très complexes les processus d'échange et d'intégration de données. L'interopérabilité entre les données et les systèmes doit être envisagée afin d'amenuiser cette hétérogénéité qui freine le partage de l'information. C'est donc pour ces raisons qu'il y avait nécessité du côté de l'OGC à standardiser ce processus de partage de l'information géographique. Ce travail de standardisation de l'OGC a résulté en différentes spécifications de services web standards telles que le *Web Map Service (WMS)*, le *Web Feature Service (WFS)*, le *Web Coverage Service (WCS)*, le *Web Terrain Server (WTS)* ainsi que le *Web 3D Service (W3DS)*. À noter aussi que l'OGC a récemment adopté comme standard pour la représentation et le partage de données graphiques à contenu géographique en ligne, le format de données propriétaire KML de Google qui permet la représentation sous format XML de modèles 3D destinés au visualisateur Google Earth. Dans les sections suivantes, nous présenterons ces spécifications en prenant soin d'identifier leurs capacités à gérer le 3D.

Web Map Service (WMS)

La spécification WMS est un standard international qui dicte le comportement d'un service Web permettant la création dynamique de cartes géographiques livrées sous format graphique telle qu'une image (WMS, 2001). Les cartes générées par le service sont créées à partir de données spatiales stockées sur un serveur. Suite à une requête d'un client contenant les détails de l'information désirée sous forme de paramètres (format de la carte, symbologie, coordonnées du secteur désiré...), la carte est générée et renvoyée vers le client pour consultation. Cette spécification a été la première à être développée dans la série de standards de services Web de l'OGC et définit trois opérations standards soit GetCapabilities, GetMap et GetFeatureInfo.

Le WMS est sans contredit une spécification limitée à la diffusion de modèles géométriques 2D. En effet, les paramètres passés au service lors des requêtes contiennent uniquement l'information requise pour la création d'une carte et seraient insuffisants pour

permettre la création d'une image d'un modèle 3D. Dans ce cas-ci, l'OGC a mis sur pied la norme WTS spécifiquement pour la diffusion d'images de modèles 3D.

Web Terrain Service (WTS)

La spécification WTS définit une interface standard pour requêter des scènes de terrains 3D générées à partir de données sur un serveur (OGC, 2003). Un WTS fonctionne d'une manière très semblable au WMS à l'exception qu'il sert des images de scènes 3D. Une scène 3D est une projection 2D d'un modèle 3D (ou 2.5D) dans une vue plane. Un WTS doit supporter les opérations *GetCapabilities* et *GetView*. La spécification WTS peut être vue comme un complément au WMS et est utile lorsqu'on veut obtenir des perspectives 3D de certains lieux ou objets.

Web Feature Service (WFS)

La spécification WFS (OGC, 2005b) est la seconde interface Web proposée par l'OGC qui permet de distribuer la donnée spatiale à partir d'un service Web. Contrairement au WMS, un WFS ne sert pas seulement une image de la donnée, mais bien la donnée elle-même en format standard. L'objectif du WFS est donc de recevoir les requêtes du client, de récupérer les données demandées sur le serveur et de retourner au client cette donnée en format XML encodée selon la norme GML. À l'inverse, un WFS peut aussi permettre de mettre à jour des données dans une base de données à partir de requêtes du client. L'interface du WFS propose minimalement les opérations suivantes *GetCapabilities*, *GetFeature* et *DescribeFeatureType*. La spécification WFS exige maintenant du GML version 3.x comme format de transfert de données. Bien qu'il n'ait jamais été présenté officiellement comme un service Web de distribution de données spatiales 3D, le WFS pourrait théoriquement servir des données 3D puisqu'il repose sur le GML 3, un format supportant la représentation de modèle 3D.

Web Coverage Service (WCS)

La spécification WCS définit une interface standard pour l'échange de données géographique sous forme de couverture i.e d'information numérique représentant les caractéristiques spatio-temporelles ou même multidimensionnelles d'un phénomène (OGC, 2006). Contrairement au WMS qui renvoie des cartes statiques sous forme d'image, un service Web WCS permet de distribuer la donnée avec sa sémantique associée comme par

exemple l'information contenue dans chaque pixel d'une image raster. Cette donnée peut être interprétée ou extrapolée contrairement à une image qui n'offre qu'un rendu graphique sans information sous-jacente. D'une manière générale, on peut décrire le WCS comme l'équivalent du WFS mais avec la capacité de distribuer des données en tessellation au lieu de données discrètes. Un service WCS supporte les opérations `GetCapabilities`, `DescribeCoverage` et `GetCoverage`. Supportant les couvertures multidimensionnelles, la spécification WCS pourrait théoriquement permettre la distribution de modèles 3D à base de voxels.

Web 3D Service (W3DS)

La spécification W3DS, toujours à l'étape de « Draft », définit une interface standard, permettant de servir des modèles 3D sous la forme de scènes 3D destinées à être consultées et manipulées de manière interactive dans un navigateur Web par le biais d'un plugin ou dans un visualisateur de globe virtuel. Un des principaux aspects du W3DS est la diffusion de ces scènes 3D accompagnées de l'information graphique permettant non pas de télécharger les données brutes comme dans le cas d'un WFS mais bien une représentation visuelle de celles-ci incluant par exemple couleurs, textures, animations et éclairage (Schilling et Kolbe, 2010). Contrairement au WMS, ces scènes ne sont pas servies sous forme d'images mais sous des formats standards tels que le X3D ou le KML, formats d'échanges centrés sur la visualisation graphique. Il est important de mentionner que le W3DS ne vient pas remplacer le WFS mais bien offrir un service complémentaire à celui-ci. En effet, ce standard prévoit que ces scènes 3D pourraient très bien être créées à partir de données brute obtenues d'un WFS, lesquelles seraient formatées et augmentées de l'information graphique nécessaire à leur visualisation et leur animation.

2.6 Inventaire des géotechnologies 3D

Maintenant que nous connaissons mieux les différentes facettes de la manipulation des données 3D et des standards s'y rapportant, il est pertinent de réaliser un inventaire plus approfondie des technologies 3D en liens avec ces aspects.

Contrairement au domaine du 2D où l'on retrouve une grande diversité de solutions ou technologies faisant intervenir des données spatiales, le domaine du 3D reste relativement

peu exploité vis-à-vis certains aspects tels la gestion, l'analyse ou la diffusion des modèles. Plusieurs projets de recherche se sont attardés et s'attardent encore à faire avancer le domaine du 3D en ce sens (Balovnev *et al.*, 1997; Breunig, 1999 et 2001; Shumilov et Breunig, 2000 ; Zlatanova, 2000; Stoter et Zlatanova, 2003; Arens *et al.*, 2003; Shi, 2003; Apel, 2004; Lachance, 2005; Bédard, 2006; Pouliot *et al.*, 2008; Ledoux et Gold, 2008; Kolbe, 2009). Afin de présenter ce bilan, nous distinguerons ces géotechnologies 3D par leur finalité et les regrouperons en différentes catégories soit les outils CAO, DAO, les SIG, les visualisateurs de globes virtuels et les SGBD spatiaux. Pour chacune de ces catégories, nous dresserons un portrait de ses aptitudes vis-à-vis du 3D en faisant un bref rappel et en venant compléter l'information déjà présentée au chapitre 1 le cas échéant. Un bref aperçu de quelques-unes des géotechnologies analysées les plus populaires est aussi présenté en annexe D. Les aptitudes 3D des différentes catégories de géotechnologie sont évaluées et présentées sous la forme d'un tableau comparatif. Cette comparaison a été effectuée sur la base de l'évaluation de trois solutions jugées représentatives de chaque catégorie de géotechnologie soit celles présentées en annexe D. Chaque catégorie de géotechnologie est analysée sur la base des critères modélisation géométrique 3D, analyse spatiale 3D, gestion des données 3D (usage de bases de données), partage des données 3D (sur le Web) et usage des standards en 3D. Ces critères d'évaluation ont été choisis puisqu'ils permettent à notre avis de faire ressortir les caractéristiques fondamentales des produits analysés. Pour ce qui est du critère de la modélisation géométrique 3D, nous ferons la distinction entre visualisation 3D, modélisation volumique et modélisation 2.5D.

Cet inventaire des géotechnologies a été réalisé en 2008 et repose sur diverses expériences personnelles (professionnelles et pédagogiques) réalisées sur plusieurs des logiciels présentés et sur différentes lectures complémentaires (documentation technique, aide en ligne, sites Web officiels, blogues, etc.). Dans le cas particulier des outils DAO, des entrevues ont aussi été réalisées avec des professionnels du milieu de l'infographie et des jeux vidéo.

Le tableau 3 présente le résumé comparatif des différentes catégories de géotechnologie. Les sections suivantes présentes un bref aperçu de chacune des catégories et de leurs

aptitudes face au 3D. Pour plus de détail sur la justification de chacun des choix du tableau comparatif, se référer à l'annexe E.

	Capacité en termes de				
	CAO (Autocad, Gocad, Micro- Station)	DAO (3dsMax, Maya, Blender)	SIG (ArcGIS, MapInfo, GeoMedia)	Visualisateur de globe virtuel (Google Earth, Virtual Earth, ArcGIS Explorer)	SGBD Spatiaux (Oracle, SQL Server, PostGIS)
Visualisation 3D	Possible	Possible	Possible	Possible	Non disponible
Modélisation de volume	Possible	Possible	Non disponible	Non disponible	Non disponible
Modélisation 2.5D	Possible	Possible	Possible	Possible	Non disponible
Analyse spatiale 3D	Partiellement possible	Non disponible	Non disponible	Non disponible	Non disponible
Stockage base de données 3D	Non disponible	Non disponible	Partiellement possible	Non disponible	Partiellement possible
Partage données 3D sur le Web	Non disponible	Partiellement possible	Partiellement possible	Possible	Non disponible
Usage des Standards en 3D	Non disponible	Non disponible	Possible	Partiellement possible	Partiellement possible

Tableau 3: Résumé comparatif des capacités des géotechnologies en termes de 3D.

2.6.1 Les outils CAO

Les logiciels de conception assistée par ordinateur (CAO) sont des outils de modélisation qui permettent la conception, l'édition et la visualisation de modèles géométriques 3D. Ces outils ont été conçus à la base pour des besoins spécifiques en architecture, ingénierie mécanique, et construction, solutions dites AEC-Architecture/Engineering/Construction (Kunwoo, 1999 ; Lattuada *et al.*, 2004). De nos jours, les logiciels CAO sont aussi utilisés dans plusieurs secteurs tels la modélisation géologique (Bédard, 2006), archéologique (Losier *et al.*, 2007), etc.

Ces logiciels permettent l'exploitation de diverses structures de données géométriques 3D comme le B-Rep, le CSG, la décomposition en cellule, le Voxel, le balayage, etc... et font aussi appel aux courbes de Bézier ou aux NURBS. Tel que mentionné au chapitre 1, ils exploitent peu les bases de données et ne permettent habituellement pas d'effectuer d'analyse topologique (métrique seulement). Malgré de récents efforts de dialogue entre certains acteurs de l'industrie de l'AEC, celle-ci souffre encore d'un manque d'interopérabilité notamment au niveau des données et les outils CAO sont souvent pointés du doigt (Gallaher *et al.*, 2004).

2.6.2 Les DAO

Les logiciels de dessin assisté par ordinateur (DAO) sont des outils de modélisation 3D, généralement utilisés par des infographistes, plutôt orientés vers le dessin, la construction et la visualisation de modèles 3D destinés à l'animation cinématographique, aux jeux vidéo et à la création d'environnements de réalité virtuelle. Tel que mentionné au chapitre 1, les outils DAO implique généralement l'instanciation de forme de base qui sont ensuite modelées manuellement par le modélisateur. On exploite principalement les structures CSG, les B-Rep, et surtout les formes paramétrables. Le stockage des modèles des outils DAO inventoriés se limitaient aux fichiers et l'analyse spatiale à certains opérateurs métriques comme le calcul de distance. Pour ce qui est de la diffusion, les modèles créés au sein d'un DAO sont rarement destinés à être montrés ou publiés dans leur état brut. Ils sont plutôt texturés, colorés et éclairés pour ainsi pouvoir être publiés de manière plus artistique. Ils sont aussi souvent intégrés dans un environnement plus complexe comme un jeu vidéo ou animés au sein même du DAO ou par le biais d'un produit externe pour produire une séquence vidéo. Lors de notre inventaire, nous n'avons pas trouvé d'outil DAO implémentant les standards de l'ISO et de l'OGC.

2.6.3 Les SIG

Tel que mentionné, au chapitre 1, les systèmes d'information géographique (SIG) sont des outils bien adaptés pour l'acquisition, la structuration et la manipulation de données géométriques. Ils permettent l'analyse spatiale autant métrique que topologique et permettent l'usage des bases de données pour le stockage des modèles. Ils gèrent aussi les systèmes de référence cartographique. La plupart des principaux fournisseurs de SIG

offrent un module de diffusion venant compléter leur produit phare en permettant de publier et de consulter dynamiquement les cartes construites à l'aide de leur SIG sur le Web. Toutefois, notre inventaire nous a permis de constater que les SIG sont des systèmes qui sont spécialisés dans la gestion de données spatiales 2D. Ils peuvent lire certains formats 3D mais se limitent à la création de MNT et à l'extrusion de primitives géométriques 2D à l'aide d'une hauteur. Au niveau de la standardisation, la plupart des SIG implémentent les standards et spécifications de l'ISO et de l'OGC (SFS, GML, WMS, WFS, ...) soit directement dans leur produit de base ou par le biais d'extensions dites « d'interopérabilité ».

2.6.4 Les visualisateurs de globes virtuels

D'autres solutions de visualisation cartographique possédant certaines capacités pour la diffusion Web de données 3D, les visualisateurs de globe virtuel, ont aussi fait leur apparition plus récemment. Nous parlons ici des applications Google Earth, Virtual Earth, Nasa World Wind et ArcGIS Explorer. La grande popularité de ces applications et leur distinction des SIG fait en sorte que nous pensons qu'il vaille la peine d'en faire une catégorie à part entière et de les présenter un peu plus en détail.

Bien que l'on diffuse déjà de l'information spatiale sur le Web depuis plusieurs années, ces nouvelles applications innovent par leur mode de navigation saisissant, leur vitesse, leur gratuité et par l'énorme quantité d'information qu'ils servent en ligne et ce, dans un univers 3D. En effet, ces produits offrent la possibilité de naviguer sur une terre virtuelle en 3D et y visualiser différentes couches vectorielles (réseaux routiers, commerces, lieux touristiques, etc.), une couverture d'imageries satellitaires drapée sur le relief (TIN) ainsi que des bâtiments en 3D (extrusion par balayage). Côté analyse spatiale, ces applications se limitent en général à des calculs de distance et de corridors (analyse métrique). Le mode de stockage des modèles offert aux usagers se limite aussi aux fichiers et ces solutions ne supportent généralement pas le standard GML. Il est possible de superposer sur le globe des images provenant d'un service WMS.

2.6.5 Les SGBD spatiaux

Pour ce qui est de la gestion des modèles géométriques, nous avons vu plus tôt dans ce chapitre que les systèmes de gestion de bases de données (SGBD) sont des outils qui offrent généralement la possibilité de stocker et de requêter des objets géométriques sous forme de points, de lignes, de polygones et exceptionnellement de solides (Oracle Spatial depuis sa version 11g) ou sous forme matricielle au sein de bases de données. Bien qu'ils permettent le stockage des coordonnées en 3D (x,y,z), les SGBD spatiaux implémentent le standard SFS qui est à la base un schéma spatial vectoriel destiné à la modélisation 2D. Les opérateurs spatiaux qu'ils fournissent vont aussi en ce sens et ne tiennent pas compte de la troisième dimension lors des calculs.

2.6.6 Recensement des projets de recherche sur le SIG 3D

Maintenant que nous avons vu plus en détail les principales géotechnologies 3D et leurs particularités liées à la 3D, il est pertinent de recenser les principaux projets de recherche qui touchent à cette même thématique. En 1999, Breunig présente GeoStore/Geotoolkit (Balovnev *et al.*, 1997; Breunig, 1999 et 2001), une application développée pour permettre un accès distant, la visualisation 2D/3D et certains traitements appliqués à des modèles géologiques 3D stockés dans une base de données spatiale 3D, à partir de différents outils client. Pour le stockage des modèles 3D, Geotoolkit repose sur le SGBD orienté-objets ObjectStore dans lequel sont implémentées les primitives spatiales suivant une structure de *simplicial complexes* i.e. les surfaces sont construites à l'aide de *mesh* de triangles et les solides sont construits par un assemblage de tétraèdres. Bien qu'extensible et ouvert, le GeoToolkit se limite à l'utilisation d'un seul SGBD. Dans Shumilov et Breunig (2000), on décrit comment Gocad fut connecté à GeoToolKit et ObjectStore dans un effort visant l'intégration de données géologiques et géophysiques 3D dans une base de données commune à l'aide d'un adaptateur Corba. Ce projet représente un bel effort de couplage entre un outil CAO et une base de données. Mentionnons que l'équipe de Breunig travaille depuis 2007 sur le projet DB4GeO, une version plus moderne du GeoStore/Geotoolkit prenant maintenant en charge le 4D et qui utilise db4o au lieu de ObjectStore pour le stockage des modèles. Cette application permet l'interrogation des modèles à travers le Web et l'exportation des données vers les formats Gocad, GML et VRML (Breunig *et al.*, 2009).

En 2003, Stoter et Zlatanova expliquent comment il est possible de stocker des solides dans un SGBD spatial en les stockant comme des multi-polygones (ensembles de polygones disjoints ou se touchant uniquement en un nombre fini de points et ne s'intersectant pas qui sont traités comme une seule entité) ou comme un ensemble de polygones. Bien que cette façon de faire permette effectivement de profiter de plusieurs avantages qu'offrent les SGBD, elle comporte plusieurs problèmes comme on le mentionne dans Arens *et al.*, 2003 :

- Le SGBD spatial ne reconnaît pas les objets spatiaux 3D parce qu'il n'a pas de primitive 3D pour les modéliser. Cela résulte en un mauvais fonctionnement de certaines fonctions spatiales du SGBD
- Il n'y a pas de validation pour les objets 3D en temps qu'élément à part entière (seulement en temps qu'ensemble de polygones)
- Les mêmes coordonnées sont stockées plusieurs fois causant un risque d'inconsistance des données
- Les opérateurs spatiaux du SGBD ne fonctionnent que lorsque les objets sont projetés en 2D parce que la 3^e dimension est ignorée
- Il n'y pas d'information permettant de distinguer l'intérieur et l'extérieur des objets
- Lorsque les objets 3D sont représentés et stockés comme un ensemble de polygones, il y a une relation 1:n (un à plusieurs) entre l'objet et le nombre d'enregistrements de la table. Pour gérer de façon plus claire et efficiente un large jeu de données, une relation 1:1 est nécessaire entre les objets de la base de données et la réalité

Afin de régler une partie de ces problèmes, Arens *et al.*, 2003 proposent d'implémenter une primitive spatiale volumique dans Oracle spatial 9i. Ils expliquent comment stocker, valider, indexer et requêter un objet spatial 3D dans ce SGBD et ce, par l'implantation d'une primitive spatiale 3D suivant la structure géométrique du B-Rep. Ces travaux présentent sans doute la première démonstration d'un SGBD pouvant supporter directement une primitive 3D. Toutefois, les problèmes au niveau des opérateurs spatiaux persistent toujours puisque ceux-ci ne reconnaissent pas la nouvelle primitive et la 3^e dimension des coordonnées est donc toujours ignorée.

En 2004, Apel présente un projet de développement d'un SIG 3D orienté vers le Web et suivant une architecture distribuée (client-serveur-SGBD). Le système qu'il propose consiste en le couplage d'un outil de modélisation 3D (Gocad), d'une base de données XML et d'un serveur d'applications servant d'intermédiaire entre la base de données et les clients. L'objectif de ce projet est de favoriser l'interopérabilité des données par le partage et le stockage des données en XML ce qui lui permet d'être indépendant des langages et des systèmes d'exploitation. Apel propose de stocker directement les données en XML dans un SGBD spécialisé dans le stockage XML natif (TAMINO) et de retirer de la base de données, les tâches reliées à l'analyse spatiale. Pour ce faire, il propose un nouveau serveur d'application spécialement conçue pour les calculs d'analyse spatiale 3D. Bien que très intéressant au niveau de son architecture, le requêtage et le stockage des données présentent certaines lacunes puisqu'aucune structure de données ni spécification de service Web standard n'ont été utilisées. Le projet est aussi resté au niveau du prototype, la gestion des solides et des opérateurs spatiaux 3D n'étant implémentée que partiellement.

En 2006, un autre groupe de chercheurs du secteur des sciences de la Terre au Ministère des Ressources Naturelles du Canada et de Mira Geoscience ont aussi travaillé à étendre les capacités de Gocad (Sprague *et al.*, 2006). Un système de requêtage appliqué à l'exploration minière est proposé supportant par exemple des requêtes de proximité, d'intersection et aussi des requêtes portant sur les propriétés descriptives. Bien qu'intéressant pour les acteurs du domaine minier, ce plug-in Gocad ne permet que le requêtage de propriétés descriptives de type numérique et cette information reste gérée au niveau de fichiers n'utilisant pas de SGBD pour le stockage. Son utilisation est aussi restreinte aux modèles de données de Gocad.

En 2006, dans un effort de requêtage topologique d'objets 3D issus d'une modélisation par un outil CAO, Lachance *et al.* présentaient le prototype TQuery (Lachance, 2005; Pouliot *et al.*, 2008). TQuery pour 3D Topological Query, permet l'exportation d'objets Gocad vers une structure de stockage 3D topologique implémentée dans un SGBD (SQL Server) qui régularise certaines relations de voisinage de type touche et intersecte, entre les objets volumiques (représentés par un assemblage de tétraèdres). Bien qu'intéressant, ce prototype présente certaines lacunes. Par exemple, plusieurs manipulations de nettoyage du modèle

3D sont nécessaires avant de pouvoir être capable de requêter celui-ci et il n'existe pas de lien réel entre Gocad et la base de données puisque l'import/export se fait manuellement. L'interface limite aussi l'analyse topologique à seulement deux objets à la fois.

Plus récemment, Kolbe et son équipe travaillent au développement de modèles 3D urbains interopérables avec le City Geography Markup Language (CityGML) (Gröger *et al.*, 2008; Kolbe, 2009; Stadler *et al.*, 2007; Stadler *et al.*, 2009). CityGML est un standard de l'OGC basé sur GML 3 qui définit les classes et les relations pour les objets spatiaux que l'on retrouve en milieu urbain en prenant en charge leurs propriétés géométriques, topologiques, sémantiques et graphiques.

À noter aussi les efforts de Lapierre *et al.*, 2007 et Hagedorn et Dollner, 2007 qui ont démontrés la faisabilité d'intégrer des données provenant de CAD, SIG et BIM (Building Information Model) dans une approche de services Web standard. Ou encore, ceux de Chang et Li, 2008 qui présente l'architecture d'une application SIG 3D collaborative synchrone destinée à la visualisation et au partage de modèles 3D dans un contexte de prise de décision. Sans aller dans les détails, d'autres projets se sont aussi penchés sur le thème du SIG 3D et ont produit des prototypes comme celui de Zlatanova en 2000 ou le SpaceInfo de Shi, 2003.

2.7 Conclusion du chapitre

Nous avons vu que les SIG actuels couplés aux SGBD spatiaux ont atteint un niveau de maturité important au niveau de la modélisation, de la gestion, de l'analyse et du partage des données géométriques 2D et 2.5D. Toutefois, les SIG sont des systèmes qui permettent la visualisation de modèles 3D mais qui reste très limités quant à leur construction ou édition. Les SGBD spatiaux, pour leur part, ne sont pas conçus pour gérer des objets volumiques (à l'exception de Oracle 11g tout récemment). De leur côté, les outils CAO se sont distingués comme de robustes outils de modélisation 3D permettant la gestion explicite des objets volumiques. Toutefois, aucun de ces systèmes n'est encore apte à réaliser au niveau 3D, et ce, de manière intégrée, la construction, la gestion, l'analyse et la diffusion des modèles géométriques. D'ailleurs, il n'existe pas encore de bibliothèque offrant les algorithmes permettant d'implémenter les opérateurs spatiaux capables de

calculer la proximité, l'intersection, l'inclusion, l'égalité ou l'adjacence d'objets géométriques 3D provenant de modèles basés sur des surfaces ou des solides. Plusieurs projets de recherche ont amené de bonnes idées et ont proposé des solutions pour contourner les limites de ces systèmes (Balovnev *et al.*, 1997; Breunig, 1999 et 2001; Shumilov et Breunig, 2000; Zlatanova, 2000; Stoter et Zlatanova, 2003; Arens *et al.*, 2003; Shi, 2003; Apel, 2004; Sprague *et al.*, 2006). Cependant, et comme nous l'avons mis en évidence, aucune de ces solutions n'a été jugée adéquate pour répondre complètement et correctement à ces tâches. Entre autres, les solutions de type SIG 3D :

- ne reposent pas sur des standards au niveau de la structure géométrique pour la gestion des objets 3D.
- n'utilisent pas de spécifications de service Web standards pour l'interrogation de données sur le Web.
- n'utilisent pas de standard de représentation de données pour le partage de l'information géométrique sur le Web.

Nous estimons pourtant que la communauté géomatique a besoin de ce genre de solution 3D et c'est dans cette perspective que nous allons faire une proposition au chapitre suivant.

Chapitre 3. Proposition d'une architecture de SIG 3D orientée service Web

3.1 Les composantes fondamentales d'un SIG 3D

Nous avons discuté au chapitre précédant de différents concepts et technologies touchant l'information géographique en lien avec le 3D. En se basant sur cette information, nous sommes maintenant en mesure de résumer et mettre en évidence les aspects importants à tenir en compte dans la définition de ce que devrait être, selon nous, un SIG 3D. Dans cette réflexion, il nous est apparu important de garder en mémoire les capacités actuelles des SIG 2D afin de s'en servir comme ligne directrice dans l'établissement du cahier des charges du SIG 3D. Rappelons qu'un SIG est défini comme un système assisté par ordinateur pour l'acquisition, le stockage, la récupération, l'analyse et la visualisation des données spatiales (Clarke, 1986). Les SIG gèrent aussi les systèmes de référence cartographique et permettent d'associer des propriétés descriptives aux objets géométriques des modèles. Cette dernière caractéristique est très importante puisque c'est à partir de cette sémantique associée à chaque objet que peut être élaborée la cartographie thématique. On pourrait aussi spécifier que l'analyse spatiale offerte par les SIG est de type métrique et topologique. Ils permettent aussi généralement la gestion de plus d'une structure géométrique. En effet, la plupart des SIG gèrent par exemple une structure topologique vectorielle, une structure matricielle à base d'images rasters et une structure 2.5D comme les modèles numériques de terrain (MNT) avec une technique de maillage comme le TIN. Un SIG 3D devrait évidemment offrir ces mêmes fonctionnalités tout en les adaptant à la donnée volumique. Un SIG 3D offrirait donc les outils permettant de répondre aux questions suivantes :

- Où (coordonnées x,y,z) est situé l'objet et son étendue spatiale (celle de l'objet et du modèle 3D qui l'englobe)
- De quel type d'objet 3D s'agit-il (en référence à ses propriétés descriptives)
- Comment sont répartis les objets 3D et leurs relations spatiales dans l'espace tridimensionnel (opérateurs métriques et topologiques 3D)

D'une manière plus spécifique, un SIG 3D devrait être capable de construire, manipuler, gérer et présenter des primitives géométriques de type point, ligne, polygone et volume à

l'aide de l'une ou plusieurs des structures géométriques 3D identifiées plus tôt. Il n'est pas nécessaire qu'il les maîtrise tous, mais comme le SIG 2D, il serait intéressant qu'il en offre plus d'une afin d'être polyvalent et de permettre d'opter pour la structure la plus appropriée aux besoins du moment. Par exemple, il serait intéressant qu'il puisse offrir une structure géométrique basée sur des surfaces comme le B-Rep pour sa simplicité et sa grande flexibilité au niveau de la représentation des modèles. Il serait aussi important de pouvoir remplir ces objets surfaciques et donc de maîtriser l'une des structures de type ensemble de solides comme la modélisation tétraédrique. Puisque certains types d'objets ou de phénomènes sont mieux représentés à l'aide de structures orientées-espace, la maîtrise d'une structure à base de voxels pourrait aussi être envisagée. Un SIG 3D devrait également permettre de stocker les modèles au sein de bases de données. Il devrait offrir des opérateurs spatiaux métriques et topologiques capables de calculer des distances, des surfaces et des volumes, mais aussi la proximité, l'intersection, l'inclusion, l'égalité et l'adjacence entre objets spatiaux en tenant compte de la 3^e dimension. Un SIG 3D devrait aussi idéalement pouvoir s'adapter et interagir facilement avec d'autres systèmes, i.e. être interopérable. Pour ce faire, il devrait implémenter des interfaces standards et adopter les différents standards internationaux offerts par les organisations de standardisation telles que le W3C, l'ISO et l'OGC. Ces organisations présentent des standards et spécifications au niveau de la structure géométrique et de la représentation des données spatiales (ISO 19107, XML, GML), mais aussi au niveau de leur diffusion sur le Web par le biais de services Web (WMS, WTS, WCS, WFS). Cette diffusion des modèles est aussi selon nous un élément important que devrait maîtriser un SIG 3D.

3.2 Stratégies de développement d'une solution SIG 3D

En partant des composantes fondamentales du SIG 3D précédemment énoncées, nous allons maintenant voir dans quelle mesure il est possible et quels sont les efforts requis pour effectuer un tel développement.

Un SIG 3D devrait permettre de construire, de visualiser, de valider et de manipuler des modèles 3D. Pour ce faire, le système devra reposer sur une structure géométrique 3D permettant de gérer de manière explicite des objets volumiques. On devra s'assurer que les objets géométriques créés au sein de cette structure soient valides en respectant des règles

de validations préétablies en fonction de la structure choisie. La visualisation et manipulation des modèles impliquent de devoir offrir une interface graphique à l'utilisateur permettant d'attribuer des variables graphiques telles qu'une couleur, une épaisseur ou une texture aux primitives géométriques et offrir des outils de manipulation afin de faire bouger les modèles avec des rotations, des translations et des changements d'échelles. Heureusement, nous avons vu que ces dernières fonctionnalités sont bien connues et existent déjà au sein de plusieurs solutions comme les outils CAO. Le système devra aussi pouvoir compter sur des opérateurs spatiaux métriques et topologiques capables de calculer des distances, des surfaces et des volumes, mais aussi la proximité l'intersection, l'inclusion, l'égalité et l'adjacence entre objets spatiaux en tenant compte de la 3^e dimension. À cet effet, nous proposons de s'en tenir à la liste des opérateurs spatiaux proposée par l'OGC comme référence au niveau des opérateurs topologiques puisqu'elle se veut suffisamment complète pour permettre de combler la majorité des besoins. Toutefois, ceux-ci devront faire appel à une algorithmie 3D i.e. qui tient compte des coordonnées x,y,z lors des traitements d'objets 3D ponctuels, linéaires, surfaciques et volumiques. Alors que l'implémentation des algorithmes d'analyse spatiale métrique 3D ne devrait pas trop poser de problèmes puisqu'étant relativement simple et bien documentée, le cas des algorithmes topologiques pourrait s'avérer plus complexe. En effet, du côté 2D, il existe des bibliothèques d'algorithmes d'opérateurs spatiaux topologiques comme par exemple la Java Topology Suite (JTS). Toutefois, ce genre de librairie n'existe pas encore pour le 3D (ou sont incomplète et non standard) et très peu d'applications offrent ce genre d'opérateur. Le système proposé devra aussi, comme le permettent les SIG 2D, stocker les modèles au sein de base de données afin de permettre la gestion intégrée des modèles 3D et de leurs propriétés descriptives. L'usage de bases de données permettra, contrairement au stockage par fichier, de gérer l'information spatiale de manière institutionnelle. Pour y arriver, une stratégie permettant de stocker des objets 3D au sein d'un SGBD devra être adoptée. Pour ce faire, on peut opter pour un SGBD comme Oracle Spatial 11g qui offre une primitive volumique pour le stockage des objets 3D ou implémenter sa propre structure de stockage 3D de préférence au sein d'un SGBD existant afin de ne pas avoir à réinventer la roue. Pour ce qui est du partage des modèles 3D, il devrait pouvoir être réalisé à partir d'internet puisqu'étant de loin le meilleur media de communication existant, et ce, de manière

standard pour des raisons d'interopérabilité. Les spécifications de services Web de l'OGC ainsi que le GML devraient aussi être adoptés pour aller en ce sens. Plusieurs stratégies ont été examinées pour la conception de notre SIG 3D soit :

- Créer un tout nouveau système à partir de zéro
- Créer un nouveau système en tirant profit de bibliothèques de développement offrant un soutien 3D
- Extensionner un SIG existant
- Extensionner un outil CAO existant

La première stratégie consiste à développer entièrement un nouveau produit à partir de zéro, sans récupérer de composantes existantes. Bien que cette stratégie offre beaucoup de flexibilité et permettrait d'orienter sans contraintes notre développement vers les objectifs précis que nous nous sommes fixés, elle s'avère la plus coûteuse en terme de temps et de ressource et est aussi la plus complexe puisque tout est à développer. Très peu de projets de développement informatique aujourd'hui, le moins complexes, redéveloppent entièrement sans faire usage de composantes ou de bibliothèques déjà existantes.

Les trois autres stratégies examinées impliquent la réutilisation de ressources existantes afin d'éviter de devoir redévelopper entièrement le système et de profiter de la programmation existante déjà validée, optimisée et blindée. Dans cette optique, la seconde stratégie serait de créer un nouveau système en tirant profit de bibliothèques et d'APIs (Application Programming Interface) 3D existantes telles que la Computational Geometry Algorithms Library (CGAL), Open Cascade ou OpenGL. Ces bibliothèques offrent un ensemble de classes, de méthodes et d'algorithmes de base reliés à la représentation et à la manipulation d'objets et de structures géométriques sur lesquels on peut bâtir une application fiable et ainsi pouvoir éviter de devoir en redévelopper les fondations qui sont souvent communes à un grand nombre d'applications. Ces bibliothèques et plusieurs autres ont été utilisées comme base de développement dans un grand nombre d'applications existantes. L'utilisation d'une telle bibliothèque permet de sauver du temps au niveau du développement tout en s'assurant de faire reposer son application sur une base solide ayant déjà fait ses preuves. Toutefois, ces bibliothèques peuvent être qualifiées de « bas niveau »

ce qui signifie qu'elles fournissent les outils de base ou la fondation d'une application. Beaucoup de travail reste à faire avant d'obtenir une application complète.

Afin d'aller vers une solution un peu plus complète, on peut opter pour la réutilisation d'un SIG ou d'un outil CAO existant. En effet, comme un SIG offre déjà les fonctionnalités que nous cherchons à implémenter, le travail consisterait donc à étendre ses capacités afin de lui permettre de supporter le 3D. Pour ce faire, on doit minimalement ajouter au système la gestion d'une primitive volumique puisque c'est précisément ce qui manque au SIG pour leur permettre de gérer le 3D. Le problème avec cette stratégie c'est qu'elle demande de modifier le fondement même du programme i.e. le schéma spatial décrivant les primitives géométriques. Arens *et al.*, 2003 avaient proposé une telle stratégie dans le cas spécifique de la base de données Oracle. Cette stratégie à long terme est toutefois intéressante car elle profite de la mise en marché du produit lui-même (si ce produit possède déjà une bonne part du marché, alors la viabilité de la solution sera probablement élevée). Cependant, cette stratégie exige une connaissance complète de cette structure, elle peut être complexe à programmer et surtout cette solution ne serait valable que pour la technologie choisie. De plus, le fait d'ajouter la gestion d'une nouvelle primitive volumique au cœur du système ne va pas du même coup permettre le stockage, l'analyse et la diffusion de celle-ci. Au contraire, chacune de ces composantes devra à son tour être adaptée pour tenir compte des modifications apportées à la structure de base du programme. Ces développements exigeraient donc de nombreux efforts et une très bonne connaissance tant au niveau du système lui-même que de l'algorithmie géométrique associée.

Au lieu d'opter pour un SIG, on peut aussi étendre les capacités d'un outil CAO. L'avantage de cette stratégie est que les outils CAO possèdent déjà de bonnes aptitudes pour la modélisation géométrique 3D et possèdent, entre autres, une structure géométrique permettant la représentation de volume. Ainsi, sans devoir modifier la structure de base de l'application, les améliorations à effectuer se situeraient davantage au niveau de l'ajout de fonctionnalités liées à l'analyse spatiale et à l'interrogation des données descriptives. Par exemple, des opérateurs spatiaux topologiques 3D devraient être implémentés afin de donner au système les capacités d'analyse spatiale désirées. L'intégration à un SGBD pour la gestion des modèles volumiques serait aussi nécessaire. Une stratégie de partage des

modèles devrait aussi être envisagée afin de permettre d'accéder aux modèles volumiques à partir du Web.

3.3 Choix d'une stratégie de développement

Nous avons tout d'abord jugé que le développement complet d'un SIG n'était pas une solution optimale. Nous avons en effet estimé qu'il existe actuellement une large panoplie de géotechnologies (notre inventaire le démontrant) et que nous devons profiter des forces et avantages de ces produits. Nous avons ainsi jugé que les technologies SIG et CAO étaient suffisamment performantes et matures quant à la manipulation et la gestion de données spatiales, même si elles présentent chacune des lacunes quant à la 3D. Ainsi, en résumé, nous avons le choix entre nous attaquer directement au noyau d'un SIG afin de lui greffer une primitive volumique ou rajouter à un outil CAO certaines capacités de gestion, d'analyse et de diffusion des données.

En informatique, il est très complexe d'aller revoir les fondements mêmes d'une application lorsque les changements à apporter sont majeurs et n'ont pas été prévus lors de sa conception. On dit souvent d'une manière symbolique que dans le processus de développement d'une application, une modification apportée au moment de la conception coûte 1\$, durant le développement coûte 100\$ et durant l'implantation coûte 1 000\$. Nous pensons donc qu'extensionner un SIG existant représente un travail colossal étant donné la nature des changements à apporter.

Lorsqu'on parle d'intégration d'une application existante à un autre système ou d'ajout de fonctionnalités à celles déjà existantes en respectant les fondements de cette application, on se trouve dans une situation relativement courante et de complexité généralement moindre que de devoir s'attaquer au cœur d'un programme. Les systèmes d'aujourd'hui étant souvent pensés et construits de manière évolutive et modulaire, leur structure se prête généralement bien à une telle intégration ou à de tels ajouts de fonctionnalités lorsqu'ils ont été bien construits à la base. De plus, il était essentiel pour nous d'avoir un système qui implémente une ou plusieurs des structures géométriques 3D présentées au chapitre précédant. Pour ces raisons, nous avons décidé d'extensionner un outil CAO existant pour lui ajouter certaines fonctionnalités de gestion et de diffusion de données 3D. Il était alors

important de sélectionner un outil CAO spécifique dit "ouvert" i.e. qui possède par exemple un API permettant de venir ajouter des fonctionnalités au système de base. Il était aussi important de choisir un outil CAO qui implémente une ou plusieurs structures géométriques identifiées comme intéressantes pour la représentation du territoire en trois dimensions.

Comme les outils CAO ne permettent pas, de manière native, le stockage des modèles au sein de bases de données, nous proposons de coupler le système à un SGBD. Ainsi, les modèles 3D pourront être stockés au sein de bases de données et on pourra tirer profit de tous les avantages que procurent les SGBD par rapport au stockage par fichier (sécurité et intégrité des données, accès multi-usagers, etc.). Et puis, comme nous avons privilégié la diffusion et l'interopérabilité, Internet et l'utilisation de standards ont été identifiés comme composantes essentielles. L'interopérabilité sera considérée tant au niveau de la structure, de l'échange et du requêtage des données. Ainsi, nous proposons le développement d'un SIG 3D suivant une approche de service Web standard. L'adoption d'une telle approche, contrairement à l'approche traditionnelle client/serveur i.e. un serveur Web publiant une page Web, offre plusieurs avantages. Le service Web n'impose pas à l'utilisateur une interface graphique particulière, mais partage plutôt la logique applicative (le noyau fonctionnel de l'application), les données et les traitements par le biais d'une interface standard à travers le réseau local ou Internet. Par la suite, un développeur peut ajouter le service Web à une interface graphique comme un programme ou une page Web pour offrir des fonctionnalités spécifiques à un groupe d'utilisateur. De plus, l'approche service Web présentant une interface standard, la communication entre différentes applications par l'intermédiaire de ce service ne demande pas de trop grands efforts d'adaptation et de programmation spécifique non réutilisable de part et d'autre. Finalement, comme les communications exploitent le langage XML, elles ne sont pas limitées à un système d'exploitation ou un langage de programmation spécifique. Cela signifie par exemple qu'une application Java tournant sous Linux pourrait très bien communiquer avec une application en .Net sous plate-forme Windows. Dans le cadre de l'architecture de SIG 3D que nous proposons, un service Web sera ajouté au système proposé afin d'offrir un service de partage de données spatiales et d'opérateurs spatiaux 3D à partir d'un réseau local ou d'Internet.

En résumé, nous proposons un SIG 3D construit sur la base d'un outil de modélisation 3D de type CAO couplé à un SGBD pour le stockage des modèles dans une approche de service Web. Apel présentait en 2004 un prototype de SIG 3D mettant en œuvre un service Web pour la diffusion des modèles géologiques 3D. Son système permettait l'accès en ligne à des modèles 3D stockés sous format XML dans un SGBD spécialisé dans le stockage XML natif à partir du client Gocad. Cet effort de Apel apportait des idées intéressantes et innovantes et a beaucoup inspiré nos travaux. Toutefois, le requête et le stockage des données de son système présentait certaines lacunes puisqu'aucune structure de données géométriques, ni spécification de service Web standard n'ont été utilisées. C'est particulièrement à ce niveau, soit celui des normes et des standards, que l'architecture du système que nous proposons se distingue de ce celui de Apel. En effet, nous proposons dans l'architecture de notre système l'utilisation de spécifications et de standards :

- Au niveau du service Web par l'adoption de la spécification WFS de l'OGC
- Au niveau de la représentation et du transfert des données par l'adoption du format GML
- Au niveau de la structure des données géométriques au sein du système par l'adoption du schéma spatial ISO 19107

Nous pensons que l'utilisation de ces standards favorisera grandement l'interopérabilité, ce qui permettra de faciliter l'accès aux données et diminuera la duplication des efforts notamment en favorisant la réutilisation de code lors du développement. Nous pensons que cela pourrait aussi aider à convaincre les organisations à adopter un tel système pour la gestion de leurs données géométriques 3D puisqu'un système s'appuyant sur les derniers standards en vigueur risque d'avoir une durée de vie plus longue ce qui peut faciliter la justification des efforts et des investissements que commande son développement.

En ce qui a trait au choix du WFS par rapport aux autres spécifications de services Web (comme le WMS, WCS), mentionnons qu'il est le seul de ces standards permettant d'extraire le modèle lui-même ainsi que ses propriétés descriptives afin d'avoir toute la latitude possible pour manipuler et interroger à sa guise le modèle au sein d'un outil performant tel un outil CAO. À terme, une spécification telle que le W3DS pourrait être

très intéressante pour un projet nécessitant avant tout la visualisation de modèles 3D et pouvant se contenter des capacités « limitées » d'un navigateur Internet. Lorsque l'interrogation, l'analyse et l'édition des modèles 3D sont nécessaires, l'accès aux données elles-mêmes (géométriques et descriptives) ainsi qu'à des opérateurs spatiaux tels que l'offre le WFS devient selon nous la meilleure option. Cependant lors du choix des technologies, cette spécification n'était pas encore disponible.

3.4 Architecture du SIG 3D proposé

Dans le cadre du développement de notre SIG 3D, nous proposons une architecture de type client-serveur qui rendra disponible sur le Web des composants applicatifs et des données par l'entremise d'une interface standard de service Web. La conception d'un tel système nécessite d'une part, la définition des applications et des données qui seront offertes (le service) et d'autre part la définition de l'interface standard elle-même. Le service offert aux clients (usagers du service) à partir d'un serveur (fournisseur de services) nécessite le recours à un protocole de communication, typiquement le HyperText Transfer Protocol (HTTP).

Le service proposé met à disposition sur le Web des modèles 3D et des traitements pouvant s'appliquer sur ceux-ci. L'architecture proposée permet à plusieurs clients d'accéder et de partager les mêmes données spatiales et alphanumériques en stockant celles-ci d'une manière centralisée au sein d'une même base de données accessible via un service Web. Alors que la partie cliente sert d'interface à l'utilisateur pour visualiser l'information, générer et lancer les requêtes (ex. : navigateur internet, logiciel client), la partie serveur inclut tout ce qui touche la logique applicative, le serveur Web ainsi que la base de données. Les applications serveurs et la base de données peuvent coexister sur une même machine ou être physiquement séparées. Dans un tel cas, on parlera alors d'architecture trois-tiers ou multi-tiers. Un des avantages importants de ce type d'architecture est la facilité de déploiement et de maintien du système puisque le cœur de celui-ci (la logique applicative) n'a pas à être installé sur chacun des postes clients. Cela signifie que lorsqu'une modification doit être apportée au programme, celle-ci est appliquée uniquement sur le serveur et une fois complétée, tous les clients ont immédiatement accès au programme actualisé.

L'adoption d'une telle architecture demande de différencier les fonctionnalités qui seront gérées côté client de celles gérées côté serveur. La figure suivante montre l'architecture générale du système avec ses principales composantes.

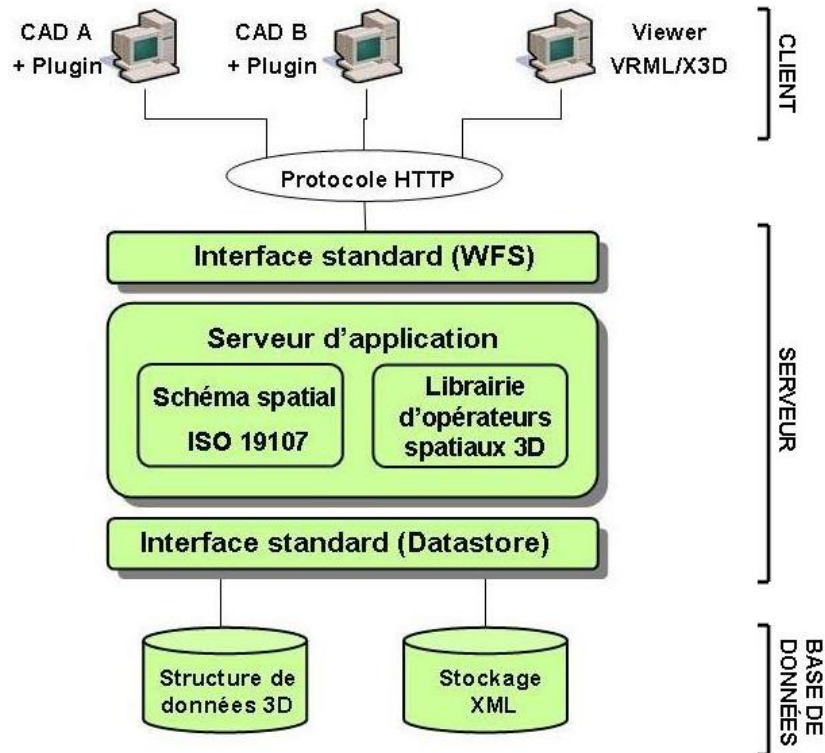


Figure 3 : Architecture multi-tiers du SIG 3D proposé.

Les sections suivantes décrivent plus en détail les composants clients et serveurs ainsi que les principales méthodes de stockage supportées par le système.

3.4.1 Le serveur

La partie serveur est le cœur de cette architecture et contient l'application sur laquelle repose l'ensemble du système. Cette application consiste en un assemblage de modules et de composantes ayant chacun des tâches bien précises dont le service Web, les *datastores* et la bibliothèque d'algorithmes d'opérateurs spatiaux 3D. Afin de pouvoir représenter les primitives géométriques ponctuelles, linéaires, surfaciques et volumiques dans un univers à trois dimensions, l'implémentation d'une structure de données 3D basée sur le schéma spatial ISO 19107 est proposée. Ce schéma est assez générique et met à disposition suffisamment de définitions de classes d'objets spatiaux de base, pour pouvoir représenter

la majorité des objets et des phénomènes en 3D de manière standard. L'ensemble des composantes du serveur est pris en charge par un serveur d'applications qui joue le rôle de moteur du système. Le serveur d'applications est responsable de l'interaction entre le client et la base de données. D'un côté, il doit pouvoir recevoir et gérer les requêtes des clients et de l'autre, il se charge de récupérer les données demandées au sein de la base de données avant d'effectuer les traitements et de renvoyer les résultats au client.

Nous verrons plus loin que ces requêtes du client reçues par le serveur par l'entremise du WFS contiennent des données spatiales en format GML. La mise en place d'un traducteur de données du côté serveur est donc nécessaire afin de convertir ces données GML provenant des requêtes XML du client en instance d'objets standards ISO 19107 et vice-versa. Ce module devra donc permettre la conversion des données géométriques :

- du format GML vers la structure de données standard ISO 19107
- de la structure de données standard ISO 19107 vers le format GML

Un parseur XML sera utilisé pour parcourir les documents XML provenant du client afin d'y récupérer l'information désirée. Ce parseur XML sera utilisé pour récupérer la donnée spatiale, mais aussi d'autres informations utiles côté serveur comme :

- le type de requête WFS (GetFeature, Insert, Delete, ...)
- le contenu de la requête elle-même permettant d'identifier :
 - le jeu de données désiré
 - le filtre sur les données
 - les opérations spatiales à appliquer

Dans les sections qui suivent, nous donnerons plus de détails sur les composantes principales du serveur comme le WFS, le *datastore* et l'analyse spatiale ainsi que sur les outils clients qui viendront s'y brancher.

3.4.2 Le Web Feature Service (WFS) 3D

L'interface standard du service Web sera implémentée suivant la spécification WFS de l'OGC. Cette spécification propose de standardiser le processus de dialogue entre le client

et le serveur pour donner accès sur internet à des données spatiales et à des opérateurs spatiaux pour leur traitement. Le WFS a été préféré aux autres spécifications standards puisque c'est le seul qui donne directement accès aux données elles-mêmes (données orientées-objet) et non seulement à une image de celles-ci comme le WMS ou le WTS. Cette standardisation s'effectue à plusieurs niveaux :

- technologie de requêtage (XML)
- définition des fonctionnalités disponibles (GetCapabilities, GetFeature, ...)
- syntaxe des requêtes et des réponses
- format des données géométriques (GML)

Bien qu'elle définisse le GML comme format standard de transfert des données géométriques, la spécification WFS elle-même ne mentionne pas la dimension (2D ou 3D) de la géométrie qui peut être exploitée par ce service. Avant l'arrivée du GML version 3 (GML 3), le WFS était essentiellement un outil qui visait l'accès à des données et à des traitements en 2D (i.e. des points, des lignes et des polygones). Maintenant que le GML 3 permet la représentation d'objets volumiques, on peut donc espérer pouvoir concevoir et implémenter des WFS 3D. Cela a été rendu possible par le fait que la spécification WFS a pris soin de séparer les concepts logiques de la donnée elle-même. En effet, le mécanisme de transaction et de requêtage entre le client et le service Web est complètement indépendant du format de données utilisé pour la représentation de la géométrie. Cela s'applique aussi pour les opérateurs spatiaux. Alors qu'on identifie les opérateurs qui peuvent être rendus disponibles par un WFS, leur implémentation sous-jacente n'est pas abordée ce qui laisse place à la diversité. On pourrait implémenter autant des algorithmes tirant profit de la topologie lors des calculs que des algorithmes 3D. Afin de rendre notre SIG 3D encore plus flexible, il serait éventuellement possible et intéressant d'implémenter plusieurs bibliothèques d'opérateurs spatiaux en parallèle (2D géométrique, 2D topologique, 3D géométrique, 3D topologique) dont l'utilisation serait dépendante de la structure géométrique du jeu de données requêté par l'utilisateur, et ce, de façon complètement transparente. Par exemple, si un modèle topologique 3D est requêté par le client, le système ferait appel à la bibliothèque d'opérateurs spatiaux topologiques 3D lors des calculs.

Pour résumer les fonctionnalités qui seront offertes par le SIG 3D de la part du WFS, les clients du système seront donc en mesure :

- de connaître les jeux de données ainsi que les opérateurs spatiaux disponibles sur un serveur (*GetCapabilities*)
- d'importer dans l'application-cliente des modèles 3D et leurs propriétés descriptives stockées dans une base de données (*GetFeature*)
- d'exporter de nouveaux modèles ou des modèles modifiés dans la base de données (*Transaction->Insert, Update*)
- d'éliminer des modèles de la base de données (*transaction->Delete*)
- d'appliquer des opérations spatiales sur les données

Cette façon de procéder pour éditer et mettre à jour les modèles fonctionne un peu à la manière du *check-in check-out* utilisée en 2D avec les technologies ArcGIS et ArcSDE d'ESRI. L'ensemble d'un modèle ou une partie de celui-ci est extrait de la base de données suite à une requête basée sur ses propriétés descriptives ou géométriques pour être copié et importé directement dans l'outil client. Si l'utilisateur possède les privilèges pour éditer le modèle, un *lock* est levé pour indiquer que la donnée est en cours de modification afin d'éviter la destruction mutuelle entre les utilisateurs. Une fois les mises à jour complétées, une requête *Insert* ou *Update* est envoyée au serveur pour stocker le modèle modifié dans la base de données. Le modèle à jour est ainsi rendu immédiatement disponible à tous les usagers.

3.4.3 Le client

La modélisation géométrique ainsi que les fonctionnalités qui lui sont associées comme la visualisation et la manipulation des modèles, seront gérées au sein de notre système du côté client pour plusieurs raisons :

- Parce que la navigation à travers un modèle 3D ainsi que sa manipulation demanderait un va-et-vient continu entre le client et le serveur pour obtenir la vue à jour du modèle et cela serait beaucoup trop coûteux en temps et en ressource

- Parce que le développement d'outils permettant la construction, l'édition et la validation des modèles est très complexe
- Parce qu'il existe déjà des logiciels comme les outils CAO qui font très bien le travail, qui sont robustes, qui ont des outils spécialisés dans un domaine d'application précis et qui sont connus des utilisateurs

Au niveau du choix de ce client, afin de tirer profit du fait que le système repose sur un ensemble de standards et spécifications favorisant l'interopérabilité, nous proposons de ne pas limiter le choix de ce client à une seule application spécifique. En effet, le système devrait permettre d'utiliser comme client n'importe quel outil de modélisation 3D de type CAO ou langage de modélisation 3D orienté Web comme le VRML ou le X3D. De cette façon, on confère au système une certaine flexibilité et portabilité, lui permettant de s'étendre ou de se connecter à d'autres systèmes. En permettant l'accès au système à une grande gamme de logiciels existants, on adapte celui-ci aux applications existantes au lieu de forcer ses usagers à apprendre un nouvel outil de travail ce qui peut en faciliter l'apprentissage et encourager son utilisation. L'utilisation d'un outil CAO existant comme logiciel client permettrait d'éviter de devoir développer de nouveau l'ensemble des fonctionnalités nécessaires à la construction, la manipulation et la visualisation des modèles 3D. Par contre, les structures de données implémentées dans chacun des outils CAO diffèrent pour chacun d'eux et il n'existe pas de format de données standard commun à tous ces logiciels. Pour permettre à chacune des applications clientes d'accéder aux modèles de notre système, il suffit de développer un programme permettant la conversion des données entre la structure standard de notre système vers la structure spécifique de chaque client. Ceci peut être réalisé par le développement d'un plug-in qui viendrait ajouter les fonctionnalités nécessaires à l'application cliente pour dialoguer avec le serveur par l'entremise du WFS et convertir les données.

En effet, alors que le service Web sur le serveur mettra à disposition des clients les requêtes standards du WFS, les applications qui seront utilisées comme clients devront avoir la capacité d'envoyer ces requêtes, recevoir les réponses du serveur et traduire les données spatiales entre le format GML et la structure propre au client. Cette intégration entre applications clientes et serveur WFS sera prise en charge par un module (qui devra être

développé) chargé d'effectuer la conversion entre les deux structures. Ce module devra donc permettre la conversion entre les données spatiales:

- du format GML vers la structure de données du client
- de la structure de données du client vers le format GML

Pour ce faire, un programme de conversion faisant appel à un parseur XML devrait normalement être implémenté dans l'environnement de développement de chacune des applications clientes afin de permettre cette traduction entre le format GML et la structure de données clientes. Par contre, ce programme de conversion peut s'avérer relativement complexe à réaliser et pourrait décourager un grand nombre d'utilisateurs désirant adapter un logiciel de modélisation 3D pour accéder à notre système. Le problème est que si l'on implémente le traducteur à l'intérieur même de l'application cliente, un nouveau traducteur devra être développé pour chacune d'elles puisque celles-ci peuvent mettre à disposition un langage de programmation différent et la majorité d'entre elles auront une structure de données différente. Afin de faciliter la conversion entre notre système et chaque application cliente, nous proposons l'implémentation d'un traducteur de données externe aux applications clientes qui permettrait de traduire les données géométriques du format GML en un format intermédiaire beaucoup plus simple. Ce format de données simplifié que nous proposons est basé sur le concept du format Well Known Text (WKT) proposé par l'OGC qui est un langage de balise permettant de représenter des objets géométriques sous forme de texte. Typiquement, une représentation WKT d'un objet géométrique contient le nom du type d'objet géométrique ainsi qu'une énumération des coordonnées qui le composent. La figure suivante montre des exemples de primitives géométriques représentées en WKT :

```
POINT (6 10)
LINESTRING (3 4,10 50,20 25)
POLYGON ((1 1,5 1,5 5,1 5,1 1), (2 2, 3 2, 3 3, 2 3,2 2))
MULTIPOINT (3.5 5.6,4.8 10.5)
```

Figure 4: Exemple de primitives géométriques représentées sous format WKT.

Ce format de stockage de données permet de réduire à sa plus simple expression, la définition de chacun des objets spatiaux d'un modèle. Bien qu'il ait été conçu à la base

pour représenter des objets spatiaux 2D, la syntaxe du WKT peut être étendue pour permettre la représentation d'objets volumiques (un des aspects particuliers de notre proposition). Afin de pouvoir représenter des solides sous format WKT, nous proposons l'ajout de deux primitives géométriques au format WKT soit le SOLID et le MULTISOLID. La figure suivante montre un exemple de représentation d'un SOLID et d'un MULTISOLID sous format WKT :

```
SOLID((0 0 0,0 2 0,1 1 2,0 0 0),(0 2 0,1 1 2,2 2 0,0 2 0),(2 2 0, 0 0 0,1 1 2,2 2 0),(0 0 0,0 2 0,2 2 0,0 0 0))

MULTISOLID(((0 0 0,0 2 0,1 1 2,0 0 0),(0 2 0,1 1 2,2 2 0,0 2 0),(2 2 0, 0 0 0,1 1 2,2 2 0),(0 0 0,0 2 0,2 2 0,0 0 0)),((0 0 0,2 0 0,2 2 0,0 0 0),(0 0 0,1 1 2,2 2 0,0 0 0),(2 2 0, 1 1 2,2 0 0,2 2 0),(0 0 0,2 0 0,1 1 2,0 0 0)))
```

Figure 5: Exemple de représentation de solides sous format WKT 3D.

Chaque sous-parenthèse d'un solide WKT 3D (SOLID) représente une face formant le solide. Par exemple, un cube ou un tétraèdre pourrait être représentés par les carrés ou les triangles qui les composent. Un MULTISOLID serait donc un assemblage de solides formant par exemple un volume remplis de tétraèdres. À noter qu'un autre jeu de parenthèses pourrait être ajouté pour représenter un trou à l'intérieur d'un solide. Le module de traduction de notre système sera donc en charge de parcourir les documents XML à l'aide d'un parseur et de convertir les propriétés géométriques des modèles entre le format GML et le WKT 3D (WKT étendu au 3D).

De cette façon, un seul et même module de conversion pourra être utilisé par tous les clients à condition d'utiliser un langage comme Java qui rend les programmes portables. La partie spécifique à chacun des clients qui restera à développer est une fonction interne à l'application cliente, capable de récupérer les coordonnées du format WKT et de recréer des objets géométriques dans la structure propre au client. Afin de pouvoir se brancher à notre système, chacun des clients devra donc être en mesure de :

- lire le format WKT 3D
- récupérer les coordonnées énumérées
- créer les objets spatiaux correspondants dans sa propre structure
- récupérer les coordonnées des objets spatiaux représentés dans sa propre structure
- écrire en format WKT 3D

De plus, chacun des clients devra pouvoir offrir une interface pour générer, recevoir et envoyer des requêtes WFS. Chacune de ces fonctionnalités peut typiquement être implantée au sein de l'application cliente sous forme de plug-in. La figure suivante montre le schéma de traduction de données du format XML/GML vers la structure de données propre au client :

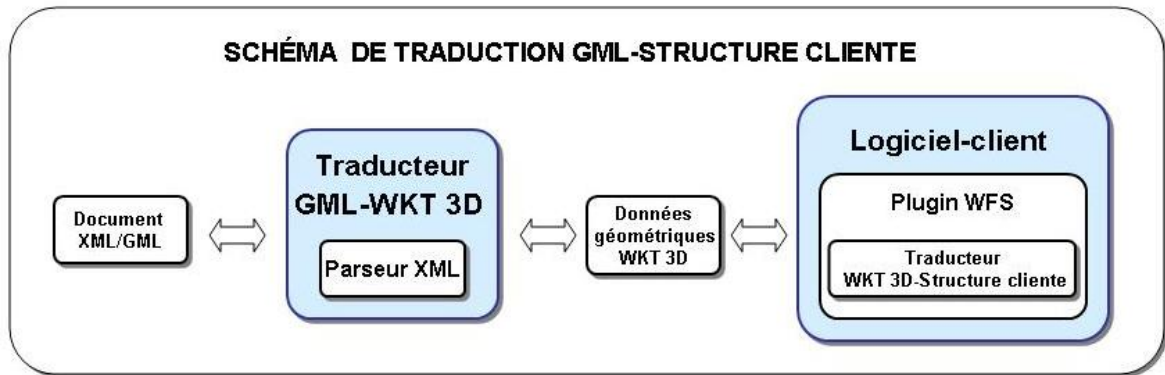


Figure 6: Schéma de traduction du format XML/GML vers la structure de données cliente

Pour ce qui est des propriétés descriptives reliées aux objets spatiaux, celles-ci sont aussi passées du côté client sous format XML lors des transactions avec le WFS. Ces données ne nécessitent toutefois pas de conversion particulière et pourront être accédées directement à partir de l'application cliente par la simple lecture de fichiers textes.

3.4.4 Le Datastore

Du côté stockage de l'information, le système est conçu pour pouvoir éventuellement se brancher à n'importe quelle base de donnée ou structure de données grâce au concept appelé *Datastore*. Un *datastore*, concept intrinsèque aux bases de données et particulièrement au monde des entrepôts de données, est une interface standard qui joue le rôle de tampon entre la base de données et le serveur d'applications. Son rôle est d'encapsuler l'accès à la base de données afin de masquer la complexité et la particularité de chaque source de données à l'utilisateur. C'est donc à l'intérieur du *datastore* que l'on gère le *mapping* des objets spatiaux entre la structure du schéma spatial standard du système et de la structure de la base de données. Concrètement, un *datastore* typique contient le code permettant de générer les requêtes SQL propres à un SGBD particulier afin de venir

extraire ou modifier les données spécifiées par le client. Connaissant l'information permettant de faire la correspondance entre les entités spatiales de la source de données et les entités correspondantes de la structure de l'ISO 19107, le *datastore* est donc aussi en charge d'appeler le constructeur de la classe d'objets ISO à créer pour chaque objet spatial qui est extrait. Une fois créés en mémoire les objets spatiaux en format ISO, le système peut procéder aux traitements (ex. : analyse spatiale).

L'utilisation du concept de *datastore* combiné à l'adoption du schéma spatial de l'ISO vient ajouter à la flexibilité et à la généricité du système en permettant théoriquement à n'importe quelle source de données de structure différente et implantée dans n'importe quel SGBD ou fichier (ex. : Oracle Spatial, SQLServer, PostGIS, MySQL, XML ...) de pouvoir être utilisée dans le système. Il suffit de créer un *datastore* propre à chaque source de données que l'on désire intégrer au système.

Ainsi, le concept de *datastore* permet à notre système d'avoir recours à diverses méthodes de stockage. Cependant, le stockage des modèles 3D au sein d'une base de données étant problématique, une solution doit être envisagée. Afin de donner un aperçu de quelques structures de données pouvant être utilisées pour le stockage des modèles 3D, trois méthodes de stockage différentes au sein d'un SGBD sont identifiées soit le langage XML, l'implémentation d'une structure topologique 3D ou l'ajout d'une primitive volumique.

Si on débute avec le XML, deux visions différentes existent lorsque vient le temps de joindre le XML au monde des bases de données. L'une, plus conventionnelle, importe et exporte la donnée en XML à partir d'un *mapping* avec des données stockées d'une manière traditionnelle dans une base de données relationnelle standard. L'autre stocke directement le XML dans la base de données. Le stockage en XML est maintenant de plus en plus utilisé pour la simple raison qu'il est intéressant de stocker l'information directement dans le format dans lequel elle est destinée à être diffusée afin d'en éviter la conversion pour son transfert ou sa diffusion sur le web. Il existe plusieurs façons de stocker du XML dans une base de données :

- Dans un champ texte d'une base de données relationnelle standard
- Dans une base de données XML-compatible (ex : Oracle, DB2, SQLServer)

- Dans une base de données XML native (ex. : Tamino)

Le XML étant du texte, la manière la plus simple de le stocker est dans un champ texte d'une table relationnelle standard. Afin d'optimiser le stockage du XML, plusieurs SGBD comme Oracle, DB2 et SQL Server fournissent maintenant un *datatype* XML permettant de stocker du XML dans une colonne d'une table relationnelle au même titre qu'un entier ou qu'une date. Ces bases de données sont appelées XML-compatibles par opposition avec les bases de données XML natives qui utilisent des documents XML comme unité fondamentale de stockage. Le stockage en XML natif ou l'utilisation d'un *datatype* XML permet d'effectuer des recherches rapides à l'intérieur du XML pour y extraire un sous-ensemble de l'information à l'aide des technologies comme le XPath et le XQuery (recommandations W3C) qui sont maintenant intégrées dans les SGBD. Les fournisseurs de bases de données XML-compatibles comme Oracle améliorant sans cesse leur produit, il devient difficile aujourd'hui de distinguer les bases de données XML-compatibles de celles dites natives XML.

La méthode de stockage de modèle 3D en XML est décrite ici pour sa simplicité et sa facilité d'implémentation. Elle est dite relativement simple parce qu'elle implique le stockage des modèles 3D directement en XML formaté selon la norme ISO GML 3 dans une base de données. Elle est dite relativement facile d'implémentation puisque n'importe quel SGBD peut être utilisé pour le stockage, mais aussi pour une raison de récupération de code existant. En effet, le SIG 3D que nous proposons, à travers le WFS, doit permettre de recevoir des objets spatiaux en format GML à partir de n'importe quel client WFS-compatible et de convertir ce GML en instance d'objets du schéma spatial ISO 19107 et vice-versa. Ces fonctions de conversion implémentées au sein du serveur d'applications pourraient donc être réutilisées par un *datastore* pour convertir les modèles géométriques 3D stockés en GML dans une base de données vers le serveur d'applications lors de requête *GetFeature* et l'inverse lors d'une requête *Transaction* comme un *Insert*. Ces fonctionnalités de conversion devant de toute manière être implémentées dans le cadre du système proposé, le stockage en XML devient une alternative intéressante pour implémenter rapidement un *datastore* et stocker relativement facilement un modèle 3D dans un SGBD. Lors du stockage de la géométrie en XML dans un champ d'une table

relationnelle, les propriétés descriptives reliées aux objets spatiaux peuvent être stockées dans les champs adjacents de la même table (voir figure 11) ou encore stockées dans une table différente qui peut être reliée à la table contenant la géométrie à l'aide d'une jointure.

ID	geometry	lithology	litho_desc	stratigraphy	stra_desc
3	<gml:Solid xmlns:gml="http://I3A	GABBRO	[arch]3	Archéen	
5	<gml:Solid xmlns:gml="http://V3B	BASALTE	[arch]dg1	Archéen, Formation de Deguisier	
2	<gml:Solid xmlns:gml="http://I3A	GABBRO	[prot]	Protérozoïque	
1	<gml:Solid xmlns:gml="http://V2J	Andésite	[arch]dg4	Archéen, Formation de Deguisier	
4	<gml:Solid xmlns:gml="http://V2J	Andésite	[arch]dg4	Archéen, Formation de Deguisier	

Figure 7 : Exemple de stockage de la géométrie d'objets spatiaux en GML et de leurs propriétés descriptives dans une table relationnelle.

Le principal inconvénient de cette méthode de stockage est la difficulté d'extraire seulement une partie de la géométrie encodée en GML étant donné l'inexistence d'index spatial. Pour contourner ce manque, on peut implémenter une version « simplifiée » d'un index spatial à condition de pouvoir connaître et stocker la boîte englobante minimum de chacun des objets insérés dans la base de données. Cette information peut être obtenue au moment de l'insertion dans la base de données lorsque nos objets spatiaux se trouvent sur le serveur d'applications sous la forme d'instances d'objets ISO 19107. En effet, le schéma spatial ISO 19107 prévoit l'implémentation d'une méthode permettant l'obtention de la boîte englobante de n'importe quel objet spatial représentée selon son schéma. Une fois stockée, cette boîte englobante de chaque objet géométrique pourrait être utilisée pour effectuer un premier filtrage grossier des objets lors de l'extraction des données (avec ou sans usage d'opérateurs spatiaux). De cette façon, les opérateurs spatiaux peuvent effectuer leurs calculs faisant intervenir la géométrie complète des objets uniquement sur ceux n'ayant pas été mis de côté lors du premier filtrage grossier afin de simplifier et d'améliorer la vitesse des requêtes. Cette méthode, bien que très utile pour améliorer la performance lors de l'extraction et l'application d'opérateurs spatiaux sur un modèle, ne remplace pas un réel index spatial 3D qui par le biais d'une décomposition de l'espace à la manière de l'Octree (version 3D du Quadree utilisé en 2D dans les SGBD Spatiaux), offrirait certainement un meilleur rendement. De plus, calculer la boîte englobante de chaque volume avant son insertion dans la base de données peut aussi ralentir la sauvegarde des modèles. Les détails de l'implémentation d'un index spatial 3D au sein d'une base de

données ont déjà été abordés dans la littérature et sorte du cadre de ce projet de recherche puisque l'implémentation d'un tel index représenterait possiblement un projet de maîtrise en soi.

La seconde structure de stockage proposée consiste en l'adoption d'une structure topologique 3D. La topologie peut grandement faciliter l'analyse spatiale en stockant explicitement les relations entre objets spatiaux voisins d'un modèle en plus d'assurer la cohérence et optimiser le stockage en évitant la redondance dans les coordonnées. La construction et l'analyse de modèles topologiques 3D peuvent donc être très utiles et sont déjà envisagées dans plusieurs domaines d'application. Pour ces raisons, l'architecture du SIG 3D proposée supporte aussi par sa grande flexibilité et généricité le stockage et l'analyse de modèle topologique 3D. Pour ce faire, on doit pouvoir stocker au sein d'une base de données un modèle topologique 3D et offrir des opérateurs spatiaux 3D topologiques. Le schéma spatial ISO 19107 et GML 3 supportant pleinement la topologie, la représentation des modèles topologiques au sein du système et le partage de ceux-ci sur internet peuvent donc être envisagés à condition d'implémenter les classes correspondantes et d'adapter les outils de conversion entre GML et ISO 19107. Il existe un grand nombre de structures géométriques topologiques 3D comme le TEN, SSM, 3D FDS ou GeoTen qui pourraient éventuellement être implémentées dans notre SIG 3D. Pour pouvoir utiliser des modèles utilisant une de ces structures dans notre système, il suffirait de dresser un *mapping* entre les primitives géométriques de cette structure avec celles du modèle ISO, c'est-à-dire les primitives topologiques comme le TP_Solid ou la TP_Surface par exemple. Un *datastore* pourrait ensuite être développé pour permettre l'extraction de la base de données et effectuer la conversion entre les deux structures. Comme pour le stockage en XML, une structure topologique pourrait être implémentée dans n'importe quel SGBD et nécessiterait l'implémentation d'un index spatial 3D pour être optimale.

Une troisième méthode de stockage consiste à stocker un modèle 3D au sein d'un SGBD spatial en utilisant une primitive volumique. Une telle primitive peut être implémentée dans un SGBD comme Oracle Spatial tel que proposé par Arens *et al.* en 2003. Il est maintenant toutefois possible d'opter pour un SGBD offrant nativement une telle primitive volumique comme c'est le cas avec la version 11g d'Oracle Spatial. Dans un cas ou dans l'autre, le

recourt à la bibliothèque d'opérateurs spatiaux 3D du serveur d'applications de notre SIG 3D serait toujours nécessaire puisqu'aucun SGBD n'offre encore d'opérateurs spatiaux topologiques tenant compte de la troisième dimension. De plus, comme pour chaque méthode de stockage différente, l'implémentation d'un *datastore* au sein du système serait nécessaire. Ce *datastore* devrait être capable de faire le *mapping* et la conversion entre les primitives géométriques du SGBD (dont la nouvelle primitive volumique implémentée) et celles d'ISO.

3.4.5 L'analyse spatiale

Dans l'architecture proposée, le serveur est le tiers qui est responsable de la majorité des traitements dont les calculs d'analyse spatiale 3D comme par exemple les calculs d'intersection, d'inclusion ou de chevauchement entre solides. Plusieurs raisons sont à la source de notre choix d'exécuter les opérateurs spatiaux côté serveur :

- La plupart des outils de modélisation 3D possèdent peu ou pas d'opérateur d'analyse spatiale 3D. Ces clients pourront donc profiter des algorithmes mis à disposition côté serveur
- Implémenter une bibliothèque d'algorithmes d'opérateurs spatiaux 3D sur le serveur permet de standardiser les calculs d'analyse spatiale en donnant accès à tous les clients à la même gamme complète d'opérateurs spatiaux 3D
- Les calculs d'analyse spatiale 3D pouvant être très lourds, le client bénéficie des capacités accrues en puissance et en mémoire du serveur pour effectuer ses traitements

Ainsi, chaque application cliente qui accèdera au système se verra offrir par l'entremise du service Web la liste des opérateurs spatiaux disponibles sur le serveur. Peu importe la structure géométrique de l'application cliente ou de celle utilisée pour le stockage des modèles, tous auront accès aux mêmes opérateurs spatiaux standards. Cela sera rendu possible par le fait que les traitements seront effectués sur les modèles convertis en objets spatiaux standards du schéma spatial de l'ISO et les résultats envoyés par la suite en format standard (GML) par le biais du service Web.

Au niveau des opérations spatiales qui seront disponibles dans le système, l'OGC propose dans la spécification *OpenGIS Simple Feature Specification for SQL* une série de méthodes

classiques pour tester les relations entre objets spatiaux: Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, Contains ainsi que des opérateurs spatiaux: Distance, Buffer, Convex Hull, Intersection, Union, Difference, Symmetric difference. Ces méthodes et opérateurs spatiaux sont principalement destinés à être utilisés dans les applications et bases de données 2D implémentant le langage SQL afin d'y ajouter des capacités de manipulation de données géométriques conjointement à la gestion des primitives de type point, ligne et polygone. À cette liste, l'OGC ajoute plus récemment les opérateurs DWithin, Beyond et BBOX. Cette liste d'opérateurs spatiaux est reprise par l'OGC dans le cadre des spécifications WFS et Filter Encoding afin de définir les opérations spatiales qui pourraient être rendues disponibles par un service Web WFS. Bien qu'à la base ces opérateurs spatiaux étaient destinés à implémenter des algorithmes 2D, nous proposons d'implémenter ces méthodes et opérateurs spatiaux dans notre système avec une algorithmie tridimensionnelle sous-jacente. Toutefois, comme mentionné précédemment, il n'existe pas actuellement de bibliothèque complète d'algorithmes d'opérateurs spatiaux tridimensionnels (géométriques et topologiques) comme c'est le cas en 2D avec par exemple avec la bibliothèque de Vivid Solutions, la JTS. Il existe bien des bibliothèques regroupant divers types d'algorithmes destinées à la manipulation d'objets spatiaux 2D et 3D comme CGAL (Computational Geometry Algorithms Library) ou Open Cascade mais celles-ci sont incomplètes pour nos besoins puisqu'elles ne permettent pas de calculer la proximité, l'intersection, l'inclusion, l'égalité ou l'adjacence d'objets géométriques 3D, soit les opérations identifiées plus tôt comme essentielles dans l'implémentation de notre SIG 3D.

La création, l'adaptation ou l'intégration d'une telle bibliothèque d'opérateurs spatiaux 3D représente un travail de grande envergure étant donné la complexité accrue qu'implique l'introduction de la 3^e dimension dans les calculs d'analyse spatiale. La non-disponibilité d'une telle bibliothèque est un frein au développement du domaine 3D et représente un obstacle majeur pour un grand nombre de projets faisant intervenir des données spatiales tridimensionnelles. Ainsi, le système que nous proposons est conçu pour éventuellement intégrer et implémenter une bibliothèque d'opérateurs spatiaux 3D. Toutefois, le développement d'une telle bibliothèque ne fait pas partie des objectifs de ce projet de recherche, mais sera plutôt envisagé lors de travaux de recherche futurs.

Chapitre 4 Implémentation et validation dans un contexte de modélisation géologique

Afin de démontrer la faisabilité et la validité de cette architecture orientée service Web 3D, nous allons montrer dans ce chapitre une implémentation réalisée pour répondre à des besoins spécifiques d'un client (le Ministère des Ressources naturelles et de la Faune du Québec - MRNF- Secteur Géologie). Le Web Geological Feature Server (WGFS) répondra ainsi à des besoins de modélisation géologique 3D effectuée à partir de la plate-forme de modélisation Gocad et d'une base de données géoscientifiques (SIGEOM). Voici donc des explications détaillées sur ces choix. Nous présenterons d'abord les besoins spécifiques liés au domaine d'application choisie soit celui de la modélisation géologique 3D. Suivront ensuite les choix technologiques, les détails sur les développements réalisés, le fonctionnement du système, son évaluation et ses limites.

4.1 Besoins liés à la modélisation géologique 3D

Le MRNF travaille depuis quelques années à la construction de modèles géologiques 3D à l'aide de la plateforme de modélisation Gocad (<http://www.pdgm.com/>). Comme nous l'avons mentionné au chapitre 2, Gocad est un logiciel CAO spécialement conçu pour la modélisation géologique et beaucoup utilisé au niveau gouvernemental, académique et privé (Fallara *et al.*, 2004; Caumon *et al.*, 2004; Lachance 2005; Bédard 2006; Sprague *et al.*, 2006; Pouliot *et al.*, 2008). Il permet, entre autres, de modéliser des réservoirs d'hydrocarbure et des forages, faire de la visualisation séismique et optimiser le choix de l'emplacement des plates-formes de forage. La modélisation 3D dans Gocad repose

principalement sur les structures géométriques B-Rep, Voxel, et la représentation tétraédrique (Lachance 2005; Bédard 2006; Pouliot *et al.*, 2008).

Pour créer ces modèles 3D, les géologues québécois exploitent divers types de données telles que les données de forages ou les cartes géologiques 2D généralement obtenues à partir de la base de données SIGEOM. SIGEOM est un système d'information à référence spatiale qui contient les données géoscientifiques québécoises recueillies depuis près de 150 ans (http://sigeom.mrnf.gouv.qc.ca/signet/classes/I1102_indexAccueil). Entre autres, SIGEOM regroupe le fond documentaire, l'ensemble des documents papier et électroniques ainsi que les données spatiales traitant du milieu géologique québécois. Ce système permet aux acteurs miniers d'accéder à des cartes géologiques, géophysiques et géochimiques, des indices minéralisés, des cartes de localisation et des descriptions de forages au diamant, diverses analyses et des documents géoscientifiques. SIGEOM permet de commander ou télécharger ce genre d'information directement à partir du Web (par le biais de l'application e-SIGEOM).

SIGEOM est devenu pour les géologues appelés à réaliser la modélisation géologique 3D du MRNF une source de données de premier ordre pour la construction des modèles 3D. Un des problèmes est que ce système n'a pas été conçu pour gérer et stocker des modèles 3D. Étant donné les efforts considérables qui sont investis pour la construction de ces modèles géologiques 3D et la richesse d'information qu'ils apportent, l'incapacité de SIGEOM à pouvoir gérer ces modèles devient problématique. En effet, les acteurs du domaine de la modélisation géologique 3D aimeraient pouvoir stocker et donner accès à leur modèle d'une manière convenable en retournant ceux-ci au sein même de la base de données d'où ils ont extrait les données de base ayant servi à leur construction. Actuellement, les modèles géologiques 3D sont rendus disponibles via un lien direct sur les modèles. Les utilisateurs doivent donc savoir que ce modèle existe et ils doivent posséder le logiciel Gocad pour pouvoir le consulter (Thomas et Nyaminani, 2008).

Le MRNF utilise donc le modélisateur géologique 3D Gocad pour la construction de leurs modèles et désire continuer à l'utiliser pour le futur. Le choix du logiciel Gocad comme client de notre prototype de SIG 3D devenait donc une première contrainte à respecter. Puis la littérature mentionne que des approches de modélisation géométrique de type B-Rep ou

tétraédrique sont bien appropriées pour des exercices de modélisation géologique couplant des bases de données (Pouliot *et al.*, 2006a; Bédard, 2006; Pilouk, 1996). Comme nous désirions exploiter une approche base de données et conserver les propriétés descriptives (issues de SIGEOM), il convenait mieux d'utiliser des primitives géométriques de type vectorielles soit les points, lignes, polygones (ou triangles) et polyèdres (ou tétraèdres). C'est ainsi que la représentation tétraédrique a été retenue et qu'un modèle géologique 3D complexe de la région de Porcupine-Destor (Nord-Ouest du Québec, Canada) contenant des points, des lignes, des surfaces et des solides tétraédriques a été construit avec Gocad (participation spéciale de Bédard 2006 et de Fallara *et al.* 2004 pour la construction de ces modèles 3D). Ce modèle géologique couvre un territoire d'environ 3,9 km cube (2 360m x 1 740m x 949m) et est composé de 16 solides contenant un total de près de 14 000 tétraèdres. La figure qui suit montre une partie du modèle tétraédrique de Porcupine-Destor tel que modélisé dans Gocad.

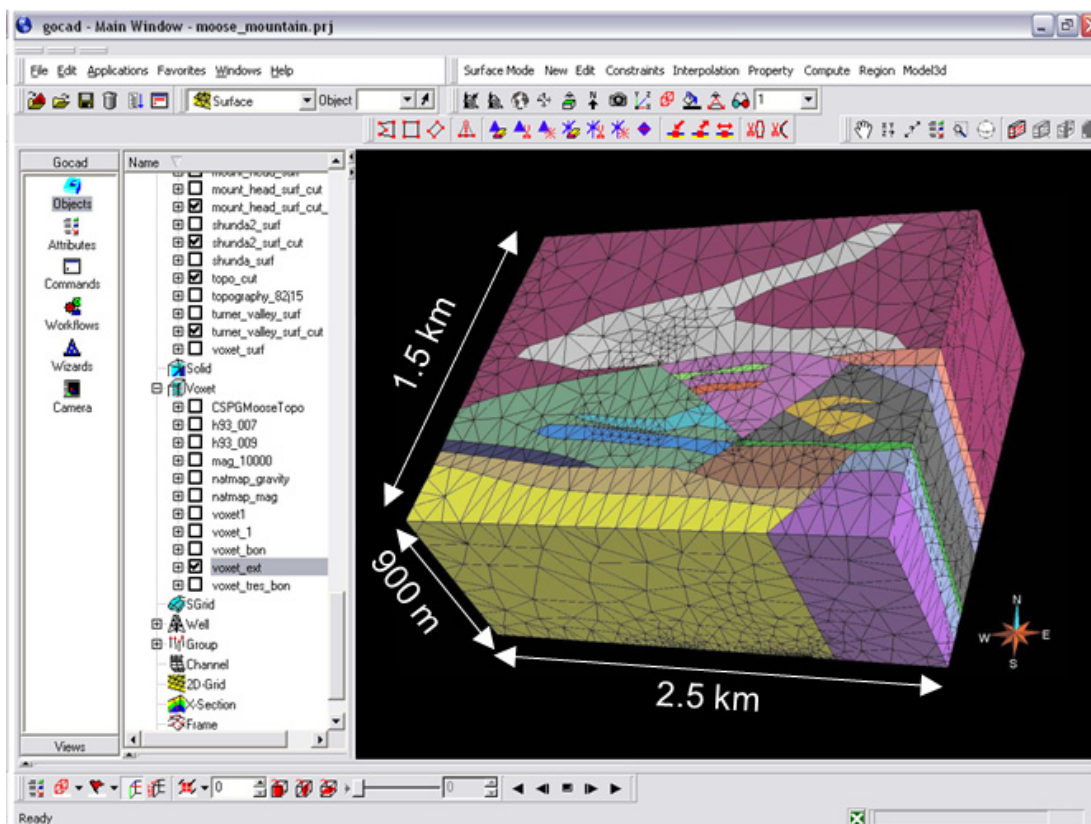


Figure 8: Modèle tétraédrique de la région Porcupine-Destor Nord-ouest du Québec (modifié de Fallara *et al.*, 2004) tel que visualisé dans l'interface Gocad.

Le prototype WGFS sera donc appliqué dans ce contexte.

4.2 Architecture du Web Geological Feature Server (WGFS)

Le prototype WGFS a été conçu suivant une architecture trois-tiers tel que présenté au chapitre précédant. Il est composé d'un client, d'un serveur d'application et d'une base de données (voir figure 9). Il a été développé dans l'optique d'être indépendant des systèmes d'exploitation, en plus d'être ouvert, flexible et extensible puisqu'il peut être accédé par un grand nombre de clients différents et permet de se brancher théoriquement à n'importe quelle base de données ou structure de stockage à condition d'en implémenter les spécificités dans un *datastore*. Il est aussi dit interopérable puisqu'il implémente plusieurs normes et spécifications standards de l'ISO et de l'OGC. Le premier client du système consiste en le logiciel de modélisation 3D de type CAO, Gocad, combiné à un plug-in implémenté dans l'environnement de développement de ce logiciel et d'un traducteur de données GML-WKT 3D. L'application côté serveur est composée de plusieurs modules dont un WFS 3D, une bibliothèque d'opérateurs spatiaux 3D contenant un premier algorithme, le BBOX, et l'interface du *datastore*. La géométrie côté serveur repose sur l'implémentation des classes d'entités 2D et 3D du schéma spatial ISO 19107. La base de données est composée d'un modèle géologique tridimensionnel représenté sous format XML couplé à des données alphanumériques provenant de la base de données SIGEOM stockées dans un SGBD relationnel standard.

Architecture du système

3D Web Geological Feature Server (WGFS)

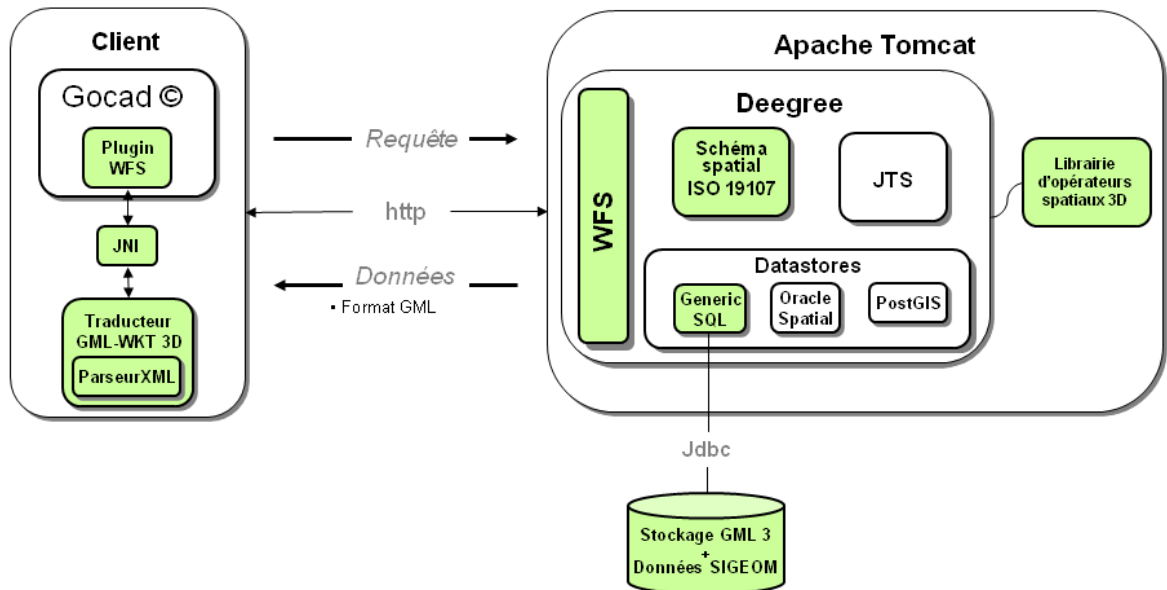


Figure 9: Architecture du Web Geological Feature Server proposée pour le stockage, le requêtage et la diffusion de modèles géologiques 3D

Les éléments représentés en vert sont ceux qui ont été créés ou modifiés dans le cadre de notre développement du WGFS.

4.3 Choix des technologies

Le système a été développé à l'aide de technologies et composants Java. Java est un langage orienté-objet, possédant de grandes aptitudes côté Web et spécialement conçu pour permettre le développement d'application portable. Aussi souvent que possible, des technologies distribuées gratuitement sous licence Open Source ont été privilégiées pour le développement du WGFS. Les sections suivantes présentent plus en détail les technologies qui interviennent dans chacun des tiers du système.

4.3.1 Le client

Comme mentionné précédemment, le premier client utilisé pour tester le prototype du WGFS est le logiciel CAO Gocad puisqu'il est l'outil de modélisation 3D utilisé par le MRNF.

Le traducteur de données qui sera chargé de convertir les données format XML/GML vers le format WKT-3D présenté au chapitre précédant, sera développé en Java afin de pouvoir profiter de la portabilité que confère l'utilisation de ce langage. Pour parcourir les documents XML et y récupérer l'information désirée, le traducteur de données fait appel à un parseur XML. Il existe deux familles de parseur XML ayant chacune leurs qualités et leurs défauts soit les parseurs SAX (Simple API for XML) et les parseurs DOM (Document Object Model). Le parseur DOM permet de traduire et garder en mémoire un document XML complet sous forme d'arbre binaire. Le principal avantage du parseur DOM est que sa structure d'arbre binaire permet de manipuler relativement facilement de façon programmatique le contenu du document XML. Un parseur DOM permet donc de simplifier significativement la navigation à l'intérieur d'un document XML. Par contre, ce genre de parseur est réputé être gourmand en mémoire puisque l'ensemble du document XML doit y être stocké. Pour sa part, le parseur SAX ne stocke pas en mémoire l'entièreté d'un document XML mais parcourt plutôt l'ensemble d'un document en générant un flux d'événements auquel peut réagir librement le programme, à la rencontre de chacun des éléments qu'il contient (début et fin de document, balises ouvrantes, balises fermantes, contenu de balise, etc.). Ce type de parseur permet de concentrer nos efforts sur la partie spécifique qui nous intéresse d'un document XML en laissant tomber les parties de document qui ne sont pas utiles pour l'application en cours. Ainsi, et grâce au fait que l'entièreté d'un document XML n'a pas à être stockée en mémoire, les parseurs SAX sont généralement plus performants. Par contre, ils n'offrent pas la facilité de manipulation des parseurs DOM et sont souvent un peu plus complexes à implémenter. La représentation d'un modèle 3D sous format GML peut facilement générer des documents XML de taille énorme. Pour des raisons de performance et d'économie de mémoire, nous avons choisi d'implémenter côté client de notre système le parseur de type SAX Xerces. Xerces (version 1.4.4) est un parseur Java, gratuit et Open Source. Il est un des parseurs Java réputés les plus performants et est beaucoup utilisé dans les différents projets de la communauté Open Source. Les algorithmes de conversion de données seront donc programmés et intégrés à ce parseur.

Notre traducteur de données fait aussi appel à une autre bibliothèque Java Open Source de la Fondation Apache soit la bibliothèque HttpClient. Cette bibliothèque implémente les

plus récents standards et recommandations du protocole HTTP et fournit les classes et méthodes de base pour permettre le développement d'application cliente communiquant à travers le Web. C'est à partir de cette bibliothèque que seront programmées les fonctionnalités permettant d'envoyer au serveur les requêtes WFS construites par le client.

Comme Gocad est développé en C++ et pour des raisons de portabilité, il a été décidé de développer le traducteur de données à l'aide des technologies Java, une tierce composante devait être mise en place pour faire le pont entre les deux langages. Pour ce faire, la Java Native Interface (JNI) qui est une technologie permettant à du code Java d'appeler et d'être appelé par des applications natives ou autres langages comme le C, le C++ ou l'assembleur, a été utilisée.

4.3.2 Le serveur

L'application centrale du WGFS, soit la logique applicative du système, repose sur Deegree, une plate-forme de développement standard Java et Open Source qui implémente plusieurs des normes et spécifications de l'ISO et de l'OGC. Deegree, comme ses concurrents GeoServer et RedSpider, propose entre autres une implémentation du WMS, du WFS et du WCS de l'OGC ainsi que du schéma spatial ISO 19107. Il implémente aussi la mécanique de l'interface du *datastore* ainsi que plusieurs *datastores* spécifiques permettant de se brancher aux principaux SGBD et formats de fichier comme Oracle, PostGIS, MySQL, Shapefile, etc. Deegree a été choisi parce qu'il est gratuit et que nous avons besoin d'un produit dont le code source était disponible afin de pouvoir le modifier pour le rendre 3D-compatible. RedSpider a été rejeté parce que c'est un produit commercial et GeoServer bien qu'étant gratuit et Open Source, présentait à ce moment une implémentation moins complète du schéma spatial de l'ISO. L'ensemble de l'application côté serveur, a été déployé au sein du bien connu et très répandu serveur d'applications Java et Open Source Tomcat de la fondation Apache (<http://tomcat.apache.org/>). Tomcat, est un conteneur de Servlet et de JSP qui fournit la technologie permettant de déployer et de rendre disponible l'application serveur du WGFS sur le Web.

4.3.3 La base de données

Le WGFS, grâce au concept de *datastore* qu'il implémente par l'intermédiaire de Deegree, est un système conçu pour pouvoir théoriquement se brancher à n'importe quelle source de données, que ce soit une base de données ou un fichier. Un premier *datastore* a été adapté pour permettre l'interaction avec un modèle géologique 3D stocké au sein du SGBD MySQL (<http://www.mysql.com/>). MySQL est le SGBD gratuit et Open Source le plus utilisé dans le monde. Il a été choisi parce qu'il est performant, mais surtout parce qu'il est très simple à implanter et à utiliser. MySQL fournit une interface graphique très intuitive qui permet d'administrer facilement la base de données.

4.4 Développement côté client

La figure suivante montre le schéma détaillé des composantes logicielles côté client. Elle présente aussi les différentes étapes de transformation de l'information entre les documents XML générés ou reçus par le serveur et le logiciel Gocad.

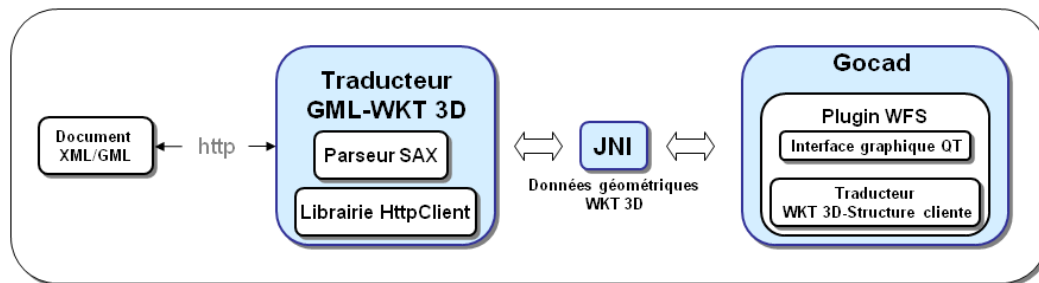


Figure 10 : Schéma détaillé du passage des données entre le document XML et l'application cliente Gocad.

4.4.1 Le plug-in Gocad

Gocad offre aux développeurs, par l'intermédiaire du *Gocad developer kit*, une API contenant les classes et les méthodes permettant d'étendre les capacités du logiciel par le développement de nouvelles fonctionnalités à l'aide du langage C++. Entre autres, il offre un mécanisme permettant le développement de nouveaux modules, appelés plug-ins, regroupant de nouvelles fonctionnalités qui viennent s'ajouter aux modules de base du logiciel. Les modifications apportées au logiciel Gocad afin d'en faire un client du WGFS ont été apportées sous forme de plug-in. Une des nouvelles fonctionnalités implémentées

consiste en une fonction qui reçoit en entrée une chaîne de caractères contenant les coordonnées formatées d'une manière semblable à la spécification WKT (étendue pour permettre la gestion de solides) et fournit en sortie des objets spatiaux propres à la structure de Gocad tels que des TSolid, TSurf, TFace, etc.

Un autre ajout au logiciel consiste en une interface graphique, développée en QT (<http://www.qtsoftware.com/>), permettant à l'utilisateur de se brancher au serveur WGFS, de générer et envoyer des requêtes WFS ainsi que de recevoir les réponses du serveur. Cette interface graphique permet d'envoyer, par l'intermédiaire de JNI, l'information entrée par l'utilisateur ainsi que la définition des objets géométriques d'un modèle 3D au module Java en charge de la traduction en GML des données et de l'envoi des requêtes au serveur.

Le plug-in ajoute aussi au logiciel Gocad un bouton «Identifier» typique aux SIG, permettant de connaître les propriétés descriptives rattachées à un objet spatial. Lors d'un clic sur l'objet, l'identifiant de cette entité est récupéré et utilisé pour venir lire l'information s'y rattachant stockée côté client sous format texte lors de l'importation du modèle.

4.4.2 Le traducteur de données

Tel que mentionné précédemment, un module Java côté client a été développé pour traduire les données GML provenant du serveur en notre format WKT-3D. Un parseur SAX a été utilisé pour récupérer l'information GML contenue dans les documents XML reçus du serveur. Notre application a donc été programmée pour réagir à chaque fois qu'une balise représentant un objet spatial de la syntaxe du GML est rencontrée. Ce programme peut donc, lors de la rencontre d'une balise <gml:Solid> par exemple, récupérer la liste des coordonnées de cette entité et l'envoyer au plug-in Gocad dans une chaîne de caractères après un reformatage en WKT-3D. Le programme récupère aussi l'identifiant et la liste des propriétés descriptives reliées à chaque objet spatial et va écrire cette information dans un fichier texte côté client. C'est cette information qui sera récupérée et montrée à l'utilisateur lors d'un clic sur un objet spatial à l'aide du nouveau bouton Identifier ajouté par notre plug-in dans Gocad. Le module Java en charge de traduire les données XML, contient aussi un programme faisant appel à la bibliothèque HttpClient, permettant d'envoyer du XML

sur le Web utilisant le protocole HTTP. Ce programme permet aussi de recevoir les réponses sous format XML provenant du serveur. Ce sont ces documents XML en provenance du serveur WGFS qui sont parcourus par notre parseur XML qui récupère, traduit et envoie l'information au plug-in Gocad par l'intermédiaire de JNI.

4.4.3 La structure Gocad versus GML

Afin de pouvoir effectuer la conversion entre les entités GML et celles de la structure Gocad, il est important de bien connaître les primitives géométriques de cette structure et la manière dont elles sont assemblées. Si l'on ne s'attarde qu'aux objets de base dits « vectoriels » de la géométrie de Gocad, on peut distinguer ceux-ci en deux groupes (Gocad Developer Guide, 2005). Un premier groupe contient les « Application Objects » c'est-à-dire les objets que l'utilisateur peut créer et visualiser dans Gocad. Ce groupe inclut les VSet, Pline, Tsurf et Tsolid qui d'un point de vue vectoriel pourrait être décrit respectivement en tant que point, ligne, surface et solide. L'autre groupe, contenant les « Topological Elements » (Node, Edge, Trgl, Tetra) et « Topological Objects » (Iline, Tface, Tvolume) fait référence aux objets qui ne sont pas directement instanciables par l'utilisateur, mais sont plutôt utilisés d'une manière interne par Gocad pour construire les « Application Objects ».

- Un VSet est un ensemble de points sans connectivité entre eux.
- Un PLine est une collection de ILine (*Isolated Line*).
- Un ILine est un ensemble de segments interconnectés. Un ILine doit obligatoirement être créé dans un PLine et ne peut appartenir qu'à un et un seul PLine. Lorsqu'un PLine est détruit, tous les ILine qui le composent sont aussi détruits. De la même façon, lorsqu'un ILine est détruit tous ses segments le sont aussi.
- Un Tsurf peut être défini comme une collection de Tface.
- Un Tface est un ensemble continu de triangles interconnectés. Un Tface est aussi en charge de la gestion des éléments qui composent sa limite et contient à cet effet une liste de Tfaceborder. Tous les Tface doivent obligatoirement être créés dans un Tsurf et ne peuvent appartenir qu'à un et un seul Tsurf. Lorsqu'un Tsurf est détruit, tous les Tface qui le composent sont aussi détruits. De la même façon, lorsqu'un Tface est détruit tous ses triangles le sont aussi.
- Un Tsolid est une collection de Tvolume.

- Un TVolume est une collection de Tetra (tétraèdres) interconnectés qui définissent un volume dans l'espace. Un TVolume doit obligatoirement être créé dans un TSolid et ne peut appartenir qu'à un et un seul TSolid. Lorsqu'un TSolid est détruit, tous les TVolume qui le composent sont aussi détruits. De la même façon, lorsqu'un TVolume est détruit tous ses tétraèdres le sont aussi.

La structure de données de Gocad n'est pas une structure topologique proprement dite. En effet, les relations entre les différents objets ou « Application Objects » qui composent un modèle (macro topologie) ne sont pas stockés explicitement. Toutefois, Gocad implémente ce qu'on appelle une micro topologie c'est-à-dire que bien qu'il n'y ait pas de relations topologiques entre les différents objets spatiaux d'un modèle, chacun de ces objets est composé de sous-éléments qui eux sont représentés de façon topologique. Par exemple, il n'y a pas de relations topologiques entre deux solides (TSolid) adjacents d'un modèle. Par contre, un TSolid est composé de tétraèdres qui partagent des faces communes i.e. une seule face commune à deux tétraèdres est stockée. De la même façon, les segments communs entre les faces de tétraèdres sont stockés une seule fois.

Une fois définies les primitives du modèle Gocad et relations entre celles-ci, il est possible d'établir une correspondance avec les éléments du modèle GML de l'ISO. Le tableau suivant montre quelques-uns des principaux objets de la structure Gocad et leur correspondance dans le langage GML.

Objets Gocad	Élément GML
Node	gml:Point
VSet	gml:MultiPoint
Edge	gml:LineString (contenant seulement 2 nœuds), gml:LineStringSegment
ILine	gml:LineString, gml:Curve, gml:OrientableCurve
PLine	gml:CompositeCurve, gml:MultiCurve
Trgl	gml:Triangle, gml:Polygon
TFace	gml:Surface, gml:OrientableSurface
TSurf	gml:CompositeSurface, gml:MultiSurface
Tetra	gml:Solid
TVolume	gml:CompositeSolid

TSolid	gml:CompositeSolid, MultiSolid
--------	--------------------------------

Tableau 4 : Correspondance entre les objets Gocad et les éléments GML.

Il peut paraître étrange que le TVolume et le TSolid soient tous deux représentés en GML par un CompositeSolid même si le TSolid est une collection de TVolume. En fait, le TVolume doit être représenté par un CompositeSolid dont chacun des membres est un solide. De son côté, le TSolid peut être aussi représenté par un CompositeSolid mais dont chacun des membres est à son tour un CompositeSolid. Ceci est possible puisque les normes ISO 19107 et GML mentionnent qu'un Composite est un élément complexe qui contient une collection homogène de primitives interconnectées formant un tout qui peut être considéré comme une primitive. Ainsi, l'imbrication de Composites est permise. Toutefois, pour faire partie d'un Composite, chacun des membres doit être interconnecté avec ses voisins ce qui n'est pas le cas pour une Agrégation. Si un TSolide contient plusieurs TVolume qui ne sont pas interconnectés, une Agrégation de type MultiSolid doit être utilisée pour représenter notre TSolid. Le même raisonnement peut s'appliquer au niveau des PLine et des TSurf, ils peuvent être à la fois représentés par des composites ou des agrégations dépendamment si les ILine et TFace qui les composent sont interconnectés.

4.5 Développement de l'application serveur

Deegree est à la base une plate-forme de développement d'application web implémentant les principaux standards et spécifications de l'ISO et de l'OGC. Il offre entre autres une implémentation des interfaces de services web WMS, WFS et WCS. Dans sa version 2 utilisée pour le développement du WGFS, Deegree implémente en partie le schéma spatial ISO 19107. Deegree étant à la base destiné à gérer uniquement des données géométriques 2D, il contient seulement les classes de base du schéma spatial de l'ISO correspondant aux primitives géométriques 2D (le profil « *Simple Features* »). En effet, même s'il implémente le schéma spatial de l'ISO, Deegree est une application qui se limite en grande partie à la gestion des primitives spatiales définies dans le SFS de l'OGC. Cette limitation se justifie par le fait que la bibliothèque d'opérateurs spatiaux qu'il implémente (JTS) et les SGBD spatiaux qu'il supporte se limitent tous à cette spécification. D'ailleurs, les développeurs de Deegree ont dû implémenter un adaptateur JTS afin de pouvoir convertir les entités ISO 19107 à la structure du SFS implémentée dans la JTS. Le tableau suivant

montre les principales primitives géométriques implémentées par Deegree et leur correspondance avec les primitives de la JTS:

Deegree (ISO 19107)	JTS (SFS)
Point	Point
Multipoint	Multipoint
Curve	LineString
MultiCurve	MultiLineString
Surface	Polygon
MultiSurface	Multipolygon
MultiPrimitive	GeometryCollection

Tableau 5 : Les principales primitives géométriques implémentées par Deegree et leur correspondance avec les primitives de la Java Topology Suite (JTS).

Afin de rendre Deegree 3D-compatible, les classes du schéma spatial ISO 19107 nécessaires à la représentation d'objets 3D ont dû être implémentées. Le tableau suivant montre la liste des classes du schéma spatial de l'ISO que nous avons implémentées complètement ou en partie dans Deegree dans le cadre de ce projet de recherche.

Classes ISO 19107
GM_Shell
GM_Solid
GM_Solid_Boundary
GM_Composite_Solid
GM_Complex
GM_Composite_Surface

Tableau 6: Liste des classes du schéma spatial ISO 19107 implémentées dans Deegree dans le cadre de ce projet.

Étant donné l'introduction de ces classes, plusieurs modules de Deegree ont dû être modifiés pour s'adapter au 3D et à ces nouveaux éléments de structure. Entre autres, des méthodes pour permettre l'instanciation des nouveaux types d'objets spatiaux ont dû être implémentées. Pour ce faire, Deegree implémente une classe (GeometryFactory) dont le rôle est de créer des objets spatiaux en respect de la structure ISO. Cette classe a donc dû être bonifiée de plusieurs méthodes permettant de créer les objets ISO nouvellement

implémentés. D'autres méthodes comme par exemple *GetEnvelope* ont aussi dû être adaptées pour tenir compte de la troisième dimension. En effet, alors qu'une telle classe était anciennement destinée à retourner un rectangle englobant, celle-ci doit maintenant pouvoir renvoyer une boîte englobante lorsqu'elle est appelée dans le contexte d'un objet 3D. Le module permettant de convertir les entités ISO 19107 en GML et vice-versa a aussi dû être bonifié pour lui permettre de traiter les nouvelles classes d'objets. En annexe F sont présentés, plusieurs exemples d'implémentation de nouvelles classes introduites dans Deegree pour permettre la gestion des objets géométriques 3D.

Deegree prend également en charge l'implémentation des *datastores*. Pour permettre le stockage en XML dans une base de données relationnelle, nous avons modifié et complété le *datastore* (appelé *GenericSQLDatastore*) pour lui permettre de stocker nos modèles 3D dans le SGBD MySQL. Un pseudo index spatial a été implémenté à l'intérieur de ce *datastore* afin de permettre trois choses :

- La récupération des coordonnées de la boîte englobante des objets spatiaux à l'aide de la méthode *GetEnvelope* définie dans le schéma spatial de l'ISO.
- Le stockage de ces coordonnées (minX, minY, minZ, maxX, maxY, maxZ) dans la base de données lors des Insert et Update d'objets spatiaux.
- La vérification de ces coordonnées avant l'extraction d'objets spatiaux de la base de données lorsque la requête du client fait appel à l'opérateur spatial BBOX pour permettre l'extraction d'un sous-ensemble du modèle.

C'est donc par l'intermédiaire de ce pseudo index spatial qu'a été implémenté le premier opérateur spatial du WGFS soit le BBOX (*bounding box*). En annexe F, la méthode « *appendBBOXOperationAsSQL* » de la classe « *GenericSqlWhereBuilder* » contient l'implémentation de ce pseudo index spatial. Pour ce qui est des autres opérateurs spatiaux 3D qui pourront être implémentés ultérieurement, un premier filtrage grossier des données (tel que décrit au chapitre 3) devra être réalisé en comparant la géométrie d'entrée de la requête avec les coordonnées minX, minY, minZ, maxX, maxY, maxZ (boîte englobante)

des objets visés. Par exemple, si une requête demande l'extraction de tous les objets d'un modèle qui croisent (opérateur « crosses ») une surface dont la géométrie est spécifiée en entrée, l'algorithme devra d'abord comparer la géométrie de cette surface avec toutes les boîtes englobantes des objets du modèle. Une fois déterminées les quelques boîtes englobantes croisant la surface, le résultat final sera affiné en comparant la géométrie réelle des objets retenus avec celle de la surface afin de réellement calculer le croisement désiré.

4.6 Implémentation de la base de données

Une première stratégie de stockage de modèles géologiques 3D a été implémentée dans le cadre du développement du prototype du WGFS. Tel que décrit au chapitre précédant, cette stratégie consiste en le stockage d'un modèle 3D sous format XML/GML dans une table relationnelle du SGBD MySQL. Chacun des enregistrements de la table représente un objet spatial de la réalité (faille, unité géologique, etc.). Les champs de la table représentent les propriétés reliées à chacun de ces objets. Minimalelement, huit champs sont nécessaires pour la représentation d'un modèle. Un de ces champs contient la clé primaire permettant d'identifier de manière unique chacun des objets. Un autre champ de type texte contient la géométrie en format XML/GML. Si le SGBD utilisé était XML-compatible, un champ de type XMLType pourrait être utilisé pour le stockage de la géométrie. Six champs sont consacrés au stockage des coordonnées de deux coins opposés de la boîte englobant l'objet spatial modélisé soit les champs MinX, MinY, MinZ, MaxX, MaxY, MaxZ. Ces champs permettent à notre pseudo index de pouvoir extraire une partie du modèle lors de l'utilisation de l'opérateur spatial BBOX. Un nombre indéterminé de champs peuvent ensuite être insérés dans la table pour stocker d'autres propriétés descriptives venant ainsi relier de l'information à chacun des objets. Il est aussi possible de relier de l'information aux objets spatiaux par un joint avec une autre table par le biais de la clé primaire.

Pour ce qui est du modèle géologique lui-même, à l'aide du logiciel Gocad, du plug-in Gocad et du WGFS, le modèle de Porcupine-Destor du MRNF a été traduit en GML et stocké dans la base de données MySQL. De l'information descriptive a été extraite de la

base de données SIGEOM et rattachée au modèle. Quelques attributs descriptifs contenant de l'information sur la lithologie et la stratigraphie d'unités géologiques du territoire de la région de Porcupine-Destor ont été extraits d'un jeu de données spatiales obtenu de SIGEOM (sous format Shapefile) et rattachés aux solides de notre modèle 3D. La figure suivante montre un extrait de la base de données MySQL contenant le modèle géologique 3D de Porcupine-Destor ainsi que les propriétés descriptives s'y rattachant extraites de SIGEOM.

ID	NAME	stratigraphy	stra_desc	geometry	minx	miny	minz	maxx	maxy	maxz
12	QFP29_so	[arch]mdq	Archeen, Intrusions de la Mine Du...	<gml:CompositeSolid xm...	645...	5.37...	-4...	645...	5.37...	318...
5	I3A13_so	[arch]3	Archeen	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	327...
15	SSW_so	[arch]mn	Archeen, Formation de Mont-Brun	<gml:CompositeSolid xm...	643...	5.37...	-6...	645...	5.37...	307...
14	SSE_so	[arch]mn	Archeen, Formation de Mont-Brun	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	326...
8	centreSE_so	[arch]ln3	Archeen, Formation de Lanaudiere	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	333...
7	centreSS_so	[arch]ln3	Archeen, Formation de Lanaudiere	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	331...
3	I3A16_so	[arch]3	Archeen	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	334...
9	centreNE_so	[arch]ln3	Archeen, Formation de Lanaudiere	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	334...
4	I3A15_so	[arch]3	Archeen	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	329...
6	centreSW_so	[arch]ki	Archeen, Groupe de Kinojevis	<gml:CompositeSolid xm...	643...	5.37...	-6...	644...	5.37...	313...
2	I3A17_so	[arch]3	Archeen	<gml:CompositeSolid xm...	644...	5.37...	-6...	645...	5.37...	319...
11	QFP26_so	[arch]mdq	Archeen, Intrusions de la Mine Du...	<gml:CompositeSolid xm...	644...	5.37...	-6...	645...	5.37...	319...
1	LA36_so	[arch]ln4	Archeen formation de Lanaudiere	<gml:CompositeSolid xm...	644...	5.37...	-6...	646...	5.37...	348...
16	SW_so	[arch]mn	Archeen, Formation de Mont-Brun	<gml:CompositeSolid xm...	643...	5.37...	-6...	645...	5.37...	310...
13	QFP30_so	[arch]mdq	Archeen, Intrusions de la Mine Du...	<gml:CompositeSolid xm...	645...	5.37...	-6...	645...	5.37...	316...
10	nordES	[arch]ln1	Archeen, Formation de Lanaudiere	<gml:CompositeSolid xm...	645...	5.37...	-6...	646...	5.37...	320...

Figure 11: Extrait de la base de données MySQL montrant la table et les données stockées du modèle géologique 3D de Pocupine-Destor.

4.7 Le WGFS en action

Les sections qui suivent présentent en détail le fonctionnement du WGFS. L'utilisateur accède d'abord au système par la fenêtre de requête WFS rendue disponible dans Gocad lors de l'installation du plug-in WGFS (voir figure 12).

WFS query Dialog

Input

Server URL:

BBOX (Point 1): x: y: z:

BBOX (Point 2): x: y: z:

getFeature equal

Insert plus petit

Delete plus grand

Feature:

```
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>app:id</ogc:PropertyName>
<ogc:Literal>1</ogc:Literal>
</ogc:PropertyIsEqualTo>
```

Figure 12 : Extrait de la boîte de dialogue WFS du plug-in Gocad.

4.7.1 Fonctionnalité "*Voir la liste des modèles 3D disponibles*"

L'obtention de la liste des modèles rendus disponibles par le WGFS ou quelconque serveur WFS s'effectue par l'envoi d'une requête *GetCapabilities*. La figure suivante schématise la transaction entre le client et le serveur lors d'une telle requête.

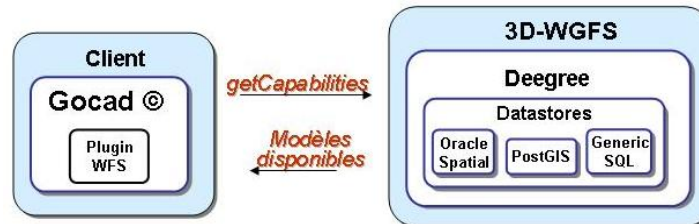


Figure 13 : Schéma de la transaction entre le client et le serveur lors d'une requête *GetCapabilities*.

La première chose à faire pour se brancher au serveur WGFS à l'aide du client Gocad est d'entrer l'adresse URL du serveur dans la fenêtre et de demander au système la liste des modèles disponibles dans la base de données. Une fonction C++ du plug-in Gocad appelle alors par l'intermédiaire de JNI, une classe Java (provenant du module Java côté client en charge de la conversion des données) qui implémente une série de méthodes permettant de créer et de lancer des requêtes WFS. Une requête *GetCapabilities* est alors générée et la bibliothèque Java HTTPClient est utilisée pour l'envoyer au serveur. Celui-ci construit la liste des jeux de données disponibles et renvoie sa réponse au client en format XML. Ce document XML est parcouru par le module Java et la liste des modèles disponibles est récupérée et renvoyée comme valeur de retour à la fonction du plug-in Gocad qui affiche le résultat à l'utilisateur dans une liste déroulante comme le montre la figure suivante :

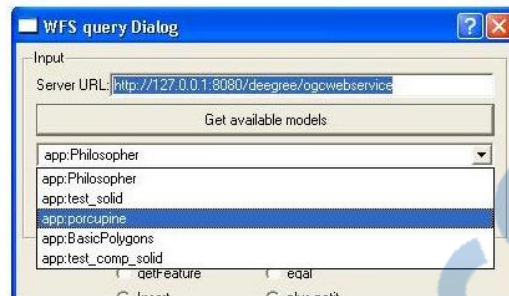


Figure 14 : Liste déroulante de la boîte de dialogue du plug-in Gocad montrant la liste des jeux de données disponibles sur le serveur.

4.7.2 Fonctionnalité "Importer un modèle"

L'importation d'un modèle à partir de la base de données est effectuée par l'entremise d'une requête *GetFeature*. La figure suivante schématise la transaction entre client et serveur lors d'une requête *GetFeature* :

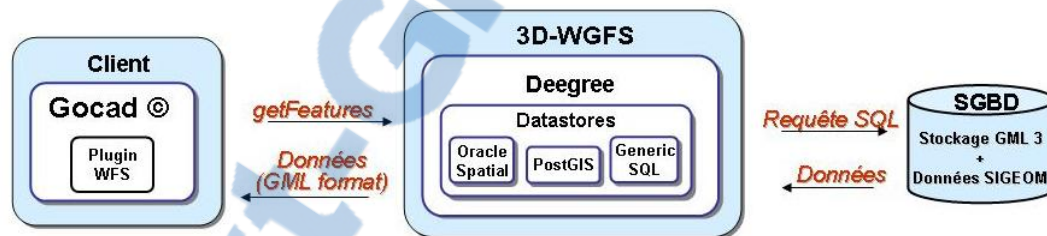


Figure 15 : Schématisation de la transaction entre client et serveur lors d'une requête *GetFeature*.

La fenêtre WFS du plug-in Gocad permet à l'utilisateur de construire facilement une telle requête. Le modèle désiré peut être sélectionné dans la liste déroulante obtenue précédemment par la requête *GetCapabilities*. Une requête *GetFeature* contient une section appelée filtre, qui permet de spécifier si l'on désire l'importation complète d'un modèle ou seulement une partie de celui-ci. Si aucun filtre n'est spécifié, le modèle complet sera importé. Un filtre contient une combinaison de propriétés descriptives, d'opérateurs logiques et arithmétiques et de valeurs (ex : lithologie = "V3B" ET stratigraphie = "[arch]3") pour permettre d'importer seulement les entités possédant certaines propriétés. Le filtre peut aussi faire appel aux opérateurs spatiaux rendus disponibles par le serveur. Dans le cadre du prototype du WGFS, un premier opérateur, le BBOX, a été implémenté.

La fenêtre WFS de Gocad permet à l'utilisateur d'entrer les coordonnées x,y,z de deux coins opposés de la boîte englobante de la zone désirée. Lorsque l'utilisateur envoie sa requête, l'URL, le filtre et les coordonnées du BBOX sont passés en paramètre à la classe Java du module de traduction de données par l'intermédiaire de JNI. La requête WFS *GetFeature* est alors construite à partir des paramètres reçus et envoyée au serveur. La figure suivante montre un exemple de requête *GetFeature*:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <wfs:GetFeature version="1.1.0" outputFormat="text/xml; subtype=gml/3.1.1" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc">
- <wfs:Query xmlns:app="http://www.deegree.org/app" typeName="app:porcupine">
- <ogc:Filter>
  - <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>app:name</ogc:PropertyName>
    <ogc:Literal>nordES</ogc:Literal>
  </ogc:PropertyIsEqualTo>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

Figure 16 : Exemple de requête WFS de type *GetFeature*.

Cette requête décrit l'importation d'une partie du modèle appelé Porcupine. Son filtre va restreindre l'importation aux entités géométriques dont la propriété « *name* » est égale à « nordES ». Ce filtre pourrait aussi faire appel à l'opérateur spatial BBOX comme dans l'exemple suivant :

```
- <ogc:Filter>
- <ogc:BBOX>
  <PropertyName>app:GEOM</PropertyName>
- <gml:Box>
  - <gml:coord>
    <gml:X>645778</gml:X>
    <gml:Y>5370460</gml:Y>
    <gml:Z>-610</gml:Z>
  </gml:coord>
  - <gml:coord>
    <gml:X>646263</gml:X>
    <gml:Y>5370780</gml:Y>
    <gml:Z>343</gml:Z>
  </gml:coord>
</gml:Box>
</ogc:BBOX>
</ogc:Filter>
```

Figure 17 : Exemple d'un filtre OGC montrant les paramètres de l'opérateur BBOX.

Une fois la requête rendue au serveur, celui-ci analyse son contenu, trouve le *datastore* approprié pour se connecter à la base de données où est localisé le modèle demandé et construit la requête SQL en tenant compte des spécificités propres au SGBD hébergeant le modèle ainsi qu'aux contraintes spécifiées dans le filtre de la requête. Cette requête SQL est exécutée au niveau de la base de données par l'intermédiaire du protocole Java Database Connectivity (JDBC). Les données géométriques format GML et les propriétés descriptives reliées à celles-ci sont récupérées et envoyées à l'application serveur. Afin de pouvoir manipuler les données géométriques, l'application serveur traduit les données du format GML dans sa structure standard. Des instances d'objets du schéma spatial de l'ISO sont alors créées en mémoire. Si des traitements nécessitant l'utilisation d'un ou plusieurs opérateurs spatiaux sont mentionnés dans le filtre, c'est à ce moment que le serveur procède au calcul. Une fois les calculs terminés, l'application serveur se charge d'écrire la requête de retour devant être renvoyée au client. Pour ce faire, l'application serveur traduit les objets ISO 19107 résultants en GML et renvoie le tout vers le client. La figure suivante montre un extrait de document XML montrant le modèle 3D de Porcupine-Destor et ses propriétés descriptives sous format GML tel que formaté par le WGFS :

```

- <app:porcupine gml:id="porcupine_e03f4017010edda5dde000000193a00">
- <gml:boundedBy>
- <gml:Envelope>
  <gml:pos srsDimension="3">645954.25 5370906.0 -600.0</gml:pos>
  <gml:pos srsDimension="3">646313.0625 5371254.0 320.560455</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<app:id>10</app:id>
<app:name>nordES</app:name>
<app:lithology>V1B</app:lithology>
<app:litho_desc>Rhyolite</app:litho_desc>
<app:stratigraphy>[arch]ln1</app:stratigraphy>
<app:stra_desc>Archeen, Formation de Lanaudiere</app:stra_desc>
- <app:GEOM>
- <gml:CompositeSolid xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:solidMember>
- <gml:Solid xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:exterior>
- <gml:CompositeSurface xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:surfaceMember>
- <gml:OrientableSurface orientation="+">
- <gml:baseSurface>
- <gml:Polygon xmlns:gml="http://www.opengis.net/gml">
- <gml:outerBoundaryIs>
- <gml:LinearRing>
  <gml:coordinates cs="," decimal="." ts="">646069.8125,5371092.5,-270.100555
  646017.875,5371137.0,-205.76387 646096.1875,5371089.0,-174.089264
  646069.8125,5371092.5,-270.100555</gml:coordinates>
</gml:LinearRing>
</gml:outerBoundaryIs>
</gml:Polygon>
</gml:baseSurface>
</gml:OrientableSurface>
</gml:surfaceMember>
+ <gml:surfaceMember>
+ <gml:surfaceMember>
+ <gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</gml:solidMember>
+ <gml:solidMember>
+ <gml:solidMember>

```

Figure 18 : Extrait de document XML montrant le modèle 3D de Porcupine-Destor et ses propriétés descriptives sous format GML tel que généré par le WGFS.

Encore une fois, une classe du module client Java reçoit la réponse qu'il parcourt à l'aide du parseur XML. Les coordonnées des objets spatiaux du document XML sont alors extraites et concaténées ensemble d'une manière semblable au format WKT afin de simplifier au maximum l'information qui est envoyée au client Gocad. Les propriétés descriptives de chacun des objets sont aussi stockées côté client dans un fichier texte afin de pouvoir être récupérées plus tard et affichées dans Gocad à la demande de l'utilisateur. Une fois transférés dans Gocad par l'intermédiaire de JNI, les objets spatiaux sous format WKT 3D sont traduits par des fonctions programmées au niveau du plug-in en objets Gocad et affichés à l'utilisateur. La figure suivante montre une partie du modèle de Porcupine-Destor tel

qu'affiché dans Gocad après l'importation à partir de la base de données. Dans la légende à gauche on peut voir la liste des solides importés composant le modèle.

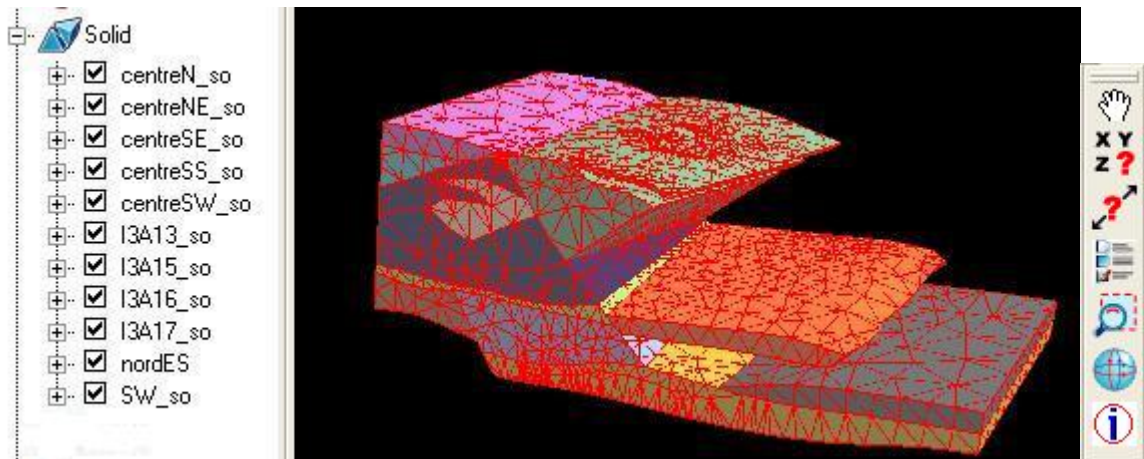


Figure 19 : Partie du modèle de Porcupine Destor tel qu'affiché dans Gocad après l'importation à partir de la base de données.

4.7.3 Fonctionnalité "*Requêter les propriétés descriptives d'un modèle*"

Une fois un modèle importé dans Gocad par l'intermédiaire du WGFS, les propriétés descriptives reliées à chaque objet spatial peuvent être requêtées. Pour ce faire, un outil d'identification (le symbole **i** dans la figure précédente) ajouté dans Gocad lors de l'installation de notre Plugin permet de faire afficher d'une manière intégrée, les propriétés d'un objet lors d'un clic de souris sur celui-ci comme le permettent les logiciels SIG. La figure suivante montre les propriétés d'un solide affichées dans l'interface graphique de Gocad suite à un clic de souris sur un solide.



Figure 20 : Exemple de fenêtre du plug-in Gocad montrant les propriétés descriptives d'un solide.

Ces propriétés sont récupérées dans le fichier texte contenant l'ensemble des propriétés du modèle créé lors de l'importation de celui-ci à partir du WGFS et sont accompagnées de l'identifiant unique de l'objet spatial auquel elles se rattachent. L'identifiant unique de l'objet localisé aux coordonnées du clic est récupéré par une fonction du plug-in qui vient par la suite parcourir le fichier texte pour récupérer l'information et l'afficher à l'utilisateur.

4.7.4 Fonctionnalité "*Exporter un modèle vers la base de données*"

Le WGFS permet aussi de mettre à jour à partir du client Gocad, un modèle 3D stocké dans la base de données afin de permettre aux autres usagers du système de pouvoir profiter de données actualisées. Ceci peut être effectué par le biais d'une requête WFS de type *Transaction* qui permet de faire des insertions (*Insert*), des actualisations (*Update*) et des destructions (*Delete*). La figure suivante schématise la transaction entre client et serveur lors d'une requête WFS *Transaction* de type *Insert*, *Update* ou *Delete* :

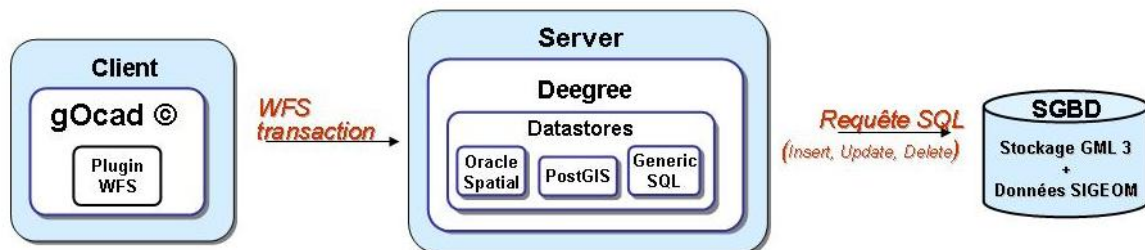



Figure 21 : Schématisation de la transaction entre client et serveur lors d'une requête WFS *Transaction* de type *Insert*, *Update* ou *Delete*.

La fenêtre de requête WFS du plug-in Gocad permet à l'utilisateur de construire facilement une requête *Transaction*. Pour ce faire, l'utilisateur doit sélectionner le type de *Transaction* qu'il veut effectuer après avoir sélectionné le modèle désiré dans la liste préalablement construite par une requête *GetCapabilities*. Lorsqu'une insertion est demandée par l'utilisateur, une requête WFS *GetFeatureType* est envoyée au serveur afin de connaître la structure des propriétés du modèle. Une requête *GetFeatureType* permet de connaître la liste des propriétés descriptives d'un modèle ainsi que le type de chacun d'eux (texte, entier, date, géométrie,...). Encore une fois, c'est une fonction du plug-in Gocad qui par l'intermédiaire de JNI envoie les paramètres nécessaires (URL du serveur et modèle désiré) au module client Java qui construit la requête *GetFeatureType* et l'envoie au WGFS. L'application côté serveur récupère l'information sur la structure de la base de données stockant le modèle et renvoie le tout au client sous format XML. La liste des propriétés est récupérée et envoyée vers Gocad qui l'affiche à l'utilisateur dans la fenêtre de requête WFS. La figure suivante montre la fenêtre de requête WFS du plug-in Gocad affichant les propriétés descriptives du modèle après une requête *GetFeatureType* :



```

<app:porcupine gml:id="porcupine_1">
<app:id?</app:id>
<app:name?</app:name>
<app:lithology?</app:lithology>
<app:litho_desc?</app:litho_desc>
<app:stratigraphy?</app:stratigraphy>
<app:stra_desc?</app:stra_desc>
<app:GEOM>LA GEOMETRIE</app:GEOM>
</app:porcupine>

```

Figure 22 : Exemple de fenêtre du plug-in Gocad montrant la liste des propriétés descriptives d'un modèle suite à une requête *GetFeatureType*.

Par la suite, l'utilisateur peut entrer les valeurs à insérer dans la base de données vis-à-vis chacune des propriétés. Une fois complété, l'utilisateur spécifie le nom de l'objet spatial qu'il désire stocker dans la base de données et envoie la requête. Le plug-in Gocad se charge de récupérer la liste des propriétés descriptives et leurs valeurs correspondantes pour l'envoyer au traducteur de données Java par l'intermédiaire de JNI. La norme WFS suggère que la géométrie décrite dans cette requête *Transaction* soit formatée en GML. Le plug-in

s'occupe donc de récupérer et d'envoyer les coordonnées composant l'objet spatial spécifié par l'utilisateur (en format WKT-3D) et envoie le tout vers l'application Java. Celle-ci se charge de construire la requête WFS *Transaction* et l'envoie au serveur. Une fois côté serveur, le WGFS identifie le *datastore* approprié et construit une requête SQL permettant l'insertion de l'information géométrique et descriptive dans la base de données MySQL. Cette requête SQL permettra aussi de stocker les coordonnées de la boîte englobante de l'objet spatial à stocker qui sont obtenues par le *datastore* par un appel à la méthode standard *getEnvelope*. Une fois l'insertion complétée, le serveur construit un document XML contenant une confirmation ou infirmation de la réussite de la transaction qu'il envoie au client. Cette réponse est récupérée par le plug-in Gocad et affichée à l'utilisateur. Dans le cas d'une transaction WFS de type *Delete*, on élimine de la base de données l'entité spécifiée dans la requête de l'utilisateur au lieu de l'insérer. Dans le cas d'un *Update*, il suffit d'effectuer un *Delete* pour éliminer l'élément à actualiser pour ensuite réinsérer l'élément à jour avec un *Insert*.

4.8 Évaluation du WGFS

Lors de l'élaboration du concept de SIG 3D, nous avons déterminé qu'un tel système devrait selon nous permettre la construction, la visualisation et la manipulation des modèles 3D (impliquant la gestion d'une ou plusieurs structures géométriques 3D), leur gestion et leur stockage au sein d'une base de données (incluant la gestion des propriétés descriptives) ainsi que l'analyse spatiale métrique et topologique. Nous avons aussi mentionné qu'un tel système devrait être interopérable, devrait adopter les standards et favoriser le partage des données spatiales 3D. Nous évaluerons donc l'application WGFS sur ces aspects.

4.8.1 Description des tests réalisés

L'objectif général des tests effectués était de valider à l'aide du prototype WGFS, la faisabilité (en termes d'effort et de moyens requis), la validité (au niveau technique et informatique), la performance, l'utilité, l'accessibilité, la complétude, la portabilité ainsi que l'extensibilité de la solution proposée.

Différents tests ont permis de quantifier et qualifier certains aspects du prototype comme la validité et l'intégrité des données servies par le serveur WGFS, l'intégrité des données qu'il

stocke ainsi que la rapidité des réponses et des traitements. Une présentation du système à des géologues du MRNF a aussi été réalisée afin de recueillir leur opinion et voir leur intérêt. La période de tests s'est déroulée en deux phases. La première fut effectuée dans les locaux du Centre de Recherche en Géomatique de l'Université Laval soit l'endroit même où a été développé le prototype. La deuxième phase de tests, réalisée en collaboration avec Vincent Thomas un étudiant du Département des sciences géomatiques, consistait en l'implantation du système dans les locaux du MRNF en Abitibi et sa présentation à des géologues (Thomas, 2007). Dans les deux cas, les tests consistaient en l'exécution de chacune des fonctionnalités du système, en l'analyse des résultats et en la comparaison des possibilités du système avec les tâches théoriques qu'on lui a attribuées. Chacun des types de requêtes WFS que permet le WGFS (GetCapabilities, GetFeatures, Transaction de type Insert, Delete) a donc été exécuté à partir du client Gocad. Dans le cas des requêtes *GetFeature*, plusieurs tentatives étaient lancées pour vérifier différentes situations :

- importation du modèle complet
- importation d'un seul solide
- importation de quelques solides à partir d'une requête basée sur les propriétés descriptives des entités géométriques
- importation de quelques solides basée sur l'utilisation de l'opérateur spatial BBOX

Dans le cas des requêtes *Transaction* de type *Insert*, on vérifiait que les solides exportés ainsi que les propriétés descriptives qui leurs sont attribuées étaient stockés adéquatement dans la base de données. On procédait aussi à des comparaisons de la géométrie entre solides avant et après leur exportation dans la base de données. Un solide Gocad ayant séjourné dans la base de données devrait avoir une géométrie identique à sa copie originale n'ayant jamais été transformée en GML pour le stockage dans la base de données. Dans le cas de l'importation de solides à partir d'une requête basée sur les propriétés descriptives, il était non seulement intéressant de mesurer les temps de réponse, mais aussi la validité de la requête. En effet, les solides retournés devaient bel et bien posséder les propriétés demandées dans le filtre de la requête pour être valides. Même chose au niveau de l'importation de quelques solides basés sur l'utilisation de l'opérateur spatial BBOX. Dans ce cas, les solides retournés devaient être complètement inclus dans la boîte englobante

spécifiée par l'utilisateur par l'intermédiaire du client Gocad. Tous les tests ont été réalisés sur le modèle géologique 3D de Porcupine-Destor provenant du MRNF (présenté à la figure 19).

4.8.2 Résultats

En premier lieu, la présence au sein du WGFS des différentes composantes théoriques de la définition de notre SIG 3D a été validée afin de vérifier sa complétude. Les tâches ou composantes théoriques ainsi que leur présence ou non au sein du WGFS sont résumées dans le tableau suivant :

Tâche\composante	Présence au niveau du WGFS
Construction, manipulation, visualisation d'un modèle 3D	Oui
Gestion et stockage des modèles 3D au niveau d'une base de données	Oui
Gestion et requêtage des propriétés descriptives associées au modèle 3D	Oui
Analyse spatiale métrique 3D	Oui
Analyse spatiale topologique 3D	Non
Diffusion des modèles 3D	Oui
Interopérabilité du système	Oui
Adoption des standards	Oui

Tableau 7 : Validation du prototype WGFS par rapport à la définition théorique du SIG 3D

D'une manière générale, on peut voir que le prototype du WGFS offre la quasi-totalité des fonctionnalités présentées dans la définition théorique de notre SIG 3D. En effet, le client Gocad permet la modélisation, la visualisation et la manipulation des modèles 3D. Ces modèles reposent sur une structure géométrique 3D permettant la gestion explicite des objets volumiques. Les propriétés descriptives du modèle peuvent aussi être requêtées. Le serveur et la base de données du WGFS permettent la gestion, l'analyse et le stockage des modèles alors que le service Web (WFS 3D) permet leur diffusion. Le système repose aussi sur des standards tels qu'ISO 19107, GML et WFS ce qui lui permet d'être interopérable. Pour l'instant, bien que le WGFS repose sur une structure lui permettant éventuellement d'offrir des capacités d'analyse spatiale topologique 3D, aucun opérateur topologique 3D n'a été implémenté dans le cadre de ce projet.

Au niveau plus technique, le prototype obtient une note parfaite en ce qui a trait à la validité des données lors de l'importation du modèle. Cela signifie que les entités qu'il retourne au client sont toujours adéquates dans le sens où elles correspondent aux critères spécifiés dans la requête du client (tant au niveau des propriétés descriptives que spatiales). Par exemple, si l'on demande d'importer uniquement les solides dont la propriété lithologie = 'V1B', l'utilisateur reçoit bel et bien uniquement les solides qualifiés ainsi dans la base de données. Même chose au niveau de l'opérateur spatial BBOX, uniquement les solides complètement inclus dans la boîte englobante spécifiée sont importés. Pour ce qui est des comparaisons entre la géométrie des solides en format Gocad avant et après le processus de transformation lors du stockage dans la base de données, les résultats sont concluants. En effet, aucune différence au niveau de la géométrie n'est observée ce qui permet de confirmer le maintien de l'intégrité des données lors des différentes conversions (structure Gocad-GML-ISO 19107).

Pour ce qui est de la performance au niveau des temps de traitement des requêtes, des tests plus poussés ont été réalisés dans les locaux du MRNF à Val d'Or. Le client Gocad fut installé sur un ordinateur différent du serveur WFGS et ce dernier fut installé sur la même machine que la base de données MySQL contenant le modèle. Le tableau suivant présente les caractéristiques des machines cliente et serveur utilisées pour ces tests :

Client	Serveur
Windows XP PRO Service Pack 2	Windows XP PRO Service Pack 2
Processeurs x86 Family 15 Model 4 Genuine Intel de 3.192 GHZ	Processeur Intel Pentium 4 de 1.6 GHZ
Mémoire de 3 000 Mo de RAM	Mémoire de 256 Mo de RAM

Tableau 8 : Sommaire des caractéristiques des machines cliente et serveur utilisées lors des tests du WFGS.

Les temps de traitement de chacun des seize solides du modèle ont été enregistrés lors de leur importation dans Gocad (requête GetFeature) à partir de la base de données. Le tableau suivant montre les temps enregistrés pour l'importation de chacun des solides du modèle :

Id	Nom solide	Temps (s)	Nombre de tétraèdres	Temps/ tétraèdre
1	LA36_so	460	1 503	0.306
2	I3A17_so	175	862	0.203
3	I3A16_so	227	974	0.233
4	I3A15_so	92	535	0.172
5	I3A13_so	45	314	0.143
6	centreSW_so	54	343	0.157
7	centreSS_so	347	1 300	0.267
8	centreSE_so	28	267	0.105
9	centreNE_so	727	2 045	0.356
10	nordES	24	263	0.091
11	QFP26_so	96	705	0.136
12	QFP29_so	29	310	0.094
13	QFP30_so	44	412	0.107
14	SSE_so	47	422	0.111
15	SSW_so	420	1 664	0.252
16	SW_so	520	1 876	0.277

Tableau 9 : Temps d'importation dans Gocad des différents solides du modèle Porcupine.

Ces résultats montrent des temps d'importation passant de 24 secondes dans le cas du plus petit solide (263 tétraèdres) à plus de 12 minutes pour le solide contenant le plus grand nombre de tétraèdres (2 045 tétraèdres). On peut voir que les temps d'importation des solides augmentent de façon exponentielle en fonction du nombre de tétraèdres. À la lumière de ces résultats, il paraît évident que les temps d'importation sont pour le moment trop élevés et des solutions pour optimiser les traitements devraient être envisagées. Toutefois, des tests plus poussés devraient aussi être réalisés avec une machine serveur plus performante afin de voir si les temps d'importation peuvent diminuer substantiellement. D'ici là, on peut quand même penser qu'une des principales raisons de la lenteur des transactions provient de la grande complexité des objets spatiaux qui une fois convertis en XML génèrent des documents de grande taille devant être parcourus par un parseur XML et devant voyager sur un réseau comme le Web. À titre indicatif, le plus petit solide du modèle génère un document XML contenant plus de 25 000 lignes.

Au niveau de l'accessibilité du système, celui-ci répond aussi aux attentes en permettant de réaliser facilement les différentes tâches, et ce, en un nombre d'opérations très raisonnable.

Par exemple, l'exécution d'une tâche comme l'importation des données d'un modèle en entier peut être réalisée en aussi peu que trois clics de souris une fois la fenêtre WFS affichée à partir d'un menu de Gocad. L'utilisation d'un opérateur spatial comme le BBOX pour l'importation d'une portion spécifique du modèle peut aussi être réalisée en trois clics de souris et l'entrée de six valeurs au clavier (les coordonnées x,y,z minimums et maximums). La visualisation des propriétés descriptives d'un solide peut être réalisée en deux clics de souris (sélection du bouton dans la barre d'outils et clic sur le solide). L'importation de certains objets spatiaux spécifiques basés sur des propriétés descriptives ou l'exportation du modèle vers la base de données peuvent être réalisées en quatre, cinq ou six clics en plus de l'entrée d'un nombre variable de valeurs au clavier (dépendamment du nombre de propriétés descriptives entrant dans la composition de la requête ou devant être stockées dans la base de données).

Au niveau de l'utilité du système, les géologues auxquels le système a été présenté ont démontré beaucoup d'enthousiasme et d'intérêt. Leur principale préoccupation se situait au niveau des temps d'importation des données pour l'instant un peu trop élevés. Il était toutefois très clair, suite à la démonstration des capacités du système, qu'une telle application leur serait d'une grande utilité et qu'il nous encourageait à aller de l'avant avec le projet.

L'utilisation de la technologie Java pour le développement du système a permis de le rendre portable et l'architecture flexible de Degree comme l'adoption de standards et d'un concept comme le *datastore* rendent le système facilement extensible.

Finalement, à la lumière de ces résultats, nous pensons que le développement du prototype WGFS a permis sans aucun doute de remplir son objectif soit de démontrer la faisabilité de l'architecture de SIG 3D proposée.

4.9 Limites du système et travaux futurs

Pour l'instant, le WGFS est un prototype développé avec l'objectif de vérifier la faisabilité et l'utilité d'une telle architecture de solution SIG 3D. Toutefois, bien qu'il ne soit pas prêt à être implanté en production, le prototype du WGFS contient une base solide sur laquelle on pourrait compter pour construire un système plus complet destiné à être utilisé en

situation réelle. En effet, le système repose sur un ensemble de composantes et applications robustes qui sont utilisées un peu partout dans le monde dans le cadre de divers projets. Afin de compléter le WGFS, plusieurs composantes du système devraient être créées ou complétées :

- Comme nous l'avons déjà mentionné, il n'existe pas actuellement de bibliothèque standard et complète qui soit disponible contenant l'implémentation d'algorithmes d'opérateurs spatiaux 3D métriques et topologiques semblable à celles existantes dans les SIG et SGBD 2D. Il existe présentement un besoin grandissant pour une telle bibliothèque et ce manque représente un frein au développement de système géomatique 3D. Le WGFS présente l'infrastructure pour permettre d'accueillir une telle bibliothèque, mais n'implémente présentement que l'opérateur BBOX.
- Un des objectifs du prototype WGFS était de prouver qu'on pouvait bel et bien stocker et requêter des modèles 3D à partir du client Gocad, c'est-à-dire des modèles composés de primitives géométriques volumiques telles que des solides. Pour cette raison, un effort particulier a été porté au développement des classes et outils de conversion permettant de requêter, représenter, convertir et stocker les primitives de type solide. Nous avons ainsi délaissé le traitement de données ponctuelles, linéaires et surfaciques. Rappelons que la gestion et la manipulation des éléments volumiques représentaient le défi principal de ce travail, le distinguant entre autres avec les travaux de Apel (qui lui s'était concentré sur les points, les lignes et les surfaces). Afin de rendre le système complètement fonctionnel, il faudrait compléter l'implémentation de certaines classes et modules applicatifs côté client et serveur afin de traiter ces types de primitive.
- Pour l'instant, seuls les modèles 3D non topologiques sont supportés par le système. Toutefois, le WGFS repose sur une architecture standard et ouverte qui pourrait très bien supporter la topologie. Pour ce faire, le schéma spatial ISO 19107 qu'implémente le WGFS possède une section destinée spécialement à la topologie. Afin de permettre au WGFS de supporter les modèles topologiques 3D, il suffirait d'implémenter les classes d'entités géométriques topologiques de l'ISO (TP_Solid, TP_Surface, ...). Il faudrait aussi adapter les modules du WGFS en charge de la conversion géométrique

des modèles entre la structure ISO 19107 et le format GML 3 et vice-versa. Un *datastore* devrait aussi être développé pour chaque structure de stockage topologique 3D différente. Afin de profiter pleinement des avantages que confère la topologie, des opérateurs spatiaux topologiques 3D devraient idéalement être implémentés au sein de la bibliothèque d'opérateurs spatiaux du WGFS. Ces opérateurs topologiques seraient appelés plutôt que ceux de la bibliothèque d'opérateurs spatiaux d'une manière transparente lorsque la requête du client porterait sur un modèle topologique 3D.

- Afin d'optimiser l'extraction des objets spatiaux de la base de données, un index spatial 3D devrait être implémenté. Le pseudo-index spatial implémenté dans le prototype nous permet d'extraire une partie d'un modèle à partir d'une paire de coordonnées (x,y,z). Toutefois, ce pseudo-index n'est pas aussi optimal que les index spatiaux normalement disponibles dans les principaux SGBD spatiaux tels que le Quadtree et le R-Tree.
- Le prototype du WGFS permet à la suite d'actualisation d'un modèle à l'aide d'un client comme Gocad de venir exporter le modèle à jour dans la base de données par le biais d'une transaction WFS de type *Insert*. Toutefois, l'option de « *lock* » des données décrite dans la spécification WFS permettant d'éviter la destruction mutuelle de l'information durant les transactions effectuées dans un environnement multi-usager n'est pas implémentée. Tel que mentionné dans la spécification, ce *lock* devrait normalement être implémenté au niveau de la base de données. En effet, la plupart des SGBD possèdent des fonctionnalités intégrées permettant de gérer d'une manière sécuritaire la mise à jour des données. Afin de sécuriser l'actualisation des modèles avec le WGFS, il serait important d'implémenter le mécanisme de *lock* en respectant d'une part, les recommandations de la spécification WFS et d'autre part en ayant recours aux fonctionnalités internes du SGBD afin d'éviter de devoir redévelopper un tel mécanisme de sécurité.
- La traduction en GML d'un modèle 3D tel que le modèle géologique de Porcupine-Destor peut rapidement générer un document très volumineux. En effet, lorsqu'un territoire relativement grand doit être couvert et que l'on a besoin d'un minimum de précision, un tel modèle 3D peut facilement contenir des dizaines de solides totalisant

plusieurs milliers de tétraèdres ce qui peut représenter en GML des dizaines de milliers de lignes de texte. Le transfert de ces documents à travers le Web peut donc devenir problématique. Ce problème de transfert de documents XML volumineux n'est pas propre au domaine de la modélisation 3D et des recherches sont présentement en cours pour en améliorer les résultats. Afin d'améliorer les performances du WGFS lors de la gestion de modèles volumineux, différentes avenues pourraient être expérimentées comme le XML binaire qui permet de réduire substantiellement le volume de données ou la compression des documents XML par l'utilisation d'algorithmes pour leur archivage et leur désarchivage lors du transfert sur le web. À noter que l'utilisation du XML binaire diminue grandement l'intérêt premier d'un langage comme le XML puisqu'un tel traitement ferait perdre aux documents leur portabilité sur différentes plateformes.

- Tel que présenté en introduction, le MRNF aimerait pouvoir stocker au sein d'une base de données, particulièrement SIGEOM, leurs modèles géologiques 3D créés avec Gocad. Le WGFS étant conçu justement pour le stockage de ce type de modèles, il pourrait s'avérer une solution intéressante pour le stockage et la distribution des modèles géologiques 3D au sein même de SIGEOM. Bien que SIGEOM soit un système complexe qui pourrait difficilement être modifié à moindre coût, la gestion des modèles 3D dans SIGEOM serait envisageable. Tel que présenté précédemment dans ce chapitre, un modèle 3D peut être stocké de différente façon au sein d'une base de données. SIGEOM étant implémenté au sein du SGBD Oracle, une table pourrait être ajoutée dans la base de données afin de pouvoir stocker un modèle 3D. L'information géologique contenue dans la base de données SIGEOM pourrait être associée aux objets spatiaux du modèle 3D afin de pouvoir être requêtée conjointement à l'aide d'un client WGFS comme Gocad sans devoir modifier la structure de SIGEOM. En effet, il serait possible par un simple joint de tables de joindre de l'information de SIGEOM à un modèle 3D à condition d'utiliser pour identifier uniquement chacun des objets spatiaux du modèle, la même clé primaire que dans SIGEOM. Par exemple, une table de SIGEOM contenant de l'information portant sur des unités géologiques pourrait être jointe à un modèle 3D représentant ces mêmes unités géologiques lorsque correspond

l'identifiant unique de chacune des tables. Le schéma suivant montre un exemple simplifié d'une telle intégration entre SIGEOM et le WGFS :

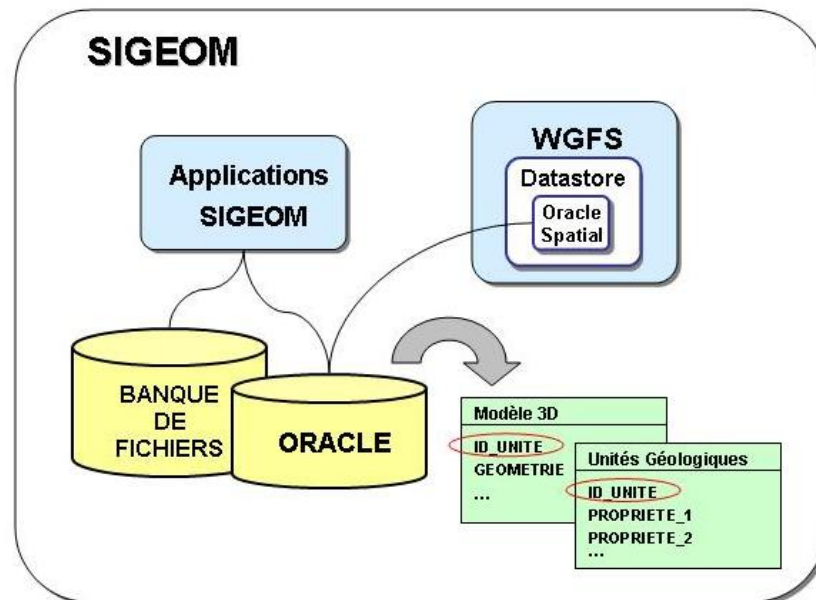


Figure 23 : Schéma simplifié d'intégration entre SIGEOM et le WGFS.

De cette façon, il serait possible à partir d'un client WGFS comme Gocad, d'importer et de venir mettre à jour des modèles 3D de SIGEOM en venant extraire l'information de la table contenant le modèle ainsi que de celles des tables jointes contenant les propriétés descriptives. Comme SIGEOM possède déjà ses propres processus de mise à jour, seulement la géométrie du modèle 3D pourrait être rendue disponible pour actualisation et non les tables alphanumériques contenant les propriétés jointes au modèle. Au niveau sécurité, comme SIGEOM n'est pas un système ouvert au grand public, seuls les usagers possédants les privilèges appropriés pourraient importer ou modifier les modèles. Aussi, comme le WGFS est un système hautement standardisé, différents clients pourraient être utilisés pour accéder aux modèles 3D de SIGEOM. De cette façon, différents intervenants du milieu géologique travaillant avec différents outils de visualisation/modélisation 3D, pourraient développer des plug-ins pour permettre le dialogue avec l'interface WGFS de SIGEOM.

Chapitre 5 Conclusion

5.1 Bilan et apports de l'étude

Dès les débuts de cette recherche, nous étions convaincus que pour un grand nombre de domaines d'application, l'usage de la 3^e dimension lors de la modélisation d'une réalité territoriale à partir de données à référence spatiale était requis. Les architectures de systèmes de type SIG, CAO ou SGBD, même s'ils présentent des fonctionnalités intéressantes, ne sont pas parfaitement adaptées à la création, à l'interrogation, à l'analyse, au stockage et au partage de modèles volumiques représentant un territoire.

Les travaux menés dans le cadre de cette recherche avaient pour objectif général d'établir les concepts nécessaires à la définition et au développement d'une architecture logicielle qui permettrait le stockage et l'interrogation spécifique de données géométriques 3D. Cette

architecture logicielle, que nous avons nommée SIG 3D, devait s'approprier les fonctionnalités typiques des systèmes d'information géographique actuels. Partant des concepts liés aux SIG, nous avons proposé la liste suivante des fonctionnalités associées à un SIG 3D :

- construire, manipuler, gérer et présenter des primitives géométriques de type point, ligne, polygone et volume à l'aide de l'une ou de plusieurs structures géométriques 3D
- gérer le stockage des modèles géométriques 3D au sein d'un SGBD
- gérer la sémantique des objets spatiaux 3D par la gestion de leurs propriétés descriptives
- réaliser de l'analyse spatiale métrique et topologique 3D (ce qui implique une algorithmie sous-jacente tenant compte de la troisième dimension lors des calculs)
- diffuser des modèles 3D en particulier sur le Web à l'aide par exemple de techniques modernes comme les services Web
- être interopérable et adopter les différents standards offerts notamment par les principaux organismes de standardisation comme l'ISO et l'OGC

Nous avons par la suite proposé une description et un bilan détaillés de ces fonctionnalités 3D. Ce bilan a été dérivé de plusieurs lectures. En particulier, les aspects de standardisation 3D et de diffusion 3D étaient peu couverts par la littérature. Dans ce sens, ce bilan fournit une présentation intégrée singulière nécessaire à l'élaboration d'un SIG 3D. Par 3D, nous faisons référence à la gestion et à la manipulation d'objets ponctuels, linéaires, surfaciques et volumiques positionnés dans un univers 3D. Point d'information important mis en évidence dans notre travail.

Sur la base de ces fonctionnalités, nous avons par la suite testé et comparé différentes géotechnologies (comparaison unique et rarissime dans la littérature actuelle). Les résultats de cette évaluation ont montré que les outils de type CAO et DAO présentent une bonne capacité de modélisation 3D mais que les autres aspects évalués (analyse spatiale 3D, partage et stockage des modèles 3D, adoption des standards) demeurent encore peu exploités et la majorité des applications offrant de telles capacités reste orientée vers le 2D.

Puis nous avons proposé une architecture SIG 3D orientée service Web et avons examiné les aspects opérationnels de mise en œuvre d'une telle architecture. L'architecture propose d'extensionner un outil CAO puis lui adjoindre des fonctionnalités de requêtes géométriques et sémantiques (une nouveauté), qui elles s'effectueront sur un serveur qui loge une base de données stockant ces modèles volumiques sous format GML (autre aspect novateur). Cette architecture est conforme aux standards ISO et OGC et se réalise dans une approche de service Web (deux aspects novateurs par rapport aux travaux précédant recensés dans la littérature).

Nous avons conclu le tout en développant et testant cette architecture pour un cas réel de modélisation géologique. Le prototype nommé le Web Geological Feature Server (WGFS) est unique et répond adéquatement aux exigences posées pour une architecture de type SIG 3D et aux besoins du partenaire avec lequel nous avons simulé cet exercice (Géologie Québec). Le client est formé de l'outil de modélisation 3D Gocad auquel nous avons développé et greffé un plug-in lui permettant de communiquer à travers le Web avec le serveur WGFS afin de requêter, importer, actualiser et analyser les modèles 3D et leurs propriétés descriptives stockés au sein de la base de données sur un serveur distant. Le requêtage des propriétés descriptives des objets du modèle 3D s'effectue par un simple clic sur l'objet désiré. Le composant serveur, construit à partir de technologies *Open Source* et portables comme Deegree, implémente le schéma spatial ISO 19107 pour la gestion des primitives géométriques autant 2D que 3D ainsi qu'un WFS permettant le requêtage, l'import et l'export des données spatiales 3D sur le Web par le biais du format GML. Deegree implémentant à la base le GML, le WFS et le schéma spatial ISO 19107 dans une optique uniquement 2D (soit le profil « *Simple Features* »), une grande part des efforts de développement du WGFS a dû être consacrée à amener ces composantes vers la gestion des volumes. Le serveur accueille aussi la base d'une bibliothèque d'algorithmes d'opérateurs spatiaux 3D rendue disponible aux clients du système, dont un premier algorithme, le BBOX 3D a été implémenté. Au niveau de la base de données, les détails d'implémentation de la structure permettant le stockage de modèles 3D sous format GML et de ses propriétés descriptives au sein du SGBD open source MySQL ont été présentés.

L'étape suivante pourrait être une intégration un peu plus directe du WGFS avec le système de notre partenaire soit SIGEOM, par une intégration au niveau des bases de données, afin de pouvoir y stocker leurs modèles géologiques 3D. Cette intégration répondrait encore davantage à un important besoin exprimé par les gens du MRNF depuis le début de nos travaux, qui souhaiteraient pouvoir stocker leurs modèles géologiques 3D au sein même de la base de données d'où ils ont extrait les données de base ayant servies à leur construction soit SIGEOM. Nous pensons que le WGFS serait parfaitement adaptable pour cette tâche et les grandes lignes permettant d'y arriver ont été présentées.

Le prototype WGFS et les résultats de ces travaux ont donné lieu à plusieurs publications et ont été présentés lors de divers événements tels que le Gocad Mining Users Meeting de Montréal (Desgagné *et al.*, 2006), la Conférence Géomatique 2006 de Montréal (Pouliot *et al.*, 2006), le Gocad Users Meeting à Nancy, France (Desgagné *et al.*, 2007), le GéoCongrès International à Québec (Thomas *et al.*, 2007b) et le 2nd International Workshop on 3D Geo-Information à Delft, Pays-Bas (Pouliot *et al.*, 2007). Ces présentations du prototype WGFS ainsi que nos divers échanges avec notre partenaire Géologie Québec (Thomas, 2007) nous ont permis de constater que celui-ci était fort intéressant, novateur, et propose des composants interopérables faciles à faire évoluer. Évidemment, c'est un prototype et d'autres travaux devraient être réalisés pour le compléter et l'optimiser. Par contre, nous estimons que cette production scientifique montre bien l'importance et la pertinence de nos travaux. De plus, ceux-ci ont parfaitement contribué à l'atteinte des objectifs du projet GEOIDE GeoTopo3D dans lequel celui-ci s'inscrivait (communication personnelle avec Pouliot, responsable de projet). Rappelons que l'objectif principal du projet GeoTopo3D était de développer des outils/méthodes géomatiques 3D pour aider les experts à concevoir des modèles géologiques et hydrogéologiques, et ainsi progresser vers la mise en valeur et l'exploitation équitable des ressources naturelles canadiennes telles, l'eau, les minerais et le pétrole. En particulier nos travaux ont contribué à l'objectif #5 qui visait à livrer de nouvelles méthodes 3D adaptées à la plate-forme Gocad comme des technologies commercialisables et comme des outils pour l'exploration et la gestion plus efficaces des ressources naturelles.

En bout de ligne, nous estimons que le WGFS propose un assemblage pertinent de composantes informatiques qui intègrent à la fois les concepts de modélisation 3D et de services Web tout en adoptant rigoureusement les standards et favorisant une approche interoperable, *Open Source*, ouverte, extensible et portable. Le WGFS représente selon nous, et lors des derniers développements en 2005, l'une des premières infrastructures du genre à être implémentée. Globalement, nous pensons que nos travaux peuvent servir de guide de « bonne conduite » afin d'encadrer et orienter les futurs travaux de développement de SIG 3D.

5.2 Travaux futurs

Comme mentionné précédemment, l'implémentation du WFGS présenté dans cet ouvrage n'est encore qu'un prototype destiné à la validation de certains concepts théoriques et appliquée dans un contexte bien précis, celui de la modélisation géologique 3D. Plusieurs aspects mériteraient des développements complémentaires pour prétendre à une solution commercialisable. Tout d'abord, aucune solution dite SIG 3D ne sera viable sans la disponibilité d'une bibliothèque d'opérateurs spatiaux 3D implémentant des algorithmes tenant compte de la 3^e dimension. De telles bibliothèques, comme celles que l'on retrouve dans les applications SIG 2D ou comme la Java Topology Suite, sont d'une grande importance dans le domaine et n'ont pas encore d'équivalent au niveau 3D. Les bibliothèques actuelles offrant des algorithmes de manipulation géométrique 3D comme CGAL ou Open Cascade sont incomplètes pour nos besoins puisqu'elles ne permettent pas de calculer la proximité, l'intersection, l'inclusion, l'égalité ou l'adjacence d'objets géométriques 3D, soit les opérations identifiées plus tôt comme essentielles dans l'implémentation de notre SIG 3D. Une telle bibliothèque d'opérateurs spatiaux topologiques adaptée au 3D, n'existe pas encore ce qui représente selon nous un frein important à l'avancement dans le domaine.

Également, nous pensons que l'utilisation d'une telle solution ne pourra être envisagée sérieusement dans un cadre professionnel sans une nette amélioration des performances lors du requêtage des modèles 3D en ligne. Les applications actuelles dites de *Web Mapping (2D)* ont réussi à atteindre une efficacité raisonnable côté performance et vitesse de réponse dans un contexte Web ce qui a sans aucun doute contribué à leur succès. Toutefois, les

modèles géométriques 3D par leur nature même, notamment ceux composés de solides pleins i.e. remplis de primitives volumiques telles que des tétraèdres, sont en général beaucoup plus complexes et volumineux que les modèles 2D. Les travaux futurs en 3D devront se pencher sur cette question afin d'optimiser les transferts sur le Web de ces modèles surchargés par cette complexité accrue qu'apporte la gestion de la 3^e dimension. Au niveau de l'implémentation du schéma spatial ISO 19107 et des outils d'import-export vers leur correspondance dans la syntaxe du GML, les classes d'objets entrant dans la composition des modèles 3D telles que le `GM_Solid` ou le `GM_CompositeSolid` ont été développées. Toutefois, il reste encore du travail à faire notamment au niveau de l'implémentation des classes d'objets topologiques 3D comme le `TP_Solid` afin de permettre au WGFS de gérer explicitement les modèles topologiques 3D. Une fois réalisé, une structure topologique 3D telle que le GeoTEN, spécialement conçue pour les besoins de la modélisation géologique, pourrait être intégrée au WGFS en stockant ces modèles au sein d'un SGBD tel que présenté par Lachance (2005) et en développant le *datastore* approprié i.e. permettant d'établir la correspondance entre les primitives géométriques et relations topologiques du GeoTEN et celles du Schéma Spatial ISO 19107. En fait, étant donné la grande flexibilité du Schéma Spatial ISO 19107 au niveau de la représentation des objets géométriques, nous pensons que la grande majorité des structures géométriques 3D existantes (orientées-objets) pourraient être implémentées au sein du WGFS.

Aussi, lors de la programmation du plug-in Gocad, nous nous sommes concentrés uniquement sur la conversion des objets volumiques entre le GML et le format Gocad. Des travaux sont présentement en cours pour venir compléter cet effort afin de permettre d'interchanger des objets ponctuels, linéaires et surfaciques entre Gocad et le WGFS.

Il serait aussi intéressant de brancher d'autres types de clients au WGFS. On peut penser par exemple à un visualisateur X3D se branchant au WGFS afin de visualiser, manipuler et analyser les modèles 3D stockés au sein d'une base de données distante. Une telle conversion entre la syntaxe du GML et celle du X3D, toutes deux reposant sur le XML, pourrait relativement facilement être réalisée par le biais du XSLT (eXtensible Stylesheet Language Transformations), un langage spécifiquement destiné à la conversion entre différents documents XML.

En terminant, il serait important de mentionner que le domaine du 3D, de la modélisation géométrique, des normes et particulièrement des technologies touchant le Web évoluent très rapidement. En ce sens, il sera important de rester à l'affût des prochains changements et travaux de recherche susceptibles de venir améliorer ou changer la donne quant aux différents concepts pris en compte dans l'architecture de SIG 3D que nous proposons dans ce projet de recherche.

Bibliographie

- Anselin, L. (1995). Local indicators of spatial association - LISA. *Geographical Analysis*, vol.27, no2, pp.93-115.
- Apel, M. (2004). A 3d geoscience information system framework. Thèse de Doctorat, École Supérieure de Géologie, École Doctorale RPPE, Institut National Polytechnique de Lorraine, Nancy, France, 93p.
- Arens, C., Stoter J.E., van Oosterom, P.J.M. (2003). Modelling 3D spatial objects in a GeoDBMS using a 3D primitive. In *Proceedings AGILE*, April 2003, Lyon, France.
- Baumgart, B.G. (1975). Winged-Edge Polyhedron Representation for Computer Vision. *National Computer Conference*, May 1975, Stanford University.
- Bédard, K. (2006). La construction de modèles géologiques 3D : Du tutorial à la standardisation. Mémoire de maîtrise, Département des sciences géomatiques, Université Laval, Québec, Canada, 141p.
- Benouamer, M.O., Michelucci, D. (1997). Bridging the Gap between Brep and CSG via a Triple Ray Representation. in *Proceeding of 4th Symposium on Solid Modeling and Applications*, Ch. Hoffmann and Wim Bronsvort Editors, ACM Press, 1997, pp.68-79.

- Bode, T., Breunig M., Cremers, A.B. (1994). Implementation of elementary geometric database operations for a 3D-GIS. Sixth international Symposium on Spatial Data Handling Edinburgh, 5th - 9th September, pp.604-617.
- Bruenig, M., Zlatanova, S. (2004). 3D Geo-DBMS. Directions-Magazine, November 06, http://www.directionsmag.com/article.php?article_id=694.
- Breunig, M., Schilberg, B., Thomsen, A., Kuper, P.V., Jahn, M., Butwilowski, E. (2009). DB4Geo: Developing 3D geo-database services. 3DGeoInfo Workshop, Nov. 4-5th 2009, Ghent, Belgium.
- Bric, V. (1993). 3D Vector data structures and modelling of simple objects in GIS. MSc thesis, ITC, The Netherlands.
- Brodeur, J., Bédard, Y., Edwards, G., Moulin, B., (2003). Revisiting the concept of geospatial data interoperability within the scope of human communication processes, Transactions on GIS. vol.7 no2, pp.243-265.
- Caumon, G., Lepage, F., Sword, C.H. et Mallet J.L. (2004). Building and editing a sealed geological model. Mathematical Geology, vol.36, no4, May 2004, pp.405-424.
- Chrisman, N., (2002). Exploring geographic information systems. John Wiley & Sons, 2nd. Ed., 320p.
- Clarke, K.C. (1986). Advances in geographic information systems. Computers, Environment and Urban Systems, vol.10, no3/4, pp.175-186.
- Coors, V., Jung, V. (1998). Using VRML as an Interface to the 3D Data Warehouse. Proceedings of the third symposium on the Virtual reality modeling language, Monterey, California, United States , pp.121-140.
- Coors, V. (2003). 3D GIS in networking environments. Computer, Environments and Urban Systems, vol.24, no2, pp.345-357.
- Date, C. J. (2000). An Introduction to Database Systems. Addison Wesley Longman; 7th edition 938 p.
- De la Losa. A., Cervelle. B. (1999). 3D topological modelling and visualisation for 3D GIS. Computer & Graphics, vol.23, no4, pp.469-478.
- De la Losa, A. (2000). Modélisation de la troisième dimension dans les bases de données géographiques. Thèse de Doctorat, Sciences de l'Information Géographique, Université de Marne-la-Vallée, Marne-la-Vallée, 175p.
- Desgagné, E., Pouliot, J., Badard, T. (2006). Development of a Web Geological Feature Service (WGFS) for delivery and querying of 3D objects in Gocad. Gocad Mining Users' Meeting VI, 4 octobre, Mtl.

- Desgagné, E., Pouliot, J., Badard, T., Bédard, K. (2007). Extending Gocad through 3D Web Feature Service. Gocad users meeting, 18-19 juin, Nancy, France.
- Directions Staff, (2003). Daratech Reports GIS Revenues Forecast to Grow 8% to \$1.75 billion in 2003; Utilities and Government Increase Spending. Directions-Magazine, August 09, http://www.directionsmag.com/article.php?article_id=403.
- Egenhofer, M.J., Franzosa, R. (1991). Point Set Topological Spatial Relations. International Journal of Geographical Information Systems. Vol.5, no2, pp.161-174.
- Egenhofer, M.J., Herring, J. (1990). A mathematical framework for the definition of topological relationships. Proceedings of the Fourth International Symposium on Spatial Data Handling, Columbus, Ohio, pp. 803-813.
- Fallara, F., Legault, M., Cheng, L.Z., Rabeau, O., Goutier, J. (2004). 3D model of a segment of the Porcupine-Destor Fault, metallogenic synthesis of Duparquet (phase 2/2) (in French). Ministère des Ressources naturelles et de la Faune du Québec. 3D 2004-01. CD.
- Fallara, F., Legault, M., Rabeau, O. (2006). 3-D integrated geological modeling in the Abitibi Subprovince (Québec, Canada): Techniques and applications. Exploration and Mining Geology, January 2006, vol.15, no1-2, pp.27-43.
- Fotheringham, A.S., Wegener, M. (eds.) (2000). Spatial models and GIS: New potential and new models. Taylor and Francis, vol.1999, London, 279p.
- Gallaher, M., O'Connor, A., Dettbarn Jr., J., Gilday, L. (2004). Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. NIST GCR 04-867. Gaithersburg, MD: National Institute of Standards and Technology, August 2004, NIST GCR 04-867, 194p.
- Gerbe, P. (2002). Outils, données et applications de l'information géographique en 3D. GeoEvenement, Paris.
- Gocad Developer Guide, (2005). Marsh 21, Version 2.1.
- Goodchild, M.F. (2000). Part 1 Spatial analysts and GIS practitioners: The current status of GIS and spatial analysis. Journal of Geographical system. Vol.2, no1, pp.5-10.
- Gröger G., Kolbe, T. H., Czerwinski, A., Nagel C. (2008). OpenGIS® City Geography Markup Language (CityGML) Encoding Standard. Open Geospatial Consortium Inc. Reference number of this OGC® project document: OGC 08-007r1 Version: 1.0.0 Category: OpenGIS® Encoding Standard.
- Hoffmann, C.M. (1992). Geometric and solid modeling. Morgan Kaufmann, 2nd Ed., San Mateo, California.

- Kottman, C. (1999). The Open GIS Consortium and progress toward interoperability in GIS. In Goodchild M, Egenhofer M, Fegeas R, and Kottman C (eds) *Interoperating Geographic Information Systems*. Norwell, MA, Kluwer, pp.39-54.
- Haala, N., Becker, S., Kada, M. (2006). Cell decomposition for the generation of building models at multiple scales. In: *Proceedings of the ISPRS Symposium on Photogrammetric Computer Vision*. Universität Stuttgart : Sonderforschungsbereich SFB 627, 6p.
- Hagedorn B., Dollner, J. (2007). High-Level Web Service for 3D Building Information Visualization and Analysis. *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems ACM*, Seattle, Washington, pp.1-8.
- Chang, E., Li. S. (2008). Architectural Design and Prototyping of a Web-based Synchronous Collaborative 3D GIS. *Cartography and Geographic Information Science*, vol.35, no.2, pp.117-132.
- Kolbe, T. H. (2009). Representing and Exchanging 3D City Models with CityGML. Presented as keynote talk at the 3rd International Workshop on 3D Geo-Information, 13-14 November in Seoul, South Korea.
- Lachance, B. (2005). Développement d'une structure topologique de données 3D pour l'analyse de modèles géologiques. *Mémoire de maîtrise*, Université Laval, Québec, Canada, 116p.
- Lake, R., Burggraf, D., Trninic, M., Rae, L. (2004). *Geography Mark-Up Language – Foundation for the Geo-Web*. Wiley, ISBN 0-470-87154-7.
- Losier, L.-M., Pouliot, J. et Fortin, M. (2007). 3D geometrical modeling of excavation units at the archaeological site of Tell ‘Acharneh (Syria). *Journal of Archaeological Science*, vol.34, no2, February 2007, pp.272-288.
- Kunwoo, L. (1999). *Principle of CAD/CAM/CAE Systems*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 582p.
- Lapierre, A., Cote, P. (2007). Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard, CAD/GIS/BIM services. *Le GeoCongrès*, Québec, 2 au 5 Octobre.
- Lattuada, R., Raper, J. (2004). Large-Scale Three-dimensional Geographic Information Systems meet the systems for Architecture, Engineering and Construction. September 08, *Direction-Magazine*, http://www.directionsmag.com/article.php?article_id=651.
- Ledoux, H., Gold, C.M. (2008). Modeling Three-dimensional Geoscientific Fields with the Voronoi Diagram and its Dual. *International Journal of Geographical Information Science*, vol.22, no5, pp.547-574.

- Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (eds) (1999). *Geographical Information Systems: Principles, Techniques, Management and Applications*. John Wiley & Sons, New York.
- Longley, P.A., Goodchild, M.F., Maguire, D.J. et Rhind, D.W. (2001). *Geographic Information Systems and Science*. Chichester: John Wiley.
- Martens, F., Legrand, P., Legrand, J., Loots, L., Waelkens, M. (2000). *Computer-Aided Design and Archaeology at Sagalassos: Methodology and Possibilities of Reconstructions of Archaeological Sites*. *Virtual Reality in Archaeology*, J. Barcelo, M. Forte, and D. Sanders, eds., ArcheoPress, 2000, pp.205-212.
- Massé, L. (2003). *Développement méthodologique pour la modélisation géologique avec exemple pour la région de Moose Mountain, AB*. Mémoire de Maîtrise, Département de Géologie et de Génie Géologique, Université Laval, Québec, Canada, 71p.
- Molenaar, M. (1990). A formal data structure for 3D vector maps. *Proceedings of EGIS'90*, Amsterdam, The Netherlands, EGIS Foundation, p.780-781.
- Herring, J.R. (2001). *The OpenGIS™ Abstract Specification Topic 1: Feature Geometry (ISO 19107 Spatial Schema.) Version 5*. OGC document 01-101.
- Lieberman, J., Sonnet J., (2003). *Web Terrain Service (WTS)*. Open GIS Consortium Inc. OpenGIS® Implementation Specification, Request for Comment, version 0.5 document OGC 03-081r2.
- OGC (1999). *OpenGIS Simple features specification for SQL(SFS)*. Revision 1.1 OGC document 99-049, 5 May 1999.
- OGC (2004a). *Geography Markup language (GML) Implementation Specification, Version 3.1.0*. OGC document 03-105r1.
- OGC (2004b). *Web Map Service (WMS), Version 1.3*. OGC document 04-024.
- OGC (2005a). *Filter Incoding Implementation Specification Version 1.1.0*. document 05-095.
- OGC (2005b). *Web Feature Service Implementation Specification (WFS), Version 1.1.0*. document 04-094.
- OGC (2006). *Web Coverage Service Implementation Specification (WCS), Version 1.1.0*. document 06-083r8.
- OQLF (2007). *Le grand dictionnaire terminologique*. Office Québécois de la Langue Française. http://www.granddictionnaire.com/btml/fra/r_motclef/index1024_1.asp (Accédé en novembre 2007).

- Oracle (2008). Oracle® Spatial Developer's Guide. Version 11g Release 1 (11.1)
http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_intro.htm
 (Accédé en octobre 2008).
- Payrazyan, K. (1998). Modélisation 3D des réservoirs pétroliers par l'intégration des données sismiques et géologiques: Approches quantitatives multivariées. Ph.D. Thesis, Ecole Nationale Supérieure de Géologie: Institut National Polytechnique de Lorraine, Nancy France.
- Pilouk, M. (1996). Integrated modelling for 3D GIS. Thèse de doctorat, International Institute for Geo-Information Science and Earth Observation, The Netherlands, 200p.
- Pouliot, J., Lachance, B., Brisebois, A., Rabaud, O., Kirkwood, D. (2003). 3D geological modeling: Are GIS or CAD appropriate? ISPRS Workshop, WG II/5, II/6, IV/1 and IV/2 Joint Workshop on "Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis", October, 2-3, 2003, Québec, Canada.
- Pouliot, J., Lachance, B., Kirkwood, D. (2005). L'importance de la modélisation géométrique pour les SIG-3D. Exemple du développement d'un modèle topologique 3D pour une application géologique. Université Laval, Québec, Canada.
- Pouliot, J., Desgagné, E., Badard, T., Bédard, K. (2006a). SIG 3D : Où en sommes-nous et quelles sont les avenues de développement ? Conférence Géomatique 2006 de l'association canadienne des sciences géomatiques, 25-26 octobre, Montréal.
- Pouliot J., Lachance, B., Kirkwood, D. (2006b). L'importance de la modélisation géométrique 3D lors de l'élaboration d'un SIG 3D: Exemple du développement d'une structure topologique pour une application géologique. Revue internationale de géomatique: Information géographique tridimensionnelle, vol.16, no1, pp.29-49.
- Pouliot, J., Desgagné, E., Badard, T., Bédard, K., Thomas, V. (2007). Development of a Web Geological Feature Server (WGFS) for sharing and querying of 3D objects. Advances in 3D Geoinformation systems - Lecture Notes in Geoinformation and cartography, Springer, eds van Oosterom et al., pp.115-130.
- Pouliot, J., Bédard, K., Kirkwood, D., Lachance, B. (2008). Reasoning about geological space: Coupling 3D Geomodels and topological queries as an aid to spatial data selection. Computers and Geosciences vol.34, no5, pp.529-541.
- Rahman, A.A. (2000). The design and implementation of two and three-dimensional triangular irregular network (TIN) based GIS. PhD thesis, University of Glasgow, Scotland, United Kingdom, 250
- Rahman, A.A., Zlatanova, S., Pilouk, M. (2001). The 3D GIS software development: global efforts from researchers and vendors. Geoinformation Science Journal, vol.1, no2.

- Raper, J. and Maguire, D.J. (1992). Design models and functionality in GIS. *Computer & Geosciences*, vol.18, no4, pp.387-394.
- Requicha, A.A.G., Voelcker, H.B. (1982). *Solid Modeling: A Historical Summary and Contemporary Assessment*. Production Automation Project The University of Rochester. February 1982 (Vol. 2 No. 2) pp. 9-24.
- Requicha, A. (1996). *Geometric Modeling: A First Course*. University of Southern California. <http://www-pal.usc.edu/~requicha/book.html> (Accédé en août 2008).
- Samet, H. (1995). Spatial data structures in Modern Database Systems: The Object Model, Interoperability, and Beyond. W. Kim, Ed., Addison-Wesley/ACM Press, 1995, pp.361-385.
- Schilling, A., Kolbe, T.H. (2010). Draft for Candidate OpenGIS® Web 3D Service Interface Standard. OpenGIS® Discussion Paper. Version 0.4.0, document OGC 09-104r1.
- Shi, W.Z., Yang, B.S. et Li, Q.Q. (2003). An Object-Oriented data model for complex objects in three-dimensional Geographic Information Systems. *International Journal of Geographical Information Science*. Vol.17, no5, pp.411-430.
- Schneider, F., Wolf, S., Faille, I. et Pot, D. (2000). A 3D-basin model for hydrocarbon potential evaluation: application to Congo offshore. *Oil and Gas Science and Technology, Rev. IFP*, Copyright © 2000, Éditions Technip, vol.55, no1, pp.3-13.
- Smith, M.J. de, Goodchild, M.F. et Longley P.A. (2006). *Geospatial Analysis - a comprehensive guide*. 2nd edition © 2006-2008. <http://www.spatialanalysisonline.com> (Accédé en mars 2008).
- Stadler, A., Kolbe T. H., (2007). Spatio-semantic coherence in the integration of 3d city models. *Spatial data quality*. 5th International Symposium, 13-15 June ITC Enschede, NL, 8p.
- Stadler, A., Nagel, C., König, G., Kolbe, T. H. (2009). Making interoperability persistent: A 3D geo database based on CityGML. Presented at the 3rd International Workshop on 3D Geo-Information, 13.-14. November 2008 in Seoul, South Korea. The paper is published in Lee, Zlatanova (eds.): *3D Geo-Information Sciences*, Springer, 18p.
- Stoter, J. et Salzmann, M. (2003). Towards a 3D cadastre: where do cadastral needs and technical possibilities meet? In: *Computers, Environment and Urban Systems*, Theme Issue: 3D Cadastres, July 2003, vol.27, no4, 22p.
- Stoter, J.E., Zlatanova, S. (2003a). Visualising and editing of 3D objects organised in a DBMS. In: *Proceedings EUROSDR Workshop: Rendering and Visualisation*, (January 2003), Enschede, The Netherlands, 14p.

- Stoter, J.E., Zlatanova, S. (2003b). 3D GIS, where are we standing? ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, Québec, October 2003, 8p.
- Sprague, K., de Kemp, E., Wong, W., McGaughey, J., Perron, G., Barrie, T. (2006). Spatial targeting using queries in a 3-D GIS environment with application to mineral exploration. *Computers & Geosciences* vol.32, n.3, pp.396-418.
- Thomas, V. (2007). Rapport d'audi WGFS: Travaux effectués au Ministère des Ressources naturelles et Faune à Val d'Or. Rapport interne projet GeoTopo3D. Département des sciences géomatiques. Université Laval, Québec, Canada, 14p.
- Thomas, V., Pouliot, J., Badard, T., Desgagné, E. (2007b). Conception et développement d'un SIG-3D dans une approche de service Web. Le GéoCongrès Internationale, du 2 au 5 octobre, Québec, Canada, (Poster).
- Thomas, V.E. Nyaminani, E. (2008). Échange automatisé et contrôlé de données géométriques et descriptives issues d'une modélisation 3D. Rapport technique - Projet de génie II, Département des sciences géomatiques, Université Laval, Québec, Canada, 16p.
- Tsuzukil, M.G., Takase, F.K., Garcia, M.A.S. et Martins, T. (2007). Converting CSG models into meshed B-Rep models using euler operators and propagation based marching cubes. *J. Braz. Soc. Mech. Sci. & Eng.*, Oct./Dec. 2007, vol.29, no.4, ISSN 1678-5878, pp.337-344.
- Turner, A.K. (ed.) (1992). Three-dimensional modeling with geoscientific information systems. Kluwer Academic, Dordrecht, The Netherlands, 468p.
- Verbree, E., Van der Most, A., Quak, W., Van Oosterom, P. (2005). Towards a 3D feature overlay through a tetrahedral mesh data structure. *Cartography and Geographic Information Science*, vol.32, n.4, October 2005, pp.303-314.
- Vries, M.E. de, Zlatanova, S. (2004). Interoperability on the Web: the case of 3D geo-data. Paper submitted to IADIS International Conference on e-Society, Spain, July, 8p.
- Vries, M.E. de, and Stoter, J. (2003). Accessing 3D geo-DBMS using Web technology. ISPRS Joint Workshop on Spatial, Temporal and multi-dimensional data modeling and analysis. Québec, October, 8p.
- Zlatanova, S., Rahman, A.A. et Pilouk, M. (2002). Trends in 3D GIS development. In: *Journal of Geospatial Engineering*, vol.4, n.2, pp.1-10.
- Zlatanova, S., Rahman, A., Pilouk, M. (2002b). 3D GIS: current status and perspectives, in: *Proceedings of ISPRS*, 8-12 July, Ottawa, Canada, 8p.

- Zlatanova, S. (2000). On 3D Topological Relationships. Proceedings of the 11th International Workshop on Database and Expert Systems Applications. Dept. of Geodesy, Delft Univ. of Technol., Netherlands, 913p.
- Zlatanova, S. (2000b). 3D GIS for Urban Development. Thèse de doctorat, Graz University of Technology, Graz, Austria, 212p.
- Zlatanova, S., Tempfli, K. (2000c). Modeling for 3d gis: spatial analysis and visualisation through the web. International archives of photogrammetry and remote sensing. IAPRS, vol.33, Amsterdam.

Annexe A Diagrammes de classe du schéma spatial ISO 19107

Cette annexe présente les diagrammes de classe de la géométrie de base et de la topologie du schéma spatial ISO 19107 (OGC, 2001).

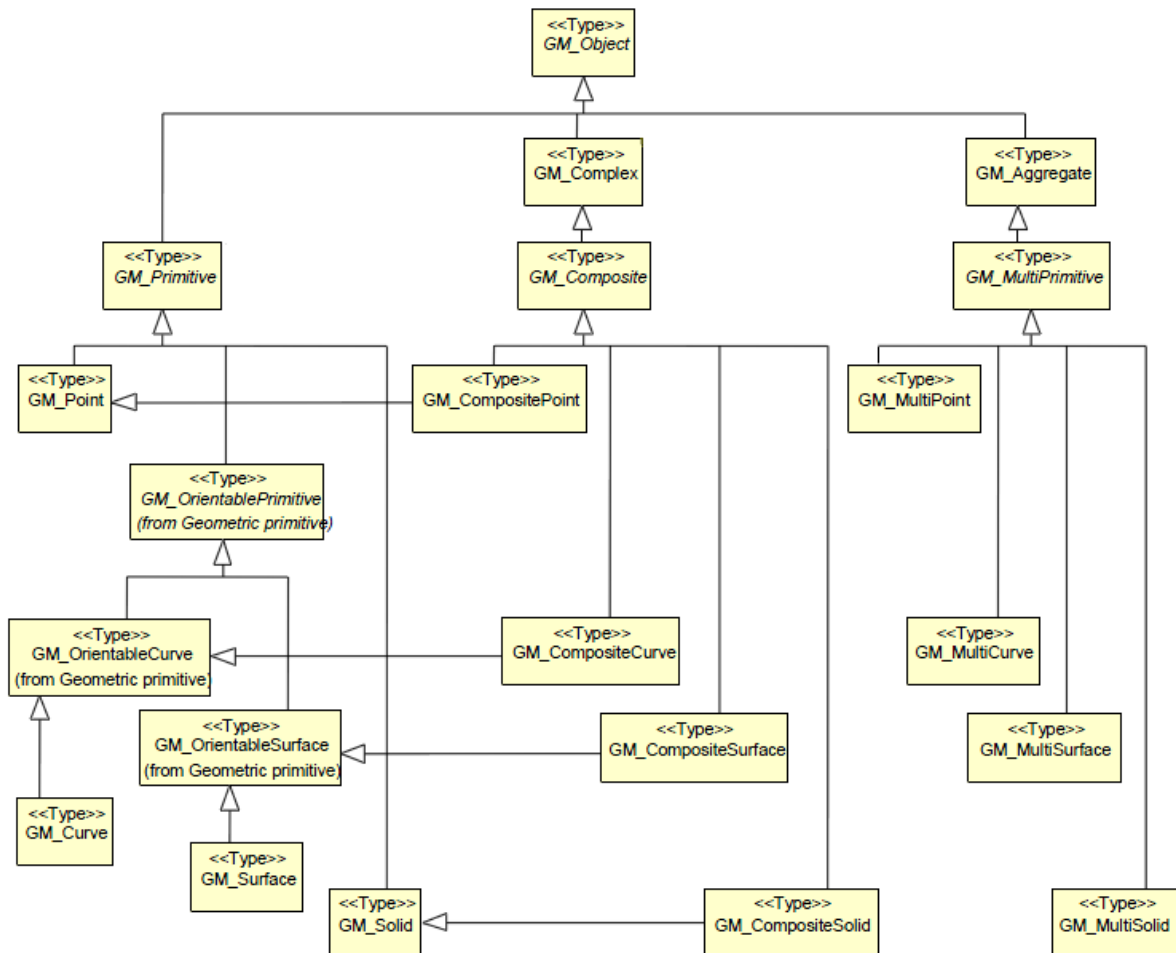


Figure 24 : Diagramme de classe de la géométrie de base ISO 19107.

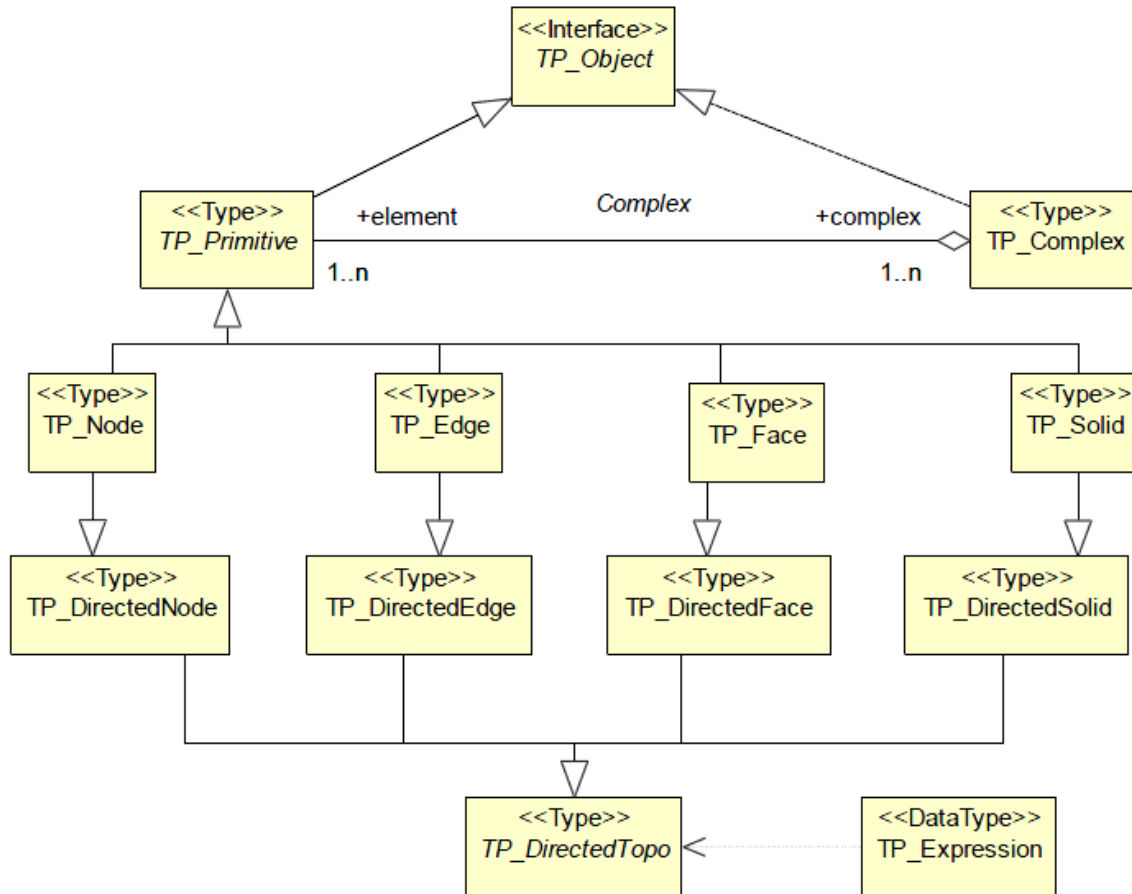


Figure 25 : Diagramme de classe des primitives topologiques ISO 19107.

Annexe B Représentation d'objets volumiques sous format GML

Cette annexe présente les détails de la représentation d'un objet volumique sous format GML. Il existe dans le GML des balises spécifiques pour la représentation des entités comme les GM_Solid et GM_Composite identifiées précédemment comme éléments nécessaires à la représentation des modèles volumiques. La figure suivante montre un exemple de représentation en format GML d'une entité (unité géologique) avec les propriétés qui s'y rattachent dont entre autres une propriété géométrique qui décrit un composite solide à l'aide de tétraèdres.

```

- <app:porcupine gml:id="porcupine_e03f4017010edda5dde000000193a00">
- <gml:boundedBy>
- <gml:Envelope>
  <gml:pos srsDimension="3">645954.25 5370906.0 -600.0</gml:pos>
  <gml:pos srsDimension="3">646313.0625 5371254.0 320.560455</gml:pos>
</gml:Envelope>
</gml:boundedBy>
<app:id>10</app:id>
<app:name>nordES</app:name>
<app:lithology>V1B</app:lithology>
<app:litho_desc>Rhyolite</app:litho_desc>
<app:stratigraphy>[arch]ln1</app:stratigraphy>
<app:stra_desc>Archeen, Formation de Lanaudiere</app:stra_desc>
- <app:GEOM>
- <gml:CompositeSolid xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:solidMember>
- <gml:Solid xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:exterior>
- <gml:CompositeSurface xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
- <gml:surfaceMember>
- <gml:OrientableSurface orientation="+">
- <gml:baseSurface>
- <gml:Polygon xmlns:gml="http://www.opengis.net/gml">
- <gml:outerBoundaryIs>
- <gml:LinearRing>
  <gml:coordinates cs="," decimal="." ts="">646069.8125,5371092.5,-270.100555
646017.875,5371137.0,-205.76387 646096.1875,5371089.0,-174.089264
646069.8125,5371092.5,-270.100555</gml:coordinates>
</gml:LinearRing>
</gml:outerBoundaryIs>
</gml:Polygon>
</gml:baseSurface>
</gml:OrientableSurface>
</gml:surfaceMember>
+ <gml:surfaceMember>
+ <gml:surfaceMember>
+ <gml:surfaceMember>
</gml:CompositeSurface>
</gml:exterior>
</gml:Solid>
</gml:solidMember>
+ <gml:solidMember>
+ <gml:solidMember>

```

Figure 26 : Exemple de représentation en format GML d'une unité géologique.

Dans cet exemple, les préfixes *app* et *gml* sont des espaces de nom (namespaces). Dans le contexte du XML, l'espace de nom est une recommandation du W3C qui a comme objectif d'identifier à quelle grammaire XML appartient un élément. En effet, plusieurs grammaires XML différentes peuvent utiliser un même mot pour identifier des entités différentes. Dans une telle situation, c'est l'espace de nom qui va permettre d'identifier de manière unique les balises. Dans notre exemple, l'espace de nom *gml* fait référence aux éléments de la grammaire GML de l'OGC tandis que *app* fait référence aux éléments que nous avons nous-mêmes créés et qui sont propres à l'application en cours.

L'unité géologique présentée dans cet exemple fait partie d'un modèle géologique (*app:porcupine*) et est identifiée par son attribut *gml:id*. Les balises *app:id*, *app:name*, *app:lithology*, *app:litho_desc*, *app:stratigraphy* et *app:stra_desc* sont les propriétés descriptives rattachées à cette unité géologique. La propriété *app:GEOM* est celle qui contient les caractéristiques géométriques de notre unité géologique. La balise *gml:CompositeSolid* correspond à la classe *GM_CompositeSolid* du schéma spatial de l'ISO qui est à la base des collections de solides. Une telle collection possède une certaine quantité de *gml:solidMember* dans lesquels sont définis des balises (*gml:solid*) correspondants à la classe *GM_Solid* du schéma de l'ISO. La balise *gml:exterior* correspond à la coquille extérieure qui permet de représenter la limite externe de chaque solide ce qui correspond à un *GM_Shell* dans le langage ISO. Un *GM_Shell* est représenté ici par une *gml:CompositeSurface* qui possède à son tour un certain nombre de *gml:surfaceMember*. Chacun des membres de ce composite est une surface orientée *gml:OrientableSurface* définie comme un polygone (*gml:Polygon*) délimité par son contour, soit une polyligne (*gml:LinearRing*). En résumé, le *gml:CompositeSolid* est l'unité géologique et les solides qu'il contient sont des tétraèdres représentés par leurs surfaces qui se composent de quatre triangles. On a donc ici la représentation complète d'un solide tétraédrique et ses propriétés descriptives en format GML.

Annexe C Détails des spécifications de service Web de L'OGC

Web Map Service (WMS)

La spécification WMS définit trois opérations standards soit *GetCapabilities*, *GetMap* et *GetFeatureInfo*. Ces opérations permettent d'obtenir respectivement les métadonnées des jeux de données disponibles sur le serveur, la carte elle-même et de l'information descriptive sur des objets spécifiques figurant sur la carte.

La requête *GetCapabilities* est commune à tous les services Web de l'OGC et permet au système de décrire ses capacités, c'est-à-dire indiquer et décrire les données qu'il peut servir et quelles opérations sont supportées. Typiquement, c'est donc la première requête que l'on envoie à un service web de l'OGC.

Dans une requête *GetMap*, les *layers*, les styles, le *bounding box*, le système de référence, la taille de l'image, son format, etc. sont passés au service Web en paramètre. Le résultat de cette requête est une image 2D conforme aux paramètres reçus. C'est donc l'opération la plus courante qui permet de servir dynamiquement des cartes en ligne sous forme d'image.

La requête *GetFeatureInfo* dont l'implémentation est optionnelle sert à implanter un outil d'identification pour connaître l'information descriptive rattachée aux entités spatiales comme dans un SIG.

Web Terrain Service (WTS)

Un WTS doit supporter les opérations *GetCapabilities* et *GetView*. Lors d'une requête *GetView*, plusieurs paramètres typiques à la navigation dans un espace à trois dimensions doivent être fournis au WTS pour permettre la création de cette vue :

- *Point of Interest (POI)* : position en x,y,z du focus de l'utilisateur
- *Distance* : distance entre l'utilisateur et le POI
- *Pitch* : angle vertical entre l'utilisateur et le POI
- *Yaw* : angle vertical entre la direction nord et la projection horizontale de la ligne du POI de l'utilisateur
- *Angle of View (AOV)* : angle représentant l'étendue du champ de vision

D'autres paramètres peuvent aussi être fournis au WTS pour spécifier les caractéristiques comme les *layers*, les styles, le *bounding box*, le système de référence, la taille de l'image, son format, etc.

Web Feature Service (WFS)

L'interface du WFS propose minimalement les opérations suivantes *GetCapabilities*, *GetFeature* et *DescribeFeatureType*.

Une requête *GetFeature* permet de demander au serveur l'ensemble ou une partie d'un jeu de données géométriques et les données descriptives s'y rattachant. La sélection peut se faire en limitant la quantité de données importées à l'aide d'un filtre (standardisé par le Filter Encoding Specification de l'OGC). Ce filtre permet de préciser quelles sont les entités à importer en se basant sur leurs propriétés descriptives ou sur un critère spatial d'une manière semblable à une clause *WHERE* en langage SQL. En retour, les données sélectionnées sont retournées au client encodées selon le format GML.

Pour sa part, une requête *DescribeFeatureType* permet de connaître la structure des jeux de données que le système peut servir. Cette structure est présentée sous forme de schéma XML dans lequel on retrouve pour chaque jeu de données la liste des propriétés ainsi que le type de données qui s'y rattache. La figure suivante présente un exemple de schéma XML provenant d'un WFS résultant d'une requête *DescribeFeatureType* :

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:app="http://www.deegree.org/app"
  xmlns:gml="http://www.opengis.net/gml" xmlns:wfs="http://www.deegree.org/wfs"
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://www.deegree.org/app">
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd" />
  <!-- configuration for the persistence backend to be used -->
  <xsd:element name="porcupine" substitutionGroup="gml:_Feature" type="app:porcupineType" />
  <!-- ===== -->
- <xsd:complexType name="porcupineType">
- <xsd:complexContent>
  - <xsd:extension base="gml:AbstractFeatureType">
    - <xsd:sequence>
      <xsd:element name="id" type="xsd:integer" />
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="lithology" type="xsd:string" />
      <xsd:element name="litho_desc" type="xsd:string" />
      <xsd:element name="stratigraphy" type="xsd:string" />
      <xsd:element name="stra_desc" type="xsd:string" />
      <xsd:element name="GEOM" type="gml:GeometryPropertyType" />
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

Figure 27 : Exemple de schéma XML provenant d'un WFS résultant d'une requête DescribeFeatureType.

Optionnellement, le WFS peut aussi être bonifié pour permettre d'insérer dans la base de données, des données nouvellement créées ou modifiées directement à partir du client. Deux opérations supplémentaires peuvent donc être implémentées pour permettre l'édition en ligne avec un WFS soit la *Transaction* et le *LockFeature*.

Une requête *Transaction* permet de décrire les opérations de transformation qui peuvent être appliquées aux données par l'intermédiaire du WFS. Ces opérations peuvent être la création de nouvelles entités (*CREATE*), la modification (*UPDATE*) ou l'élimination (*DELETE*) de données existantes. Ce genre de transaction peut par contre être problématique si plusieurs personnes tentent de modifier en même temps les mêmes données. Ceci peut être évité à l'aide de « *locks* » qui viennent bloquer l'accès aux données afin d'éviter cette destruction mutuelle entre transactions accédant aux mêmes ressources. Le standard WFS prévoit donc la possibilité d'implanter un tel mécanisme de « *lock* » par le biais de requête *LockFeature*.

Web Coverage Service (WCS)

Un service WCS supporte les trois opérations suivantes:

- GetCapabilities
- DescribeCoverage
- GetCoverage

Alors que l'opération *DescribeCoverage* permet au client de recevoir l'information structurelle portant sur une ou plusieurs couvertures, un *GetCoverage* permettra de recevoir la couverture elle-même dans un format approprié tel que le GeoTIFF, HDF-EOS, NITF ou CF-NetCDF.

Filter Incoding Implementation Specification

Chacun des différents services web de l'OGC permet, lors de requêtes visant l'accès aux données, de spécifier si l'action demandée doit être exécutée sur la totalité du jeu de données ou seulement sur un sous-ensemble de celui-ci. Comme ce besoin de pouvoir identifier un sous-ensemble des données était commun à plusieurs services web de l'OGC, il a été standardisé dans le Filter Incoding Implementation Specification (OGC, 2005a).

L'objectif du Filter Incoding Implementation Specification est de définir un encodage XML standard et indépendant des technologies, pour décrire la représentation d'une requête portant sur des données spatiales et descriptives. Grâce aux nombreux outils XML disponibles aujourd'hui, un tel encodage XML a pour avantage de pouvoir être facilement parcouru, validé et transformé en n'importe quel langage spécifique, utilisé pour accéder aux données entreposées dans n'importe quelle source de stockage. Par exemple, le filtre XML pourrait être transformé en une clause WHERE d'une requête SQL utilisée pour accéder aux données d'une base de données relationnelle ou encore être transformé en expression XPath, XPointer ou XQuery dans le cas de stockage en document XML. Plus spécifiquement, la syntaxe du Filter Incoding Implementation Specification, comprend une série d'éléments et d'opérateurs permettant la construction d'expressions tels que :

- Opérateurs logiques
- Opérateurs de comparaisons
- Opérateurs spatiaux

- Identifiant d'objets
- Opérateurs arithmétiques
- Fonctions

Les objets géométriques qui sont décrits dans un filtre sont encodés en respectant la nomenclature décrite dans la spécification d'implémentation du GML et les opérateurs spatiaux employés sont ceux standardisés par l'OGC. Ainsi, un tel filtre pourrait théoriquement s'appliquer à un modèle 3D puisqu'encore une fois, le GML est un format supportant le 3D et les opérateurs spatiaux mentionnés dans le filtre peuvent très bien implémenter des algorithmes 3D.

Annexe D Inventaires des géotechnologies : Détails techniques des logiciels

Outils CAO : Autocad, Microstation et Gocad

Autodesk (<http://usa.autodesk.com/>) avec sa suite de produit Autocad est une société clé dans le domaine du CAO. Autocad est une plate-forme de conception et de dessins 2D et 3D qui peut être qualifiée de solution AEC typique. La modélisation 3D au sein d'Autocad repose principalement sur les structures géométriques CSG et B-Rep. Des calculs de distance, d'aire et de volume peuvent être appliqués sur les modèles et ceux-ci sont stockés uniquement sous forme de fichier.

Bentley (<http://www.bentley.com/>), avec Microstation, est également un leader pour la conception de dessins 3D. Il combine aussi les structures CSG et B-Rep et offre sensiblement les mêmes capacités qu'Autocad.

Lors de notre inventaire en 2008, les produits de base d'Autocad et de Microstation n'implémentaient pas les standards de l'ISO et de l'OGC, ne permettaient pas l'analyse spatiale topologique, la diffusion Web et le branchement à des bases de données. Toutefois, d'autres produits des mêmes suites tels qu'Autocad Map 3D, Autodesk MapGuide, Autodesk Design Server, Bentley Map, et Bentley GeoWeb Publisher permettent d'étendre les capacités originales de ces produits en leur ajoutant des fonctionnalités se rapprochant beaucoup de celles des SIG telles que le branchement aux bases de données, l'analyse spatiale topologique, l'adoption des standards en géomatique (SFS, GML, WMS, WFS), etc. Ces nouvelles fonctionnalités se limitent toutefois aux modèles géométriques 2D et ne s'appliquent pas aux objets volumiques modélisés par ces outils CAO. Elles ne seront donc pas considérées dans l'évaluation de ces deux produits.

Gocad de Paradigm (<http://www.pdgm.com/>), est un outil CAO spécialement conçue pour la modélisation géologique qui est beaucoup utilisé au niveau gouvernemental, académique ainsi que dans les compagnies minières et pétrolières. La modélisation 3D dans Gocad repose principalement sur trois structures géométriques, soit le B-Rep, le Voxel, et l'ensemble de solides (représentation tétraédrique). Au niveau de l'analyse spatiale, Gocad offre quelques opérateurs d'analyse métrique tels que le calcul de distance, d'aire et de volume. Pour ce qui est du stockage des modèles, Gocad offre uniquement la possibilité d'enregistrer et de lire l'information au sein de fichiers. Aucune fonctionnalité particulière

n'existe non plus pour faciliter le partage des modèles, comme par exemple sur Internet, en dehors du partage de fichier.

Il existe plusieurs autres produits CAO. Mentionnons, entre autres, Autosolids et AltCad qui peuvent être définis comme des clones d'Autocad. Il y a aussi SolidWorks de Dassault Systèmes, Pro/Engineer de Parametric Technology Corporation et SolidEdge de Siemens qui sont également des outils CAO très utilisés dans le domaine de l'ingénierie mécanique.

Outils DAO : 3D Studio Max, Maya et Blender

Parmi les DAO les plus connus, on retrouve 3D Studio Max (3ds Max) de la société Autodesk. 3ds Max est une solution complète de modélisation, d'animation et de rendu 3D destinés aux professionnels de l'industrie du jeu et du milieu cinématographique (<http://usa.autodesk.com/>, 2009). Chez Autodesk on retrouve aussi un autre DAO parmi les plus populaires soit le produit Maya. Maya est un DAO qui offre sensiblement les mêmes fonctionnalités que 3ds Max mais qui se spécialise un peu plus dans le cinéma ou la publicité en offrant des rendus finaux des plus recherchés. Du côté Open source, on retrouve aussi le produit Blender (<http://www.blender.org/>), une suite d'animation complète offrant des capacités de modélisation, d'animation et de rendu 3D. Blender est l'un des produits DAO libres les plus performants et les plus utilisés. Parmi les structures géométriques qu'il supporte, on retrouve le B-Rep, les courbes de Bézier et les NURBS. Il existe plusieurs autres outils DAO disponibles sur le marché. Parmi les plus connus on retrouve Softimage, Houdini, Lightwave 3D, ZBrush, etc.

SIG : ArcGIS, MapInfo et Geomedia

ArcGIS de la compagnie Esri (<http://www.esri.com/>), est un des principaux leaders dans le domaine des SIG. ArcGIS, par le biais de son extension 3D Analyst, propose la création de surface continue (par interpolation) et la manipulation de MNT (sous forme de GRID ou de TIN). Cette même extension offre aussi le module ArcScene qui permet la visualisation dans un environnement 3D des MNT, la construction de modèle 3D par extrusion de primitives 2D ainsi que l'importation de certains formats de modèle de type B-Rep tel que le KML, le VRML, ou ceux de Google Sketchup, 3d Studio Max et OpenFlight (gérés dans ArcGIS sous le format *Multipatch*). Toutefois, lors de notre inventaire, l'édition de ces

modèles importés n'était toujours pas supportée. Les modèles 2D, 2.5D et Multipatch d'ArcGIS peuvent être stockés au sein d'une base de données Oracle ou SQL Server par l'entremise du produit ArcSDE (maintenant ArcGIS Server), intermédiaire entre la base de données et le SIG. Au niveau de l'analyse spatiale, les opérateurs spatiaux métriques et topologiques d'ArcGIS ne tiennent pas compte de la troisième dimension lors des calculs et seulement quelques opérateurs ont été spécialement conçus pour être appliqués sur les modèles d'ArcScène tels que l'intervisibilité entre deux points, les effets d'ombrages ou la création de carte de pente. Au niveau de la diffusion Web, les modèles 2D et 2.5D d'ArcGIS peuvent être publiés par le biais des modules ArcIMS ou ArcGIS Server qui permettent une navigation dynamique. Pour ce qui est des standards, ArcGIS se base sur le SFS pour le stockage des objets géométriques et offre la possibilité de se conformer aux spécifications WMS, WFS, WCS et GML. Le GML offert par ces produits d'ESRI, se limite au profil "*Simple Features*" i.e. objets spatiaux 2D.

Concernant les produits MapInfo (<http://www.pbinsight.com/>), l'extension Vertical Mapper se greffe à leur produit de base pour la création, la visualisation et l'analyse de surface continue (matricielle) et de TIN. Des vues 3D en perspective peuvent être créées en plus de cartes de pente avec inclinaison et orientation. La compagnie Emcom offre aussi le module Engage 3D pour MapInfo venant ajouter la possibilité de travailler avec des modèles volumiques (Voxels). Ce module permet également de visualiser des modèles 3D provenant d'un outil CAO (sous format DXF) et permet d'extruder des objets 2D. Le module SpatialWare est le pendant MapInfo du ArcSDE d'Esri et permet le branchement aux SGBD Oracle, SQL Server, MS Access et IBM Informix. Pour ce qui est de la diffusion Web, MapInfo met à disposition ses produits MapXtreme et Push'n'See afin de publier ses modèles 2D et 2.5D. Au niveau des normes, les produits MapInfo implémentent les SFS, WMS, WFS, GML 2.x et 3.x. (profil "*Simple Features*").

La suite Geomedia d'Intergraph (<http://www.intergraph.com/>) offre également une gamme de produits semblables à ses concurrents Esri et MapInfo. L'extension GeoMedia Terrain est l'équivalent du 3D Analyst d'Esri et du Vertical Mapper de MapInfo et offre sensiblement les mêmes fonctionnalités 3D. Son produit Geomedia WebMap sert de

module pour la diffusion des modèles. Pour ce qui est des normes, les produits GeoMedia supportent les SFS, WMS, WFS et le GML 2.x.

Visualisateur de globes virtuels: Google Earth, Virtual Earth et ArcGIS Explorer

Le plus populaire de ces visualisateurs de Globe Virtuel est certainement le produit Google Earth (<http://earth.google.com/>) développé par Keyhole, acheté et maintenant distribué gratuitement par Google. Google Earth permet aux usagers de draper leurs propres modèles vectoriels 2D, 2.5D et 3D (stockage des coordonnées x,y,z) sur le globe à partir du format de données KML. Depuis sa version 4, il permet de lire les modèles géométriques 3D texturés de type B-Rep stockés sous d'autres formats comme le COLLADA (COLLABorative Design Activity) afin de permettre leur affichage sur le globe virtuel. La construction des modèles s'effectue généralement à l'aide de produits externes (CAO-DAO) comme le CAO Google Sketchup ou 3D Via. Google Sketchup est un outil CAO permettant de créer de manière très intuitive des modèles 3D par extrusion de primitives géométriques 2D et de les exporter dans le format KML. Il existe aussi un plug-in pour ArcGIS permettant l'exportation des modèles de Sketchup vers ce SIG et vice-versa ce qui permet de faciliter l'intégration entre les données SIG, Sketchup et Google Earth. D'ailleurs, la plupart des SIG permettent l'exportation de leurs modèles vers le format KML soit nativement ou par le biais de plug-ins.

La réponse de Microsoft à son concurrent Google et son populaire Google Earth se nomme Virtual Earth (<http://www.microsoft.com/maps/>). Virtual Earth est un produit presque identique à Google Earth qui offre sensiblement les mêmes possibilités (visualisation du relief du territoire et des bâtiments dans un environnement 3D avec drapage d'images satellites). Virtual Earth permet aussi aux usagers d'ajouter leurs propres modèles au visualisateur par le biais du format de son concurrent le format KML.

Suite aux efforts de Google et Microsoft, Esri a développé son propre visualisateur de globe virtuel soit ArcGIS Explorer (<http://www.esri.com>). ArcGIS Explorer se veut un produit offrant les mêmes services que ses concurrents à quelques différences près. ArcGIS Explorer offre une couverture d'image satellitaire de moins bonne qualité que ses concurrents, mais en revanche, il permet d'ajouter sur son globe des données stockées sous

l'un ou l'autre des différents formats de données supportés par les produits d'ESRI (Shapefile, Geodatabase, Service ArcIMS, etc...). De plus, les différents opérateurs spatiaux d'ArcGIS (métriques et topologiques) sont aussi disponibles sous ArcGIS Explorer.

SGBD Spatiaux: Oracle Spatial, SQL Server et PostGIS

Oracle Spatial est le module du SGBD Oracle (<http://www.oracle.com/>) qui prend en charge la gestion des données géométriques pour leur stockage et leur requêtage au sein de la base de données. Oracle Spatial offre aussi une série de fonctions et d'opérateurs spatiaux lui permettant de requêter les modèles et d'effectuer des traitements sur ceux-ci. Il permet également d'indexer les objets géométriques à l'aide d'un index spatial de type R-Tree. La représentation des objets géométriques et les opérateurs spatiaux d'Oracle Spatial sont implémentés suivant la spécification SFS for SQL de l'OGC. Jusqu'à tout récemment, Oracle Spatial, comme ses concurrents, n'était pas conçue à la base pour la gestion des modèles volumiques. Depuis la version 11g, Oracle a mis sur le marché un premier SGBD spatial intégrant une primitive spatiale volumique permettant de stocker des modèles 3D contenant des solides construits à partir d'un assemblage de surface (à la manière du B-Rep) (Oracle, 2008). Un index spatial 3D de type R-Tree assure la performance des requêtes effectuées sur les modèles. Au niveau des opérateurs spatiaux, Oracle a amené au niveau 3D les opérateurs spatiaux métriques comme par exemple le calcul de distance, de superficie et de volume. Toutefois, les opérateurs topologiques ne tiennent toujours pas compte de la troisième dimension. Au niveau du partage des données, Oracle 11g implémente une interface standard de service Web (WFS) pour la distribution des données géométriques sur Internet.

De son côté, Microsoft offre pour la première fois dans sa nouvelle version de SQL Server 2008 (<http://www.microsoft.com/sqlserver/>), la gestion des données géométriques par le biais de ses nouveaux types de donnée Geometry (données planaires) et Geography (données géodésiques). Comme Oracle Spatial, SQL Server permet maintenant de stocker, indexer et effectuer des opérations spatiales sur des données géométriques ponctuelles, linéaires et surfaciques. SQL Server implémente le standard SFS pour la définition des

données et des opérateurs spatiaux et permet l'importation de données sous format GML. Aucune gestion de volume n'est actuellement supportée dans ce produit.

Du côté des logiciels libres, PostGIS (<http://postgis.refractory.net/>) est l'extension spatiale du SGBD gratuit et libre PostgreSQL. D'une manière générale, PostGIS offre les mêmes possibilités côté stockage, analyse et indexation que ses concurrents Oracle Spatial, SQL Server, etc. sans toutefois le support d'une primitive volumique. Il est aussi certifié OGC pour son implémentation du SFS pour la représentation des données géométriques (x,y,z) et opérateurs spatiaux.

Autres technologies 3D: Plateformes et langages de développement Web

Il existe aussi d'autres géotechnologies ou langages servant de base pour le développement d'applications spatiales avec certaines capacités 3D.

Geotools (<http://geotools.codehaus.org/>), par exemple, est une bibliothèque Java open source permettant la manipulation de données géométriques et qui implémente le SFS, les principales spécifications de l'OGC liées aux services web (WMS, WFS, WCS) ainsi que le schéma spatial ISO 19107 (en partie). Geotools met à disposition un grand nombre de connecteurs lui permettant de lire et écrire différents formats de données spatiales comme le Shapefile ou le GML et de se brancher aux principaux SGBD ou SIG. Comme Geotools implémente des standards supportant le 3D comme ISO 19107 et GML, ce produit offre un potentiel intéressant pour le développement d'applications spatiales 3D. Toutefois, seul le profil « *Simple Features* » de ces standards a été pour l'instant implémenté.

GeoServer (<http://geoserver.org>) se définit principalement comme un serveur de données transactionnel qui implémente les services Web standards de l'OGC (WFS-T et WMS) en s'appuyant sur Geotools. GeoServer peut donc servir de serveur d'applications dans un système destiné au partage et à l'analyse de données spatiales de manière standard sur le Web, et ce, dans un contexte d'interopérabilité. Comme Geotools, GeoServer offrira un potentiel 3D intéressant une fois complétée l'implémentation du GML (version 3.x) et du schéma spatial ISO 19107.

Deegree (<http://www.deegree.org/>) est un autre produit open source, basé sur Java, offrant plus ou moins les mêmes possibilités que GeoServer et qui repose sur les normes et spécifications de l'ISO et de l'OGC. Le schéma spatial ISO 19107 qu'il implémente n'est pas non plus complété mais devrait, à terme, offrir des capacités 3D par la gestion des solides et la lecture et l'écriture du format GML 3.x.

IONIC Software (maintenant Erdas) avec sa suite RedSpider (<http://www.ionicsoft.com/>) offre une gamme de produits commerciaux développés en Java et ayant des capacités similaires à GeoServer et Deegree permettant le développement d'applications spatiales sur le Web conforme aux spécifications de l'ISO et de l'OGC (encore une fois, support du 2D seulement).

La spécification Extensible 3D (X3D) du Web3D Consortium (W3D) (<http://www.web3d.org/x3d/>), qui est devenue une norme ISO en 2005, définit une syntaxe XML pour la représentation et la visualisation de contenu 3D sur le Web. Le X3D est le successeur du VRML (Virtual Reality Modeling Language). Il supporte des objets ayant des géométries variées telles des points, arcs, surfaces, cubes, sphères, extrusions, etc.

Finalement, mentionnons aussi Open Cascade Technology (<http://www.opencascade.org/>) qui offre une plate-forme de développement de logiciels gratuite et Open Source offrant des capacités de modélisation (surfactive et volumique) et de visualisation 3D.

Annexe E Inventaire des géotechnologies : Justification détaillée du tableau comparatif

Possible

Ce qualificatif indique que la catégorie de logiciel permet de réaliser la fonctionnalité mentionnée. Par exemple, on peut remarquer que :

- Les outils CAO, DAO, SIG et visualisateurs de globe virtuel permettent la visualisation 3D. Les outils CAO, DAO, visualisateurs de globes virtuels et les SIG permettent aussi de créer des modèles 2.5D.

- Seuls les outils CAO et DAO permettent la modélisation volumique.
- Les SIG respectent en général les principaux standards en géomatique ou offrent des modules connexes pour y parvenir.
- Les visualisateurs de globes virtuels permettent la diffusion de modèles 3D superposés à la surface du globe.

Partiellement possible

Ce qualificatif indique que la catégorie de logiciel permet de réaliser la fonctionnalité mentionnée, mais seulement de manière partielle. Par exemple, on peut mentionner que :

- Les outils CAO permettent de réaliser de l'analyse spatiale 3D mais offrent uniquement des opérateurs spatiaux métriques et aucun topologique.
- Les SGBD, et du fait même les SIG (par le biais des SGBD auxquels ils se connectent) permettent la gestion de données ponctuelles, linéaires et surfaciques à coordonnées x,y,z mais ils n'ont pas été conçus (jusqu'à présent) pour gérer les primitives volumiques (à l'exception de la dernière version d'Oracle Spatial soit la 11g). Cependant, les récentes activités de R&D tendent à démontrer que nous allons dans cette direction (Arens *et al.*, 2003; Apel, 2004; Lachance, 2005).
- Les SIG permettent la diffusion de données spatiales sur Internet par le biais de services Web, mais cette diffusion se limite aux modèles 2D ou 2.5D.
- Les DAO offrent généralement des fonctionnalités pour animer les modèles 3D et produire des séquences vidéo. Celles-ci peuvent facilement être publiées et consultées en ligne. Les modèles 3D peuvent aussi être utilisés au sein d'un environnement virtuel comme un jeu vidéo ce qui peut être considéré comme une forme de diffusion. Ces types de diffusion ne sont toutefois pas opérés dynamiquement par le DAO lui-même, mais requièrent plutôt l'usage d'une tierce application pour leur diffusion en ligne.
- Les SGBD spatiaux implémentent le SFS for SQL comme standard pour le stockage et l'analyse des données. Ce standard n'est toutefois pas aussi complet que le schéma spatial ISO 19107 et se limite aux primitives géométriques 2D. Les SGBD sont aussi limités côté spécification de services Web (à l'exception de la dernière version d'Oracle Spatial soit la 11g implémentant la spécification WFS).
- Les visualisateurs de globe virtuel n'implémentent pas ou très peu le standard GML pour l'ajout de modèles 3D sur le globe, mais utilisent plutôt le format KML. Toutefois, étant donné sa grande popularité, le KML a récemment été adopté comme standard par l'OGC. Aussi, ils permettent généralement la superposition de données externes sur leur globe par le biais de la spécification WMS, mais uniquement pour les modèles 2D.

**Non
disponible**

Ce qualificatif indique que la catégorie de logiciel n'offre pas la fonctionnalité mentionnée. Ce qualificatif peut aussi servir à indiquer que cette catégorie de logiciel n'est pas destinée à offrir cette fonctionnalité. Par exemple, on peut mentionner que :

- Les SGBD spatiaux ne permettent pas et ne sont pas destinés à la visualisation ni à la modélisation géométrique que ce soit au niveau 2D ou 3D.
- Aucune des catégories de logiciel ne permet de réaliser de l'analyse spatiale 3D (à l'exception des outils CAO qui offrent quelques opérateurs métriques 3D).
- Les SIG et les visualisateurs de globe virtuel ne permettent pas la modélisation volumique. (Ces derniers ont recourt à des produits externes de type CAO-DAO pour la création des modèles 3D qu'ils affichent).
- Les outils CAO, DAO et les visualisateurs de globe virtuel ne permettent pas la gestion de données 3D au sein de bases de données, ils se limitent à l'usage de fichiers (d'où leur difficulté à rattacher des propriétés descriptives aux entités géométriques comme le permettent les SIG).
- Les outils CAO et les SGBD spatiaux ne permettent pas la diffusion de données géométriques 3D.
- Les outils CAO et DAO n'implémentant pas les standards de l'ISO et de l'OGC et sont ainsi difficilement extensibles et interopérables.

Annexe F Extrait du code source des classes ajoutées ou modifiées pour gérer le 3D dans Deegree

Cette annexe présente l'extrait du code source de quelques-unes des principales classes qui ont été créés ou modifiées afin de permettre la gestion des primitives volumiques dans Deegree. Les classes suivantes sont présentées :

- SolidImpl
- ShellImpl
- GeometryFactory
- GMLGeometryAdapter
- GenericSQLWhereBuilder

Implémentation de l'interface d'un Solid et d'un Shell ISO 19107.

```
/**
 * Implementation of the Solid interface
 * Basis for 3-dimensional geometry. The extent of a solid is
 * defined by the boundary surfaces.
 */
public class SolidImpl extends PrimitiveImpl implements Solid, Serializable {

    private static final ILogger LOG = LoggerFactory.getLogger( SolidImpl.class );
    private double area = 0;
    public SolidImpl( SolidBoundary boundary )
        throws GeometryException {

        super( boundary.getCoordinateSystem());
        this.boundary = boundary;
    }

/**
 * Implementation of the Shell interface
 * Represents a single connected component of a solid boundary.
 * A shell consists of a number of references to orientable surfaces
 * connected in a topological cycle (an object whose boundary is
 * empty). Unlike a ring, a Shell's elements have no natural sort
 * order. Like rings, Shells are simple.
 *
 * Note: A Cycle has no boundary because it closes on itself, but
 * it is bounded (i.e. it does not have an infinite extent).
 * For example, a circle has no boundary but it is bounded.
 */
public class ShellImpl extends CompositeSurfaceImpl implements Shell {
    private CompositeSurface csur = null;
    public ShellImpl(CompositeSurface parent, CoordinateSystem crs)
```

```

    throws GeometryException {super( parent, crs);
    setCompositeSurface(parent);
}

/**
 *Set the Shell as compositeSurface.
 */
protected void setCompositeSurface(CompositeSurface parent){
    this.csur = parent;
}

/**
 *Always returns true since shell objects are simples.
 */
public boolean isSimple(){
    return true;
}

/**
 * Calculates the centroid and the envelope of the Shell.
 */
protected void calculateParam() {
    //calculateCentroid();
    calculateEnvelope();
    setValid( true );
}

/**
 * Calculates the envelope
 */
private void calculateEnvelope() {
    Set el = this.getElements();
    double[] coormin = new double[3];
    double[] coormax = new double[3];

    Iterator it = el.iterator();
    Envelope e = ((Surface)it.next()).getEnvelope();
    coormin[0] = e.getMin().getX();
    coormin[1] = e.getMin().getY();
    coormin[2] = e.getMin().getZ();
    coormax[0] = e.getMax().getX();
    coormax[1] = e.getMax().getY();
    coormax[2] = e.getMax().getZ();
    while (it.hasNext()) {
        e = ((Surface)it.next()).getEnvelope();
        if (e.getMin().getX() < coormin[0]) {coormin[0] = e.getMin().getX();}
        if (e.getMin().getY() < coormin[1]) {coormin[1] = e.getMin().getY();}
        if (e.getMin().getZ() < coormin[2]) {coormin[2] = e.getMin().getZ();}
        if (e.getMax().getX() > coormax[0]) {coormax[0] = e.getMax().getX();}
        if (e.getMax().getY() > coormax[1]) {coormax[1] = e.getMax().getY();}
        if (e.getMax().getZ() > coormax[2]) {coormax[2] = e.getMax().getZ();}
    }
    envelope = new EnvelopeImpl( new PositionImpl( coormin ), new PositionImpl( coormax ) );
}
}

```

Implémentation de la classe GeometryFactory pour la construction des primitives 3D du schéma ISO 19107.

```

/**
 * Factory to create geometry instances.
 */
public final class GeometryFactory {
...

/**
 * Creates a <tt>CompositeSurface</tt> from a <tt>OrientableSurface array</tt>.
 * <p>
 * @param surface[] an array of OrientableSurface
 * @param crs spatial reference system of the CompositeSurface
 * @return corresponding CompositeSurface
 *
 * @throws GeometryException
 */
public static CompositeSurface createCompositeSurface (OrientableSurface[] os, CoordinateSystem crs)
throws GeometryException {
    CompositeSurface cs = new CompositeSurfaceImpl(null, crs );
    Set el = cs.getElements();
    for (int i=0;i< os.length;i++){
        el.add(os[i]);
    }
    return cs;
}

/**
 * Creates a CompositeSolid from a Solid array.
 *
 * @param solid[] an array of Solid
 * @param crs spatial reference system of the CompositeSolid
 * @return corresponding CompositeSolid
 *
 * @throws GeometryException
 */
public static CompositeSolid createCompositeSolid (Solid[] s, CoordinateSystem crs)
throws GeometryException {

    CompositeSolid cs = new CompositeSolidImpl(null, crs );
    Set el = cs.getElements();
    for (int i=0;i< s.length;i++){
        el.add(s[i]);
    }
    return cs;
}

/**
 * Creates a Solid from a SolidBoundary.
 *
 * @param boundary a SolidBoundary
 * @return correspondingSolid
 *

```

```

* @throws GeometryException
*/
public static Solid createSolid (SolidBoundary boundary) throws GeometryException {
    Solid s = new SolidImpl(boundary);
    return s;
}

/**
 * Creates a SolidBoundary from an interior & exterior shells.
 * @param exterior a Shell
 * @param interior an array of Shell
 * @return corresponding SolidBoundary
 *
 * @throws GeometryException
 */
public static SolidBoundary createSolidBoundary (Shell exterior, Shell[] interior )
    throws GeometryException {
    System.out.println("GeoFactory: createSolidBoundary");
    SolidBoundary sb = new SolidBoundaryImpl( exterior, interior );
    return sb;
}

/**
 * Creates a Shell from a interior & exterior shells.
 * @param exterior a Shell
 * @param interior an array of Shell
 * @return corresponding SolidBoundary
 *
 * @throws GeometryException
 */
public static Shell createShell (OrientableSurface[] os, CompositeSurface parent, CoordinateSystem crs )
    throws GeometryException {
    System.out.println("GeoFactory: createShell");
    Shell s = new ShellImpl( parent, crs );

    Set el = s.getElements();
    System.out.println("nb de OrientableSurface: " + Integer.toString(os.length));
    for (int i=0;i< os.length;i++){
        el.add(os[i]);
    }
    return s;
}
}

```

Implémentation de la classe GMLGeometryAdapter pour la conversion des primitives 3D entre leur représentation ISO 19107 et GML.

```

/**
 * Adapter class for converting GML geometries to deegree geometries and vice versa. Some logical
 * problems results from the fact that an envelope isn't a geometry according to ISO 19107 (where
 * the deegree geometry model is based on) but according to GML2/3 specification it is.
 * So if the wrap(..) method is called with an envelope a Surface will be returned
 * representing the envelops shape. To export an Envelope to a GML box/envelope two
 * specialized export methods are available.
 * The export method(s) doesn't return a DOM element as one may expect but a StringBuffer.
 * This is done because the transformation from deegree geometries to GML mainly is required when a
 * GML representation of a geometry shall be serialized to a file or to a network connection. For
 * both cases the string representation is required. and it is simply faster to create the string
 * directly instead of first creating a DOM tree that after this must be serialized to a string.
 * In future version geometries will be serialized to a stream.
 */
public class GMLGeometryAdapter {
...

    /**
     * @param csur
     * @return
     * @throws RemoteException
     * @throws GeometryException
     */
    private static StringBuffer exportCompositeSurface( CompositeSurface csur )
        throws GeometryException {
        LOG.entering();
        StringBuffer sb = new StringBuffer( 5000 );

        String crs = null;
        if ( csur.getCoordinateSystem() != null ) {
            crs = csur.getCoordinateSystem().getName().replace( ',', ':' );
        }

        if ( crs != null ) {
            sb.append( "<gml:CompositeSurface xmlns:gml=\"http://www.opengis.net/gml\" srsName=\"" + crs +
                "\">" );
        } else {
            sb.append( "<gml:CompositeSurface xmlns:gml=\"http://www.opengis.net/gml\" >" );
        }
        Set set = csur.getElements();
        Iterator it = set.iterator();

        while (it.hasNext()) {
            // Get element
            sb.append( "<gml:surfaceMember>" );
            sb.append( "<gml:OrientableSurface orientation=\"" + "\">" );
            sb.append( "<gml:baseSurface>" );

            Surface s = (Surface)it.next();
            sb.append(exportSurface(s));
            sb.append( "</gml:baseSurface>" );
        }
    }
}

```

```

        sb.append( "</gml:OrientableSurface>" );
        sb.append( "</gml:surfaceMember>" );
    }

    sb.append( "</gml:CompositeSurface>" );
    LOG.exiting();
    return sb;
}

/**
 * @param sol
 * @return
 * @throws RemoteException
 * @throws GeometryException
 */
private static StringBuffer exportSolid( Solid sol )
    throws GeometryException {
    LOG.entering();
    StringBuffer sb = new StringBuffer( 5000 );

    String crs = null;
    if ( sol.getCoordinateSystem() != null ) {
        crs = sol.getCoordinateSystem().getName().replace( ' ', ':' );
    }
    if ( crs != null ) {
        sb.append( "<gml:Solid xmlns:gml=\"http://www.opengis.net/gml\" srsName=\"" + crs + "\">" );
    } else {
        sb.append( "<gml:Solid xmlns:gml=\"http://www.opengis.net/gml\">" );
    }
}

    SolidBoundary solb = sol.getBoundary();

    Shell exter = solb.getExterior();
    Shell[] inter = solb.getInteriors();
    sb.append( "<gml:exterior>" );
    //exter.getElements();
    CompositeSurface cse = (CompositeSurface)exter;
    sb.append(exportCompositeSurface(cse));
    sb.append( "</gml:exterior>" );

    if (inter != null){
        for (int i=0;i<inter.length - 1;i++){
            sb.append( "<gml:interior>" );
            CompositeSurface csi = (CompositeSurface)inter[i];

            sb.append(exportCompositeSurface(csi));
            sb.append( "</gml:interior>" );
        }
    }
    sb.append( "</gml:Solid>" );

    LOG.exiting();
    return sb;
}

/**

```



```

* @param csol
* @return
* @throws RemoteException
* @throws GeometryException
**/
private static StringBuffer exportCompositeSolid( CompositeSolid csol )
    throws GeometryException {
    LOG.entering();
    StringBuffer sb = new StringBuffer( 5000 );

    String crs = null;
    if ( csol.getCoordinateSystem() != null ) {
        crs = csol.getCoordinateSystem().getName().replace( ',', ':' );
    }

    if ( crs != null ) {
        sb.append( "<gml:CompositeSolid xmlns:gml=\"http://www.opengis.net/gml\" srsName=\"" + crs +
"\">" );
    } else {
        sb.append( "<gml:CompositeSolid xmlns:gml=\"http://www.opengis.net/gml\" >" );
    }

    Set set = csol.getElements();
    Iterator it = set.iterator();

    while ( it.hasNext() ) {
        // Get element
        sb.append( "<gml:solidMember>" );

        Solid s = (Solid)it.next();
        sb.append(exportSolid(s));
        sb.append( "</gml:solidMember>" );
    }

    sb.append( "</gml:CompositeSolid>" );
    LOG.exiting();
    return sb;
}
}

/**
* Returns an instance of a Solid created from the passed <gml:Solid>
*
* @param element
*     <gml:Solid>
*
* @return instance of Solid
* @throws XMLParsingException
**/
private static Solid wrapSolid( Element element )
    throws XMLParsingException, GeometryException, InvalidGMLException {
    LOG.entering();
    String srs = XMLTools.getAttrValue( element, "srsName" );
    CoordinateSystem crs = null;
    if ( srs != null ) {
        crs = getCRS( srs );
    }
}

```

```

    }

    List ext = XMLTools.getNodes( element, CommonNamespaces.GML_PREFIX + ":exterior", nsContext
);
    List cs = XMLTools.getNodes((Element)ext.get(0), CommonNamespaces.GML_PREFIX +
":CompositeSurface", nsContext );
    List sm = XMLTools.getNodes((Element)cs.get(0), CommonNamespaces.GML_PREFIX +
":surfaceMember", nsContext );

    OrientableSurface[] os = new OrientableSurface[sm.size()];
    for (int i=0;i<sm.size();i++)
    {
        List lst_os = XMLTools.getNodes((Element)sm.get(i), CommonNamespaces.GML_PREFIX +
":OrientableSurface", nsContext );
        List lst_bs = XMLTools.getNodes((Element)lst_os.get(0), CommonNamespaces.GML_PREFIX +
":baseSurface", nsContext );
        List poly = XMLTools.getNodes((Element)lst_bs.get(0), CommonNamespaces.GML_PREFIX +
":Polygon", nsContext );

        Surface surface = wrapPolygon((Element)poly.get(0));
        os[i] = surface;
    }
    Shell shell = GeometryFactory.createShell( os, null, crs);

    SolidBoundary sb = GeometryFactory.createSolidBoundary( shell, null);
    Solid s = GeometryFactory.createSolid( sb );

    LOG.exiting();
    return s;
}
/**
 * Returns an instance of a CompositeSolid created from the passed <gml:CompositeSolid>
 *
 * @param element
 *         <gml:CompositeSolid>
 *
 * @return instance of CompositeSolid
 * @throws XMLParsingException
 */
private static CompositeSolid wrapCompositeSolid( Element element )
        throws XMLParsingException, GeometryException, InvalidGMLException {
    LOG.entering();
    String srs = XMLTools.getAttrValue( element, "srsName" );
    CoordinateSystem crs = null;
    if ( srs != null ) {
        crs = getCRS( srs );
    }
    List sm = XMLTools.getNodes( element, CommonNamespaces.GML_PREFIX + ":solidMember",
nsContext );

    Solid[] s = new Solid[sm.size()];
    for (int i=0;i<sm.size();i++)
    {
        List lst_s = XMLTools.getNodes((Element)sm.get(i), CommonNamespaces.GML_PREFIX +
":Solid", nsContext );

```

```
        Solid solid = wrapSolid((Element)lst_s.get(0));
        s[i] = solid;
    }

    CompositeSolid csol = GeometryFactory.createCompositeSolid( s, crs);
    LOG.exiting();
    return csol;
}
```

Implémentation de la méthode `appendBBOXOperationAsSQL` de la classe `GenericSQLWhereBuilder` pour l'extraction de la base de données des données filtrées avec un BBOX.

```

/**
 * WhereBuilder implementation for Generic SQL datastore.
 * Supports any database that can be connected through JDBC and that
 * offers BLOB typed fields.
 *
 **/
class GenericSQLWhereBuilder extends WhereBuilder {

    ...

    /**
     * Generates an SQL-fragment for the BBOX.
     *
     **/
    private void appendBBOXOperationAsSQL( StatementBuffer query, SpatialOperation operation ) {
        query.append( "(" );
        String tableAlias = getTableAlias( operation.getPropertyName() );
        if ( tableAlias != "" ) {
            tableAlias = tableAlias + ".";
        }
        Envelope env = operation.getGeometry().getEnvelope();
        double minx = env.getMin().getX();
        double miny = env.getMin().getY();
        double maxx = env.getMax().getX();
        double maxy = env.getMax().getY();
        StringBuffer sb = new StringBuffer( 200 );
        if (operation.getGeometry().getCoordinateDimension() == 2){
            sb.append( tableAlias ).append( "minx > " ).append( minx ).append( " and " );
            sb.append( tableAlias ).append( "miny > " ).append( miny ).append( " and " );
            sb.append( tableAlias ).append( "maxx < " ).append( maxx ).append( " and " );
            sb.append( tableAlias ).append( "maxy < " ).append( maxy );
        }else if(operation.getGeometry().getCoordinateDimension() == 3){
            double minz = env.getMin().getZ();
            double maxz = env.getMax().getZ();
            sb.append( tableAlias ).append( "minx > " ).append( minx ).append( " and " );
            sb.append( tableAlias ).append( "miny > " ).append( miny ).append( " and " );
            sb.append( tableAlias ).append( "minz > " ).append( minz ).append( " and " );
            sb.append( tableAlias ).append( "maxx < " ).append( maxx ).append( " and " );
            sb.append( tableAlias ).append( "maxy < " ).append( maxy ).append( " and " );
            sb.append( tableAlias ).append( "maxz < " ).append( maxz );
        }
        query.append( sb.toString() );
        query.append( " ) ); }
    }
}

```