
Une architecture orientée service Web pour la constitution de mini-cubes SOLAP pour clients mobiles

Etienne Dubé^{*,**} — Thierry Badard^{*} — Yvan Bédard^{*,**}

^{*} Centre de recherche en Géomatique (CRG)

^{**} Chaire de recherche industrielle en bases de données géospatiales décisionnelles

Pavillon Louis-Jacques-Casault

Université Laval

Québec (Québec) G1V 0A6

Canada

etienne.dube.2@ulaval.ca, {thierry.badard,yvan.bedard}@scg.ulaval.ca

RÉSUMÉ. Les applications Spatial OLAP (SOLAP) réalisées jusqu'à présent sont conçues pour les environnements informatiques de bureau. Dans le but de supporter l'aide à la décision géospatiale en situation de mobilité, certaines adaptations à SOLAP sont nécessaires. Cet article décrit une solution, basée sur les architectures orientées services et les technologies des services Web, pour adapter les cubes de données SOLAP afin de répondre aux exigences et contraintes reliées aux environnements informatiques mobiles et aux réseaux sans-fil.

ABSTRACT. Existing Spatial OLAP (SOLAP) applications are targeted towards desktop computer environments. In order to enable geospatial decision support in mobility contexts, SOLAP requires some modifications. This paper describes a solution, based on service-oriented architectures and Web services technologies, designed to adapt SOLAP data cubes to the requirements and constraints encountered in mobile computing environments and wireless networks.

MOTS-CLÉS : Spatial OLAP, SOLAP, entrepôt de données géospatiales décisionnelles, informatique mobile sans-fil, service Web, architecture orientée service.

KEYWORDS: Spatial OLAP, SOLAP, geospatial decisional data warehouse, wireless mobile computing, Web service, service-oriented architecture.

1. Introduction

Depuis le milieu des années 1990, une catégorie de systèmes d'aide à la décision existe sur le marché : il s'agit des outils OLAP, pour *On-Line Analytical Processing*, permettant d'effectuer une analyse exploratoire des données de manière rapide, interactive et conviviale. Ces logiciels font partie du domaine appelé Business Intelligence (BI), soit « intelligence d'affaires » ou « informatique décisionnelle », et découlent du besoin des entreprises et organisations d'appuyer les processus de prise de décisions sur un éventail de données et d'analyses les plus complètes et globales que possible. Or, les outils OLAP traditionnels sont limités dans leur capacité à traiter et à représenter l'information géographique. Une nouvelle classe d'outils, regroupant certaines fonctionnalités des systèmes d'information géographiques (SIG) avec celles d'OLAP, a été développée afin de tirer profit de la composante spatiale ; il s'agit du *Spatial OLAP*, ou SOLAP (Bédard *et al.*, 2005).

D'un autre côté, l'informatique mobile a fait des avancées considérables au cours de la dernière décennie : on peut le constater avec l'ubiquité des appareils tels que les téléphones mobiles, ordinateurs portatifs et de poche (PDA) pouvant se connecter aux réseaux sans fils, qu'il s'agisse de téléphonie cellulaire ou de réseaux WiFi. Ainsi, les multiples services offerts via Internet sont désormais accessibles en tout lieu et en tout temps, au creux de la main. En parallèle, l'évolution du commerce électronique et autres systèmes répartis sur Internet ont donné naissance à un paradigme architectural ainsi qu'une série de protocoles de communication : il s'agit des architectures orientées services et des services Web. Ces technologies s'inspirent des principes du Web et des forces du langage XML pour offrir des moyens de communication efficaces entre systèmes n-tiers sur Internet. Dans le monde de la géomatique, on retrouve ces technologies dans les spécifications de l'Open Geospatial Consortium, qui sont utilisées dans le développement d'applications de cartographie sur Internet (*web mapping*) et la mise en oeuvre d'infrastructures de données spatiales.

Les applications SOLAP actuelles sont conçues pour les environnements informatiques de bureau. Afin d'aider à la prise de décision en situation de mobilité, soit sur le terrain (e.g. domaines des urgences, de la sécurité publique et de la défense), il serait souhaitable d'offrir des versions mobiles de ces applications. Or, les environnements informatiques mobiles imposent certaines contraintes : des affichages de taille réduite, des méthodes d'entrées différentes (ex. stylet) ne permettant pas par exemple une saisie facile et rapide de texte, ce qui nécessite de repenser les interfaces et les interactions avec les applications. A cela vient s'ajouter des liens de communication sans-fil au débit réduit, à la latence élevée et d'une couverture incomplète, ainsi que des capacités de stockage et de calcul réduites (Duda *et al.*, 2005). Il en résulte que les applications SOLAP nécessitent certaines adaptations pour répondre à ces contraintes de mobilité. Cet article propose une architecture orientée services Web pour la constitution de mini-cubes de données SOLAP, visant à adapter les données SOLAP aux besoins et contraintes des environnements mobiles, et à assurer leur diffusion vers les clients mobiles de manière interopérable, via des liens de communications sans-fil.

La conception d'une telle solution s'inscrit comme une première étape préalable à la définition et à la mise en oeuvre d'un client SOLAP mobile complet.

L'article se divise comme suit : nous débutons par une revue des concepts mis en oeuvre et de leur problématique par rapport aux contextes mobiles. Par la suite, une architecture orientée service est proposée, avec des explications qui guideront le développement d'un prototype de service Web pour constitution de mini-cubes SOLAP. Finalement, les travaux de recherche à venir ainsi que certaines perspectives futures sont présentées.

2. Concepts et problématique

2.1. Concepts OLAP et d'entrepôt de données décisionnelles

OLAP (*On-Line Analytical Processing*) est défini comme étant « ... le nom donné à l'analyse dynamique requise pour créer, manipuler, animer et synthétiser l'information, par des modèles d'analyse de données exégétiques, contemplatifs et selon des formules » (traduction libre ; (Codd *et al.*, 1993)). En d'autres termes, il s'agit d'applications de modélisation descriptive et d'analyse exploratoire des données, conçues à des fins de prise de décision. Les systèmes OLAP sont optimisés pour l'analyse rapide de grands volumes de données, par opposition aux systèmes transactionnels (ou OLTP ; *On-Line Transactional Processing*). Ces derniers sont plutôt destinés aux opérations courantes d'une organisation (ex. ventes, commandes, réservations, etc.), et sont adaptés à de grands volumes de transactions en lecture et en écriture, chaque transaction individuelle n'affectant qu'une petite quantité de données (Thomsen, 2002). Les outils OLAP s'appuient sur le modèle de données multidimensionnel, où les données sont représentées en termes de dimensions, mesures et faits. Les dimensions constituent des axes thématiques caractérisant les données (ex. le temps, les lieux, les catégories de produits). Une dimension est composée de membres, représentant les occurrences individuelles du thème (ex. « France », « Belgique », « Canada »). Les membres peuvent être organisés selon différents niveaux de détail, de manière hiérarchique. Quant aux mesures, elles représentent les valeurs quantifiables relevées pour les données (ex. la population, le produit intérieur brut). Les combinaisons possibles des dimensions, avec les mesures qui en découlent, forment les faits. Il est possible d'appliquer des fonctions agrégatives (somme, moyenne, médiane, etc.) pour obtenir les mesures à partir des données transactionnelles ou de mesures de membres de niveau inférieur. Ainsi, on peut calculer une valeur pour un fait caractérisé par les membres d'une dimension du niveau hiérarchique inférieur qui s'agrègent vers un membre d'un niveau supérieur (ex. la population du Canada est la somme de la population de chacune de ses provinces). Un jeu de données multidimensionnelles est

nommé « cube » ou « hypercube », en faisant référence à l'organisation des faits selon des axes dimensionnels¹.

De nombreuses applications OLAP font usage d'un entrepôt de données décisionnelles pour le stockage des cubes. Ceux-ci sont généralement basés sur des systèmes de gestion de bases de données (SGBD) relationnels, par exemple des serveurs tels qu'Oracle, Microsoft SQL Server, MySQL AB ou PostgreSQL, ou des SGBD enrichis tels NCR Terradata, Kognitio et Sand Technologies. Les données des cubes sont organisées selon un modèle multidimensionnel (schémas en étoile, en flocon ou parent-enfant), dont les principes diffèrent des modèles relationnels normalisés des systèmes transactionnels (Kimball *et al.*, 2002). Le peuplement et la mise à jour des entrepôts de données décisionnels sont faits depuis diverses sources de données opérationnelles. Des outils nommés ETL (*Extract, Transform, Load*), permettant l'extraction, la transformation et le chargement de données, sont fréquemment employés pour ce faire (Kimball *et al.*, 2004).

Plusieurs produits OLAP disponibles sur le marché offrent des langages d'interrogation adaptés aux données multidimensionnelles ; par exemple le langage MDX (*Multidimensional eXpressions*), introduit par Microsoft avec SQL Server Analysis Services (Microsoft Corporation, 2007a) mais aussi employé par d'autres produits tels que Hyperion Essbase et le serveur OLAP *open source* Mondrian (Pentaho Corporation, 2007). De tels langages sont aux données multidimensionnelles ce que le langage d'interrogation SQL est aux données relationnelles. Des clients OLAP présentant une interface utilisateur conviviale existent également. Ceux-ci offrent une vue des données multidimensionnelles sous forme de tableaux, graphiques et rapports, et permettent la navigation visuelle dans les cubes de données avec des opérateurs tels que le forage et remontage (*drill-down, roll-up*), sans avoir à connaître un langage d'interrogation des données.

2.2. Spatial OLAP : d'un environnement sédentaire jusqu'à la mobilité

Le OLAP spatial, ou SOLAP, est le fruit de recherches visant à combiner les technologies OLAP et SIG à l'intérieur d'un outil intégré d'aide à la décision spatiale (Proulx *et al.*, 2002; Rivest *et al.*, 2001). Un outil SOLAP est défini comme étant « un type de logiciel qui permet la navigation rapide et facile dans les bases de données spatiales et qui offre plusieurs niveaux de granularité d'information, plusieurs thèmes, plusieurs époques et plusieurs modes d'affichage synchronisés ou non : cartes, tableaux et diagrammes » (Bédard *et al.*, 2005). En somme, SOLAP étend OLAP avec la gestion des données (i.e. dimensions et mesures spatiales) et des analyses spatiales ainsi que la visualisation et la navigation cartographique. L'implémentation d'un outil SOLAP intégré a été réalisé par l'équipe de la Chaire de recherche industrielle

1. Notons que dans ce sens, un « cube » n'est pas restreint à trois dimensions : il peut y avoir autant de dimensions que les thèmes d'analyse le requièrent, le terme hypercube étant alors parfois plus usité pour décrire ces cas.

en bases de données géospatiales décisionnelles (Université Laval, Québec, Canada) avant le début de celle-ci et est commercialisé par la société KHEOPS Technologies ; il s'agit de JMap Spatial OLAP (KHEOPS Technologies, 2005). Ce logiciel, ayant été développé pour des ordinateurs de bureau dans un contexte sédentaire, requiert plusieurs adaptations pour pouvoir être utilisé dans un contexte mobile. Entre autres, les cubes de données typiquement déployés sont volumineux, et surtout ils n'ont typiquement pas été conçus pour satisfaire les besoins changeants des utilisateurs mobiles et de leur technologie. Son architecture 3-tiers (client/serveur SOLAP et serveur SGBD) est également peu adaptée pour les appareils mobiles, puisqu'elle nécessite en tout temps la disponibilité d'un lien réseau à haut débit, situation qui n'est pas garantie en contexte de mobilité.

(Badard *et al.*, 2008) présentent et analysent les critères d'adaptation d'un outil SOLAP (en l'occurrence JMap Spatial OLAP) pour être utilisé dans un environnement mobile. Il en ressort qu'un client SOLAP mobile devrait supporter deux modes de fonctionnement : le mode connecté et le mode déconnecté. En mode connecté, chaque requête de l'utilisateur est acheminée vers le serveur SOLAP et traitée par ce dernier. Le résultat de la requête, provenant de cubes stockés dans un entrepôt de données, est ensuite retourné au client. Ce mode de fonctionnement est donc similaire à celui des environnements de bureau. Il a comme avantage de mettre à disposition toute la puissance de traitement du serveur SOLAP et la capacité de stockage d'un entrepôt de données, mais par contre, il oblige le client à disposer en tout temps d'un lien réseau sans-fil de débit suffisant. Le mode déconnecté, quant à lui, permet au client de fonctionner lorsqu'il n'y a aucun lien réseau disponible. Ce mode s'appuie sur la notion de mini-cube, soit un cube de données SOLAP de taille réduite, dont le contenu a été sélectionné par l'utilisateur sur demande, et pouvant être stocké à même la mémoire non-volatile de l'appareil mobile. Le mini-cube est transmis au client mobile lorsqu'un lien réseau est disponible ; le temps de transmission d'un mini-cube se trouve réduit par rapport à celui de cubes de données entiers, ce qui est important avec les réseaux sans-fil, en particulier les liens de téléphonie mobile au débit limité et dont le coût d'utilisation est parfois élevé. Une fois que le client a obtenu un mini-cube, il peut consulter celui-ci de manière totalement autonome, même sans connexion réseau. Le service Web de constitution de mini-cubes, proposé et décrit dans ce présent article, offre l'infrastructure visant à supporter le mode de fonctionnement déconnecté ; il doit donc offrir les fonctionnalités requises pour réduire le volume des données et ainsi produire les mini-cubes. Par souci d'interopérabilité, l'accès à ces fonctionnalités est basé sur une architecture dite orientée service, s'appuyant sur la technologie des services Web du W3C.

2.3. Principes des architectures orientées services et des services Web

2.3.1. Architectures orientées services

Une architecture orientée services est caractérisée par une méthodologie de conception des systèmes informatiques organisationnels, centrée sur la notion de ser-

vice (Newcomer *et al.*, 2004). Un service est décrit comme un ensemble de fonctionnalités accomplissant des tâches spécifiques, accessible par un réseau informatique. La définition de l'ensemble de ces fonctionnalités et la manière spécifique de les invoquer est formalisée par un *contrat de service* ; ce concept est semblable à celui d'interface dans le vocabulaire du modèle orienté objet. Les services peuvent être combinés les uns avec les autres afin de réaliser des tâches plus complexes. Les composants d'une architecture orientée service se classent en trois catégories : les fournisseurs de services, les consommateurs de services et les médiateurs. Les fournisseurs implémentent les fonctionnalités décrites par le contrat de service. Les consommateurs invoquent les services offerts par les fournisseurs ; un consommateur peut être soit une application client, soit un service s'appuyant sur la fonctionnalité d'un autre fournisseur de service. Les médiateurs ont deux rôles principaux : ils facilitent d'abord la découverte et l'invocation de services, en présentant des catalogues de contrats de services en un lieu centralisé. Ils peuvent également effectuer le routage et le chaînage entre services en fonction de la tâche à réaliser. Par leurs caractéristiques, les architectures orientées services favorisent la réutilisation de composants, et ainsi la réduction des efforts et coûts de développement de systèmes logiciels complexes.

2.3.2. Services Web W3C

Les services Web, tels que définis par le World Wide Web Consortium (W3C), constituent une implémentation possible des architectures orientées services. Il s'agit de services accessibles sur Internet ou par l'intermédiaire de réseaux privés (intranet), utilisant une norme d'échange de messages basée sur XML (normes SOAP et WSDL), et qui ne sont pas liés à une plateforme logicielle (système d'exploitation, langage de programmation) et matérielle (jeu d'instruction du processeur) particulière (Cerami, 2002). Cette caractéristique favorise l'interopérabilité entre systèmes informatiques hétérogènes. Par exemple, un service Web réalisé en langage Java, s'exécutant sur un serveur Linux, peut sans problème être exploité par un logiciel client basé sur l'environnement .NET sous Windows. Puisqu'il s'agit de services qui communiquent par protocoles réseaux standardisés, ils peuvent être déployés sur tout réseau informatique, indépendamment de la technologie (réseaux câblés, sans-fil, etc.) ce qui les rendent idéaux pour les systèmes répartis (*n-tiers*) et l'utilisation mobile. De plus, ils gèrent efficacement les échanges d'informations asynchrones, ce qui est idéal lorsqu'un lien réseau subit des déconnexions et connexions successives, situation fréquemment rencontrée en contexte mobile (dû à une couverture discontinue des réseaux de téléphonie mobile ou WiFi). En comparaison, les protocoles client/serveur classiques requièrent une connexion ininterrompue pendant toute la durée de la session d'utilisation.

2.3.3. Services Web géomatiques OGC

Dans le monde de la géomatique, l'Open Geospatial Consortium (OGC) propose plusieurs spécifications de services Web. Entre autres, les services WMS (*Web Mapping Service*) et WFS (*Web Feature Services*) définissent des mécanismes respectivement de diffusion de cartes (i.e. de représentations de données spatiales) et d'échanges

de données spatiales. WFS, plus particulièrement, permet de servir des entités spatiales vectorielles individuelles, d'une manière transactionnelle (opérations sélection — insertion — mise à jour — suppression) (OGC, 2005). Les données vectorielles transmises sont représentées en format GML (*Geography Markup Language*), qui spécifie un encodage XML des primitives géométriques telles que points, lignes et polygones (ISO/TC211, 2004). WFS permet ainsi une manipulation des données géométriques à la source des cartes. Cependant, son modèle transactionnel convient davantage aux données spatiales tabulaires, composées d'enregistrements (rangées), et non aux données multidimensionnelles qu'on retrouve dans les cubes SOLAP.

2.3.4. Services Web pour entrepôts de données multidimensionnelles et OLAP

Dans le monde des bases de données multidimensionnelles et OLAP, la spécification *XML for Analysis (XMLA)* propose un mécanisme d'interrogation des données multidimensionnelles via un service Web basé sur le protocole SOAP (Microsoft *et al.*, 2002). Cette spécification est basée sur le langage d'interrogation mdXML (lui-même basé sur le langage MDX, cf. 2.1) et est inspiré du modèle objet OLE DB for OLAP (Microsoft Corporation, 2007b). Il est souvent lié à SQL Server Analysis Services, le serveur OLAP de Microsoft. Cependant, d'autres implémentations, telles que Mondrian, ont adopté XMLA comme interface de programmation, démontrant ainsi son interopérabilité (Pentaho Corporation, 2007).

D'un autre côté, un format XML d'échange de données multidimensionnelles a été proposé ; il s'agit de XCube (Hümmer *et al.*, 2003). Il définit une série de schémas XML pour encoder les objets des jeux de données multidimensionnelles, tels que les cubes, dimensions, hiérarchies, niveaux, membres et mesures. Bien qu'il ne s'agisse pas d'une spécification de service Web proprement dit, les éléments XML proposés dans XCube peuvent être encapsulés à l'intérieur de messages SOAP lors de la réalisation d'un service Web.

Dans leur version actuelle, ces deux spécifications s'avèrent insuffisantes pour diffuser des cubes de données spatiales multidimensionnelles. XMLA propose un mécanisme d'interrogation des données, mais son format de livraison de données est conçu pour la présentation d'un résultat de requête (exploitable par exemple pour la constitution de différents affichages : histogramme, camembert, tableau croisé), et non la transmission de cubes complets (comprenant structure et données). XCube répond mieux à ce critère, mais tout comme XMLA, il ne propose pas de solution pour l'échange des entités géographiques (comprenant des primitives géométriques) des cubes de données SOLAP. Le service Web de constitution de mini-cubes reposera donc sur un nouveau format XML d'échange de cubes, capable de représenter des cubes complets contenant également des entités (membres) géographiques.

3. Proposition d'une architecture pour le service Web

La solution proposée, soit le service Web de constitution de mini-cubes, s'articule autour de deux concepts : celui du mini-cube (cf. 2.2) visant à répondre aux contraintes

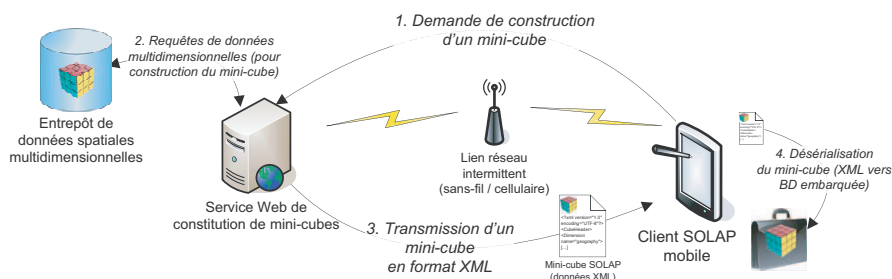


Figure 1. Mécanisme d'invocation du service Web

des appareils mobiles en termes de capacité mémoire (taille réduite des cubes) et en connectivité réseau (support du mode déconnecté), ainsi que celui des services Web, visant à apporter une solution interopérable bien adaptée aux liens de communication sans-fil. Cette section décrit l'architecture générale de ce service et les nouveaux concepts développés dans le cadre de ce projet de recherche.

3.1. Cas d'utilisation : invocation du service

Tel que décrit précédemment, le service Web de constitution de mini-cubes SOLAP vise à supporter un mode de fonctionnement déconnecté pour les clients mobiles. Un cas d'utilisation typique d'un tel service est illustré à la figure 1.

L'invocation du service se déroule comme suit :

1) Le client SOLAP mobile invoque à distance le service Web de constitution de mini-cubes. Le client peut faire plusieurs requêtes successives pour obtenir de l'information (métadonnées) sur les cubes disponibles et les objets qui en font partie (i.e. dimensions, hiérarchies, niveaux, membres et mesures). À partir de ces informations, le client peut choisir un sous-ensemble des données disponibles (soit un sous-cube) et invoquer le service pour demander la construction du mini-cube correspondant à ce sous-ensemble de données.

2) Le service Web traite la demande de construction d'un mini-cube en effectuant la sélection des données souhaitées dans l'entrepôt de données spatiales multidimensionnelles.

3) Le mini-cube est construit à partir des données sélectionnées, sérialisé en un format d'échange XML (cf. sections 2.3.4 et 3.3) et transmis au client, dans un ou plusieurs messages SOAP de réponse.

4) Le client, lors de la réception du mini-cube XML, peut désérialiser ce dernier pour stockage éventuel dans une base de données embarquée (de structure relationnelle, i.e. ROLAP, ou multidimensionnelle, i.e. MOLAP). La désérialisation du cube

et l'usage d'une BD embarquée s'explique par le fait que le mini-cube XML représente un format interopérable d'échange de données, mais n'est pas optimisé pour un stockage compact et un accès rapide aux données par le moteur OLAP.

Une fois le mini-cube stocké sur le client mobile, ce dernier peut fonctionner en mode déconnecté, de manière autonome : l'utilisateur peut donc consulter le cube sans être connecté au service Web ou à tout autre serveur (cf. 2.2). Comme la quantité de données du mini-cube est réduite, les opérations courantes d'analyse SOLAP s'exécutent dans un temps acceptable, malgré les limites en puissance de calcul et en mémoire des appareils mobiles. Des analyses plus complexes, requérant de plus grands volumes de données ainsi qu'une plus grande capacité de traitement, pourraient tout de même s'effectuer en les déléguant à un autre service Web dans une architecture élargie, lorsqu'un lien réseau est disponible.

3.2. Construction d'un mini-cube : principes conceptuels

Les cubes de données SOLAP disponibles dans les entrepôts de données spatiales décisionnelles sont souvent d'une taille considérable, ce qui demande une capacité de stockage bien supérieure à ce qu'offre les supports de mémoire non-volatile des appareils mobiles (ex. cartes mémoire flash, micro disques durs). De plus, les utilisateurs mobiles ont souvent besoin de données propres au contexte dans lequel ils se situent ; alors qu'un entrepôt de données décisionnelles d'une organisation doit offrir les données pouvant répondre à un éventail très varié de requêtes provenant de nombreux utilisateurs, l'utilisateur mobile est appelé à consulter un jeu de données d'une envergure plus restreinte. Il peut s'agir d'un sous-ensemble des données disponibles, selon des thèmes, des époques, des lieux, ou des niveaux hiérarchiques de dimension sélectionnés. Le niveau de détail géométrique des membres de dimensions spatiales et des mesures spatiales peut également être diminué, par exemple par généralisation cartographique, pour réduire l'espace de stockage et le temps de transmission.

Un sous-ensemble d'un cube est sélectionné en effectuant une restriction sur les membres des dimensions ; cette restriction a également pour effet de réduire le nombre de cellules (faits) inclus dans le sous-cube résultant. Prenons le cube SOLAP illustré à la figure 2, représentant les ventes de Boutisport, une entreprise fictive spécialisée dans les articles de sport.

Chaque dimension comporte une seule hiérarchie. La dimension *Commerce* est une dimension spatiale géométrique, avec une représentation cartographique des membres à chaque niveau (polygones représentant le découpage territorial pour les niveaux *Pays*, *Province*, *Région* et *Ville*, et points représentant la position de chaque point de vente pour le niveau *Boutique*). La dimension *Temps* est temporelle, avec les niveaux *Année* — *Mois* — *Jour*. La dimension *Produits* est thématique ; on y représente les divers produits vendus avec les niveaux *Département* (e. g. membres *vélo*, *ski*, *golf*, etc.) — *Catégorie* (membres *bicyclettes*, *pièces*, *accessoires*, *vêtements*, etc. remontant au département *vélo*) — *Produit* (membres *casque*, *porte-bagages*, *pompe à pneus*, etc.

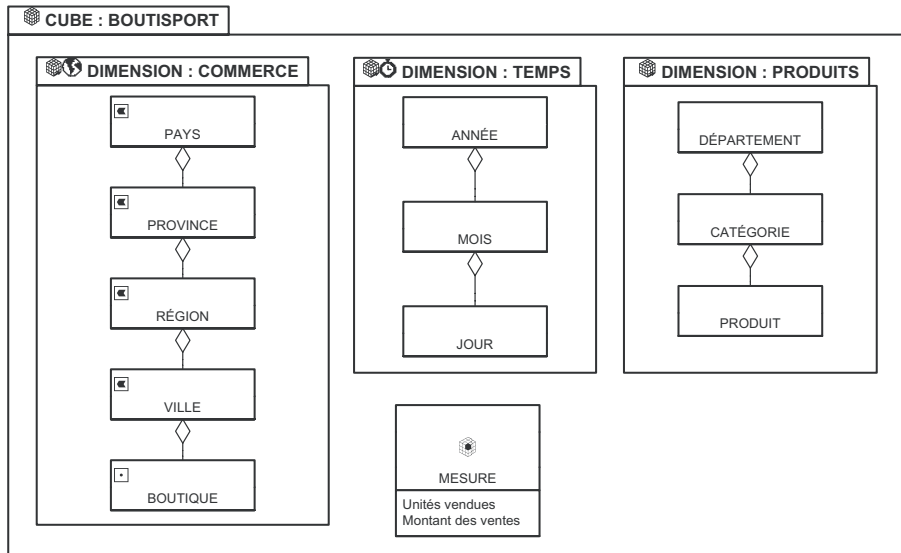


Figure 2. Schéma du cube (formalisme UML multidimensionnel de Bédard et al. (2006))

remontant à la catégorie *Accessoires*). Les faits ont comme granularité les ventes quotidiennes pour chaque produit dans chaque boutique. Les mesures décrivant ces faits sont le nombre d'unités vendues et le montant des ventes.

Les opérations suivantes sont offertes par le service Web pour la sélection d'un sous-cube :

Inclusion de membres pour une dimension : le client peut choisir les membres voulus, dans une ou plusieurs dimensions, pour réduire l'envergure du mini-cube à construire. Par défaut, le membre « Tous » est sélectionné, ce qui revient à inclure tous les membres de la dimension. La sélection d'un membre donné implique également la sélection de tous ses descendants ; ultimement, les membres au niveau feuille remontant au membre choisi sont sélectionnés. *Exemple* : dans le cube décrit ci haut, la sélection du membre Québec au niveau Province implique la sélection de tous les membres au niveau Région faisant partie du Québec ; idem pour les membres des autres sous-niveaux Ville et Boutique. Le résultat est la sélection des membres au niveau Boutique qui sont situés au Québec. Par conséquent, le mini-cube résultant ne contiendra aucun fait ayant trait aux ventes dans les boutiques à l'extérieur du Québec.

Agrégation sur un ou plusieurs membres pour une dimension : le client peut choisir un ou plusieurs membres pour lesquels les mesures des cellules correspondantes doivent être agrégées, selon une formule d'agrégation par défaut ou spécifiée, définie dans le cube. Ainsi, tous les membres descendant du membre

choisi sont exclus du cube résultant, puisque le grain des faits a été ramené à un niveau supérieur pour la dimension affectée. Le choix des membres sur lesquels agréger peut aussi s'effectuer sur un niveau ; en choisissant d'agréger sur tous les membres d'un niveau, on exclut du mini-cube les membres de tous les niveaux inférieurs. *Exemple : toujours avec le même cube Boutisport, on demande d'agréger sur le niveau Mois de la dimension Temps. Le niveau Jour ainsi que tous ses membres seront alors exclus du mini-cube résultant. L'opérateur agrégatif par défaut pour les mesures Unités vendues et Montant des ventes est la somme. La nouvelle granularité des faits sera les ventes mensuelles (somme des ventes quotidiennes, pour chaque mois) pour chaque produit dans chaque boutique.*

Réduction de dimension : une ou plusieurs dimensions choisies peuvent être exclues des thèmes d'analyse du mini-cube résultant, réduisant ainsi sa dimensionalité. En terminologie OLAP, il s'agit de l'opération *slice*. Pour exclure une dimension, on doit choisir un membre de cette dimension sur lequel projeter ou agréger les mesures ; par défaut, les mesures sont agrégées sur le membre « Tous ». *Exemple : on choisit d'exclure la dimension Produits lors de la construction d'un mini-cube. Par défaut, le membre « Tous » de cette dimension est utilisé pour agréger les mesures. Ainsi, la dimension Produit est exclue du mini-cube résultant, et les faits auront comme granularité la somme des ventes quotidiennes (pour tous les produits) dans chaque boutique.*

Sélection de mesures : le client peut choisir quelles mesures doivent être incluses dans le mini-cube. S'il s'agit de mesures calculées, les autres mesures dont elles dépendent sont automatiquement incluses. En pratique, cette opération peut être considérée comme un cas particulier d'inclusion de membres pour une dimension : en effet, plusieurs serveurs OLAP et langages d'interrogation (tels que MDX) représentent les mesures comme membres d'une dimension spéciale, dite dimension des mesures (*measures dimension*). En considérant une telle implantation, la sélection de mesure se fait en sélectionnant des membres de cette dimension de mesures. *Exemple : on demande d'inclure uniquement la mesure Montant des ventes. La mesure Unités vendues ne sera pas disponible dans le mini-cube résultant.*

Pour les dimensions spatiales (géométriques et mixtes), la sélection des membres à inclure ou à agréger peut non seulement se faire de manière descriptive (sélection des membres par leur nom ou identifiant), mais également avec des opérateurs topologiques d'analyse spatiale, tels que ceux définis dans la norme *ISO TC/211 19107 :2003 — Schéma spatial* (ISO/TC211, 2001). Par exemple, un utilisateur mobile pourrait choisir d'inclure les membres de la dimension *Commerce* qui sont complètement inclus dans un rayon de cinquante kilomètres autour de sa position actuelle.

Des traitements de simplification géométrique peuvent aussi être appliqués aux objets géométriques des membres spatiaux. Ces traitements visent à réduire la complexité des objets vectoriels, et par conséquent réduire la taille en octets qu'ils occupent. Par exemple, un algorithme de filtrage géométrique tel que Douglas-Peucker

(Douglas *et al.*, 1973) peut être utilisé sur les géométries des membres qu'on souhaite simplifier. Par contre, la simplification géométrique comporte un ensemble de problèmes reliés à la généralisation cartographique. Des techniques plus évoluées, telles que la représentation multiple (Bédard *et al.*, 2002), les patrons géométriques et les objets auto-généralisants (Sabo *et al.*, 2007) ou des agents (Ruas, 1999) pourraient améliorer la qualité de cette simplification. La mise en oeuvre de telles techniques dépasse toutefois le cadre du présent projet de recherche.

3.3. Encodage XML des mini-cubes

Afin que le service Web puisse transmettre les données du mini-cube au client, un encodage XML approprié est nécessaire. Tel que mentionné dans la section 2.3.4, les formats existants d'encodage XML de données OLAP, soit XMLA et XCube, ne répondent pas entièrement aux besoins du service Web de constitution de mini-cubes SOLAP. Dans le cadre de ce projet, un nouvel encodage a été conçu, afin de répondre aux exigences particulières des cubes de données spatiales (Dubé *et al.*, 2008). Cet encodage, s'inspirant de certains concepts du format XCube (Hümmer *et al.*, 2003), comporte trois modèles de documents : *CubeSchema*, *CubeMembers* et *CubeCells*.

Le document *CubeSchema* renferme les métadonnées ayant trait à la structure du cube, soit son schéma. Les éléments de ce document décrivent les dimensions, hiérarchies et niveaux faisant partie du cube. Les attributs (ou propriétés) des membres, pour chacun des niveaux, peuvent également y être définis. On peut aussi spécifier un type de données pour le contenu des attributs, conformément à la spécification *XML Schema* du W3C (Fallside *et al.*, 2004). Ainsi, en plus de tout type de données de base (nombres entiers et réels, chaînes de caractères, etc.), il est possible d'inclure des données complexes. Ces données peuvent être des entités géométriques encodées en GML, permettant ainsi la représentation complète (spatiale et descriptive) des membres de dimensions spatiales.

CubeMembers, le deuxième modèle de document, contient les membres des dimensions du cube. Les relations entre les membres y sont également explicitées, pour constituer les hiérarchies (e.g. enfant à parent) entre les membres de différents niveaux. Le document *CubeMembers* contient également les valeurs des attributs de chaque membre, conformément aux définitions d'attributs provenant du document *CubeSchema*. Dans le cas des membres spatiaux, des éléments de géométries GML peuvent figurer comme valeur d'attribut. On pourrait également substituer GML par un autre encodage d'entités géométriques, par exemple KML (utilisé dans Google Maps et Google Earth) ou encore SVG (Scalable Vector Graphics, du W3C). *CubeMembers* comporte aussi une définition des mesures du cube, représentées en tant que membres d'une dimension spéciale (dite dimension des mesures).

Finalement, le document *CubeCells* regroupe les cellules du cube, soit les valeurs correspondant aux mesures. Des références aux membres (présents dans un document *CubeMembers*) identifient chaque cellule, et la valeur pour la mesure est contenue

dans un élément. Encore une fois, cette valeur de mesure peut être d'un type de données simple (e.g. mesure numérique) ou encore complexe (e.g. mesure spatiale, encodée en GML).

La combinaison des trois documents (*CubeSchema*, *CubeMembers* et *CubeCells*) permet de reconstituer un cube complet. Cet encodage n'est pas uniquement réservé à la représentation des mini-cubes ; il convient à la transmission de petits volumes de données à destination des clients mobiles, mais pourrait tout aussi bien être utilisé pour échanger de plus grandes quantités d'information entre entrepôts de données. Des techniques de compression (e.g. *gzip*) et d'encodage XML binaire (e.g. EXI, Fast Infoset) peuvent aussi s'appliquer pour réduire la taille des données devant transiter sur le réseau. Finalement, cet encodage XML pour cubes SOLAP a été défini formellement grâce à une description en langage *XML Schema*². Une description détaillée de ce format d'encodage a aussi fait l'objet d'une publication ; le lecteur est invité à se référer à Dubé *et al.* (2008) pour plus de détail.

3.4. Architecture du service

Deux approches possibles ont été envisagées pour la réalisation du service Web de constitution de mini-cubes SOLAP : l'une est basée sur un outil ETL, et permet de construire les mini-cubes à partir des sources de données transactionnelles. L'autre approche est articulée autour d'un serveur OLAP, et produit les mini-cubes à partir de cubes SOLAP existants.

3.4.1. Service basé sur un outil ETL

Une des implémentations possibles s'appuie sur un outil d'extraction, de transformation et de chargement des données, communément appelé ETL (*Extract, Transform and Load*). Une telle implémentation permet de construire le mini-cube directement à partir des sources de données transactionnelles, sans nécessiter un entrepôt de données multidimensionnel ; une certaine réduction de la complexité des infrastructures et des coûts associés peut ainsi être réalisée. On assure également que les données des mini-cubes soient à jour par rapport aux données sources, ce qui n'est pas toujours le cas avec un entrepôt de données mis à jour périodiquement.

Cette approche comporte cependant quelques inconvénients. Les outils ETL existants sont basés sur le modèle de données relationnel ; ils n'offrent pas les capacités multidimensionnelles des outils OLAP, ce qui s'avère nécessaire lorsqu'on veut paramétrer la construction du mini-cube en fonction de critères de sélection multidimensionnels. L'implantation des opérateurs de construction de mini-cube (cf. 3.2) serait donc complexe et difficile dans un environnement ETL. Un inconvénient additionnel est la charge supplémentaire placée sur les systèmes opérationnels sources ; la performance de ceux-ci peut être affectée à chaque fois qu'ils sont sollicités pour la création

2. Cette description *XML Schema* est disponible à l'adresse <http://geosoa.scg.ulaval.ca/GeoCubeML/>

d'un mini-cube, puisqu'il s'agit d'opérations coûteuses en termes de temps de traitement. Enfin, les traitements spatiaux ETL ne peuvent pas tous être réalisés de façon totalement automatique.

Dans l'implémentation suggérée ici, la chaîne de traitement ETL est pilotée par le service Web. Les procédures ETL mises en oeuvre doivent produire en sortie des données dans un format d'échange de cubes SOLAP en XML (cf. 2.3, paragraphe « Services Web pour entrepôts de données multidimensionnelles et OLAP »), qui seront ensuite acheminées au client dans les messages SOAP de réponse.

3.4.2. *Service basé sur serveur OLAP*

Une autre approche possible est de développer le service Web autour d'un serveur OLAP, adapté pour supporter les objets géométriques pour les membres des dimensions spatiales. Le client peut demander la construction d'un mini-cube, selon toutes les opérations proposées dans la partie 3.2. Le service communique avec le serveur OLAP en utilisant l'interface de programmation et le langage d'interrogation de celui-ci, afin de récupérer les données et métadonnées nécessaires pour constituer le mini-cube. Enfin, le mini-cube SOLAP est sérialisé en format d'échange XML et acheminé au client.

Bien que cette approche nécessite un entrepôt de données géospatiales décisionnelles et un serveur OLAP, elle permet une implantation simple des opérateurs de construction de mini-cube, décrits dans la partie 3.2. Le temps de réponse rapide des systèmes OLAP apporte également une réponse à la contrainte de temps réel pour la constitution des mini-cubes. De plus, la charge sur les systèmes opérationnels (OLTP) est réduite, puisqu'ils ne sont pas sollicités à chaque invocation du service ; les procédures ETL sont responsables d'alimenter l'entrepôt de données, et ne vont typiquement solliciter les systèmes opérationnels que périodiquement, à des temps où leur usage est réduit (ex. à chaque nuit). D'une autre part, cela implique que les mini-cubes sont constitués à partir des données présentes dans l'entrepôt de données, donc à jour seulement depuis la dernière exécution des procédures ETL. Par contre, ils bénéficient du travail ETL géométrique, qui a souvent demandé une assistance manuelle pour résoudre les problèmes d'hétérogénéité spatiale. Finalement, il peut être souhaitable de conserver les mini-cubes déjà produits dans un cache ; ainsi, lorsque plusieurs clients demandent un même mini-cube, le temps de réponse peut être grandement accéléré. Ce cache peut être réalisé grâce à un service auxiliaire.

3.4.3. *Services auxiliaires*

Dans une architecture élargie pour applications SOLAP mobiles, d'autres services Web peuvent compléter les fonctionnalités du service de constitution de mini-cubes. Plusieurs de ces services sont décrits dans (Badard *et al.*, 2008) ; en voici un aperçu :

- Service d'authentification et de gestion des droits d'utilisation : ce service permet d'authentifier les utilisateurs et de donner accès aux cubes correspondant à leurs droits d'utilisation, afin de sécuriser l'accès aux données confidentielles.

– Service d’entreposage de mini-cubes : sert à indexer et à stocker les mini-cubes générés par le service de constitution. De multiples clients peuvent alors télécharger des cubes existants, sans avoir à redemander leur construction (concept de cache).

– Service de chaînage et de livraison de mini-cubes : permet de combiner plusieurs services de constitution ainsi que différents entrepôts de mini-cubes, afin de donner accès de manière centralisée à un éventail de sources de données. Offre également divers mécanismes de livraison (immédiate ou différée) des cubes.

Ces services auxiliaires montrent qu’on peut combiner plusieurs services Web distribués, dans une architecture orientée services, pour répondre à un éventail de besoins et de cas d’utilisation particuliers, dans des contextes autant sédentaires que mobiles.

4. Réalisation du service

Un prototype de service Web de constitution de mini-cubes SOLAP a été développé au cours de ce projet. Ce service est basé sur un serveur OLAP, tel que décrit dans la section 3.4.2. Il est réalisé en langage Java, et exploite les fonctionnalités du serveur OLAP *open source* Mondrian³ (Pentaho Analysis Services), les objets spatiaux du cadre de développement GeOxygene (Badard *et al.*, 2003) ainsi que la librairie Apache Axis2⁴ pour le support du protocole de service Web SOAP. Le choix de technologies *open source* facilite l’intégration de nouvelles fonctionnalités (i.e. membres spatiaux) dans le serveur OLAP. Un entrepôt de données a été construit à des fins de tests, en utilisant le SGBD PostgreSQL⁵ avec les extensions spatiales PostGIS⁶. Le peuplement de cet entrepôt est réalisé grâce à l’outil ETL GeoKettle⁷, une version de Pentaho Data Integration⁸ adaptée pour supporter les types de données et traitements spatiaux. Le service Web utilise l’interface de programmation du serveur OLAP pour interroger les cubes de données et ainsi constituer les mini-cubes. Le client mobile invoque les fonctionnalités du serveur à l’aide du protocole SOAP, et les données (mini-cubes SOLAP) lui sont retournées dans les messages de réponse, en format XML, conformément à l’encodage décrit dans la section 3.3. L’architecture générale de ce système est illustrée à la figure 3.

3. <http://mondrian.pentaho.org>

4. <http://ws.apache.org/axis2/>

5. PostgreSQL est un SGBD relationnel *open source*; <http://www.postgresql.org>.

6. PostGIS ajoute à PostgreSQL les types de données et opérateurs spatiaux définis par la spécification *OGC Simple Features for SQL*; <http://postgis.refractory.net>.

7. Ce logiciel libre, développé par l’équipe de recherche GeoSOA, est disponible à l’adresse <http://www.geokettle.org>

8. Pentaho Data Integration, anciennement Kettle, est un outil ETL *open source*; <http://kettle.pentaho.org>.

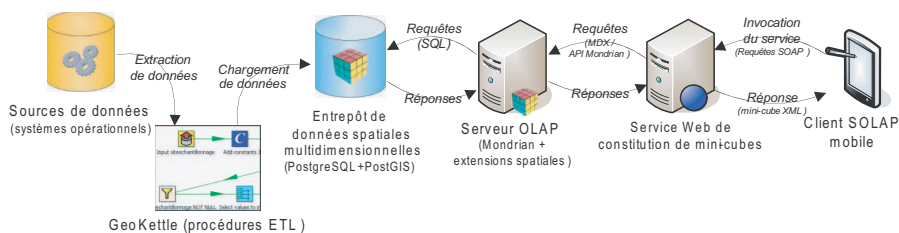


Figure 3. Architecture du prototype de service Web

4.1. Contrat de service

Le service Web propose les opérations suivantes, inspirées de celles offertes par le service d'entités cartographiques (WFS) de l'OGC. Ces opérations constituent le contrat de service, et sont généralement appelées selon la séquence suivante :

- Opération *GetCapabilities*, informant le client de la version du service et donnant des informations sur l'identification (ex. nom de l'organisation responsable du service, personnes responsables, etc.) et les capacités du service (ex. le support ou non de certaines composantes optionnelles du contrat de service).

- Opération *DescribeSchema*, permettant d'obtenir les métadonnées. On peut demander de décrire des objets tels que les cubes, les dimensions, les hiérarchies, les niveaux et les mesures. Ces objets constituent le schéma des cubes, i.e. ils décrivent la structure et non le contenu des cubes. Le message de réponse est conforme au modèle de document *CubeSchema*, décrit dans la section 3.3.

- Opération *GetMembers*, permettant d'interroger les dimensions d'un cube et d'obtenir les membres. Cette opération doit supporter la navigation dans une hiérarchie de membres, i. e. obtenir les parents ou enfants d'un membre, obtenir les membres selon un niveau, etc. Cette opération propose également une méthode de sélection spatiale des membres (selon des opérateurs métriques ou topologiques). Les données retournées sont contenues dans un document *CubeMembers*.

- Opération *GetCells*, offrant un mécanisme d'interrogation des données (faits) d'un cube. Les informations obtenues précédemment par l'intermédiaire des autres opérations (*DescribeSchema* et *GetMembers*) permettent de sélectionner les cellules correspondant au croisement des membres choisis, pour un cube donné. Elle s'appuie sur les opérateurs de construction de mini-cube (cf. 3.2). L'argument de cette opération est donc une « requête de sélection » multidimensionnelle définissant l'ensemble des cellules du cube à obtenir. La réponse à cette requête est un document *CubeCells*, renfermant toutes les cellules qui satisfont aux critères de la requête.

- Opération *GetCube*, permettant de récupérer un mini-cube complet. Cette opération s'appuie sur *GetCells* pour obtenir les données voulues, et ajoute en plus tous les membres correspondant à la sélection du sous-cube ainsi que la description du

schéma de ce sous-cube (métadonnées, i. e. structure des dimensions, hiérarchies, niveaux et mesures). Le document XML retourné est donc une représentation complète du mini-cube demandé (données et métadonnées).

Afin de formaliser et de permettre la validation de la syntaxe des opérations du contrat de service, une description en langage WSDL (Christensen *et al.*, 2001) a été développée. Celle-ci réutilise l'encodage pour cubes XML décrit précédemment, en permettant l'encapsulation d'éléments des documents *CubeSchema*, *CubeMembers* et *CubeCells* pour constituer les messages de réponse des opérations du contrat de service.

Il importe de mentionner que le contrat de service constitue une interface technique pour la construction des mini-cubes. Des outils à interface graphique conviviale (i.e. client SOLAP mobile) pourront exploiter ces fonctionnalités, afin de les rendre transparentes à l'utilisateur.

4.2. Tests de performance du prototype

Quelques tests ont été réalisés avec le prototype de service Web, afin de mesurer sa performance. L'environnement de test repose sur un PC de bureau muni d'un processeur AMD Athlon 64 X2 3800+, de 2 Go de mémoire vive, et fonctionnant sous l'environnement Windows XP. Quatre requêtes ont été testées, soit une pour chacune des opérations du contrat de service (sauf *GetCube* qui n'est pas implantée dans la version actuelle du prototype). Chaque requête est lancée une fois à froid (tout de suite après le démarrage du service) et une autre fois à chaud (après la première exécution). Les résultats sont présentés dans le tableau 1.

Requête	Temps d'exécution à froid	Temps d'exécution à chaud
<i>GetCapabilities</i>	4s 141ms	0s 265ms
<i>DescribeSchema</i>	0s 281ms	0s 31ms
<i>GetMembers</i>	30s 344ms	16s 47ms
<i>GetCells</i>	5s 313ms	0s 297ms

Tableau 1. Temps d'exécution des requêtes de test

Les requêtes *GetCapabilities* et *DescribeSchema* s'exécutent rapidement, puisque les volumes de données en jeu sont petits et qu'aucun traitement complexe n'est réalisé pour y répondre. Les requêtes *GetMembers* et *GetCells* mettent plus de temps à s'exécuter, puisque les membres et cellules demandés doivent être sélectionnés, encodés en XML et retournés au client. La requête *GetMembers* du test, qui comporte une sélection spatiale, retourne un total de 21 membres. Il serait possible d'améliorer la performance de cette requête par l'utilisation d'un index spatial ; cela n'a toutefois pas été implanté dans le prototype actuel par manque de temps. Quant à la requête *GetCells*, elle retourne un total de 297 cellules ; la majeure partie du temps de traitement va dans l'exécution de la requête MDX pour récupérer ces cellules. D'ailleurs,

on peut voir, autant dans le cas de *GetMembers* que de *GetCells*, que les requêtes lancées à chaud prennent moins de temps à être traitées que celles lancées à froid. Cela s'explique par le fait que le serveur OLAP (Mondrian) comporte un *cache* de données, donnant ainsi un temps de réponse plus court lorsque les données demandées sont déjà chargées en mémoire vive.

5. Conclusion

Les architectures orientées services, ainsi que leur réalisation sous forme de services Web, constituent une solution valable pour la diffusion de données géodécisionnelles, dans un contexte de mobilité. Nous avons établi les principes conceptuels et défini l'architecture d'un service Web permettant la constitution de cubes de données SOLAP de taille réduite, adaptés à ces environnements mobiles. La réalisation d'un tel service, s'appuyant sur des normes ouvertes et des technologies *open source*, a aussi été décrite.

La démonstration de cette architecture a également été faite par la réalisation d'un prototype fonctionnel de service Web de constitution de mini-cubes SOLAP, basé sur un ensemble de technologies *open source*. Des schémas XML pour décrire le format d'échange des mini-cubes SOLAP, élaborés selon la spécification W3C XML Schema (Fallside *et al.*, 2004), ainsi que des documents WSDL décrivant les opérations du contrat de service furent aussi développés dans le cadre de cette réalisation. Les tests du prototype, réalisés avec succès, montrent une performance adéquate et viennent valider l'approche choisie.

De nouveaux projets de recherche en lien avec celui-ci sont actuellement en cours au sein du groupe de recherche GeoSOA⁹. Ces travaux portent sur la réalisation d'applications mobiles en géomatique décisionnelle, s'appuyant sur les services Web ici décrits. Ils viendront apporter des perspectives novatrices sur les méthodes de visualisation et sur les interfaces utilisateurs en contexte mobile. À terme, les architectures orientées services pour SOLAP serviront à réaliser un vaste éventail d'applications, qu'elles soient mobiles ou sédentaires. Là où des services tels que WFS comblent les besoins d'accès aux données spatiales transactionnelles, organisées selon un modèle relationnel, des services tels que celui ici décrit pourront aider à combler les besoins d'accès aux données spatiales décisionnelles, organisées selon un modèle multidimensionnel.

Remerciements

Ce projet de recherche a été rendu possible grâce au soutien de la Chaire de recherche industrielle en bases de données géospatiales décisionnelles, dirigée par le

9. <http://geosoa.scg.ulaval.ca>

professeur Yvan Bédard, ainsi que celui du groupe de recherche GeoSOA, dirigé par le professeur Thierry Badard.

6. Bibliographie

- Badard T., Bédard Y., Hubert F., Bernier E., Dubé E., « Web services-oriented architectures for mobile SOLAP applications », *International Journal of Web Engineering and Technology (IJWET)*, 2008.
- Badard T., Braun A., « OXYGENE — D'une plate-forme interopérable au déploiement de services web géographiques », *Revue internationale de géomatique*, vol. 3, n° 13, p. 411-430, 2003.
- Bédard Y., Bernier E., « Supporting Multiple Representations with Spatial View Management and the Concept of "VUEL" », *Joint Workshop on Multi-Scale Representation of Spatial Data, International Society for Photogrammetry and Remote Sensing (ISPRS) WG IV/3. ICA. Com. On Map Generalization*, Ottawa, Canada, juillet, 2002.
- Bédard Y., Proulx M.-J., Rivest S., « Enrichissement du OLAP pour l'analyse géographique : exemples de réalisation et différentes possibilités technologiques », *Revue des Nouvelles Technologies de l'Information — Entrepôts de données et l'Analyse en ligne*, p. 1-20, 2005.
- Bédard Y., Proulx M.-J., Rivest S., « « Notions de modélisation multidimensionnelle (conceptuelle) », note du cours « Notions avancées de bases de données SIG (SCG-66124) » », 2006. <http://sirs.scg.ulaval.ca/YvanBedard/enseigne/note66124.htm>.
- Cerami E., *Web Services Essentials*, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2002.
- Christensen E., Curbera F., Meredith G., Weerawarana S., *Web Services Description Language (WSDL) 1.1, spécification technique*, World Wide Web Consortium, March, 2001. <http://www.w3.org/TR/wsdl>.
- Codd E. F., Codd S. B., Salley C. T., *Providing OLAP to users-analysts : an IT mandate*, white paper, Hyperion Solution Corporation, 1993.
- Douglas D. H., Peucker T. K., « Algorithms for the reduction of the number of points required to represent a line or its caricature », *Cartographica : The International Journal for Geographic Information and Geovisualization*, vol. 10, n° 2, p. 112-122, 1973.
- Dubé E., Badard T., Bédard Y., « An interoperable XML encoding for the exchange of Spatial OLAP data cubes in SOA environments », *31st International Convention MIPRO on Business Intelligence Systems (miproBIS)*, Opatija, Croatie, 2008.
- Duda I., Aleksey M., Butter T., « Architectures for Mobile Device Integration into Service-Oriented Architectures », *Fourth International Conference on Mobile Business (ICBM2005)*, Sydney, Australia, p. 193-198, 2005.
- Fallside D. C., Walmsley P., *XML Schema Part 0 : Primer (Second Edition)*, spécification technique, World Wide Web Consortium, October, 2004. <http://www.w3.org/TR/xmlschema-0/>.
- Hümmer W., Bauer A., Harde G., « XCube : XML for data warehouses », *DOLAP '03 : Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, ACM, New York, NY, USA, p. 33-40, 2003.
- ISO/TC211, *Geographic Information - Spatial Schema (ISO 19107) (OGC Topic 1 working draft Version 5)*, spécification technique, ISO Technical Committee 211, 2001.

- ISO/TC211, Geographic information — Geography Markup Language (GML) (Committee Draft, version 3.1.0), spécification technique, ISO Technical Committee 211 & Open Geospatial Consortium, 2004.
- KHEOPS Technologies, JMap Spatial OLAP, white paper, 2005.
http://www.kheops-tech.com/en/jmap/doc/WP_JMap_SOLAP.pdf.
- Kimball R., Caserta J., *The Data Warehouse ETL Toolkit*, John Wiley & Sons, 2004.
- Kimball R., Ross M., *The Data Warehouse Toolkit : The Complete Guide to Dimensional Modeling (2nd edition)*, John Wiley & Sons, 2002.
- Microsoft Corporation, « Multidimensional Expressions (MDX) Reference », 2007a.
<http://msdn2.microsoft.com/en-us/library/ms145506.aspx>.
- Microsoft Corporation, « OLE DB for Online Analytical Processing (OLAP), Programmer's Reference », 2007b.
<http://msdn2.microsoft.com/en-us/library/ms717005.aspx>.
- Microsoft, Hyperion, XML for Analysis Specification, version 1.1, spécification technique, Microsoft Corporation, 2002.
- Newcomer E., Lomow G., *Understanding SOA with Web Services*, Addison-Wesley Professional, 2004.
- OGC, Web Feature Service Implementation Specification, version 1.1.0, spécification technique, Open Geospatial Consortium, 2005.
- Pentaho Corporation, « Pentaho Analysis Services : Mondrian Project », 2007.
<http://mondrian.pentaho.org>.
- Proulx M.-J., Bédard Y., Nadeau M., Gosselin P., Lebel G., « Géomatique et santé environnementale : innovations résultant du projet ICEM/SE », *Géomatique 2002*, Montréal, Québec, Canada, octobre, 2002.
- Rivest S., Bédard Y., Marchand P., « Toward better support for spatial decision making : defining the characteristics of Spatial On-Line Analytical Processing (SOLAP) », *Geomatica*, vol. 55, n° 4, p. 539-555, 2001.
- Ruas A., Modèle de généralisation de données géographiques à base de contraintes et d'autonomie, PhD thesis, Université de Marne La Vallée, France, 1999.
- Sabo M. N., Bédard Y., Bernier E., Cardenas A., « Methodology for developing a database of geometric patterns to better support on-the-fly map generalization », *International Cartographic Conference*, Coruna, Espagne, 2007.
- Thomsen E., *OLAP Solutions : Building Multidimensional Information Systems*, John Wiley & Sons, Inc., New York, NY, USA, 2002.