

Institut Géographique National
Direction Technique
Service de la Recherche
Projet Diffusion Num. Réseau



DIFNUM/301
Lot Numéro 3

Gilles Cébéliu
DT/SR/TRD Poste 8426

Version 1.1
19 mars 2004

Bases de données, XML et métadonnées

```
<SPEECH>  
<SPEAKER>HAMLET</SPEAKER>  
<LINE>Real, usual, perturbed spirit!</LINE>  
<STAGEDIR>They enter</STAGEDIR>  
<LINE>So, gentle...</LINE>  
<LINE>...tell my...</LINE>  
<LINE>...man...</LINE>  
<LINE>...his...</LINE>  
<LINE>...together</LINE>  
<LINE>...the...</LINE>  
<LINE>...I was born to see it right</LINE>  
<LINE>Say, come, let's go together.</LINE>  
</SPEECH>
```



PROJET DIFFUSION DES DONNÉES NUMÉRIQUES SUR LE RÉSEAU
SERVICE DE LA RECHERCHE - INSTITUT GÉOGRAPHIQUE NATIONAL
2 AVENUE PASTEUR - 94165 SAINT-MANDÉ CEDEX
TÉLÉPHONE : 01 43 98 83 93 - TÉLÉCOPIE : 01 43 98 81 71

Métadonnées du document

Identifiant	DIFNUM/301
Titre	Bases de données, XML et métadonnées
Auteur	Gilles Cébélieu Projet Diffusion des Données Numériques sur le Réseau
Version	1.1
Date	19 mars 2004
Résumé	Ce document étudie, dans le cadre de la gestion des métadonnées, les solutions de stockage de données XML au travers de bases de données XML natives ou d'une conservation à l'aide de SGBD Relationnels classiques. Cette étude s'appuie sur des tests effectués sur deux bases XML natives Open source : eXist et Xindice.
Mots Clefs	XML, Métadonnées, SGBD

Vie du document

<i>version</i>	<i>date</i>	<i>événements</i>
1.1	19 mars 2004	Ajout d'un glossaire, des résultats des tests sur Xindice et des fichiers d'exemples de métadonnées et de demandes de prestations. Corrections diverses suite aux remarques de Didier Richard
1.0	5 mars 2004	Création du document

Table des matières

Métadonnées du document.....	3
Le Besoin.....	6
Les bases de données XML natives.....	7
Définition.....	7
L'initiative XML:DB.....	7
Aperçu des bases de données XML existantes.....	8
Solutions de stockage des fichiers XML.....	8
Stockage et mise en ligne.....	8
Schéma de données.....	8
Normalisation.....	9
Sécurité et administration.....	10
Capacité d'évolution (« Scalabilité »).....	10
Consultation.....	11
Langages de requêtes.....	11
Indexation.....	12
Mises à jour	12
Présentation.....	13
Echanges.....	13
« Round-tripping ».....	13
Conclusion.....	14
Inconvénients des bases XML natives.....	14
Avantages des bases XML natives.....	14
Mise en oeuvre de l'indexation géographique.....	14
Indexation géographique.....	14
Requêtes géographiques.....	15
Glossaire.....	16
Annexe A : Etude des bases XML natives : eXist et Xindice	18
logiciels utilisés	18
Tests sur des demandes de prestation	18
Chargement des fichiers dans la base.....	19
Récupération d'un document par sa clef (son nom) et sérialisation.....	19
Requête XPATH (1).....	19
Requête XPATH (2).....	19
Requête XPATH croisée.....	20
Requêtes XQuery	20
Mise à jour au travers d'XUpdate.....	21
Fichiers issus du logiciel TA	21
Chargement dans la base	21
Récupération d'un document par sa clef (son nom) et sérialisation.....	22
Requêtes XPATH	22
Requêtes XQuery	23
Métadonnées des scans de la série régionale	24
Annexe B : exemples de fichiers XML à gérer	25
Métadonnées (implémentation proposée par le Projet)	25

Le Besoin

Le système que le Projet « Diffusion des données » souhaite mettre en place s'appuie sur le stockage et la recherche de métadonnées associées aux données archivées ou à diffuser.

Ces dernières seront fournies sous forme de fichiers XML dont le contenu sera conforme à la norme ISO 19115 sur les métadonnées¹. Elles seront organisées hiérarchiquement pour un même produit de la manière suivante : un lot de métadonnées décrivant le produit (spécifications et caractéristiques communes à toutes les données d'un même produit) et des lots de métadonnées associés chacun à un lot de données produit et faisant référence aux métadonnées du produit².

Elles représentent pour l'utilisateur le point d'entrée du système d'archivage et de diffusion³, en ce sens que tout accès aux données (archivage, recherche, extraction) passera par les métadonnées. De ce fait, le système de gestion des métadonnées devra assurer les fonctions suivantes :

hébergement : stockage et mise en ligne ;

consultation : il faudra pouvoir interroger les métadonnées selon des critères sémantiques (mots-clefs, dates...) ou géographiques (pointés, emprises ou données de contexte); par ailleurs les informations contenues dans les métadonnées devront pouvoir être présentées à l'utilisateur sous une forme conviviale au travers d'un client web ;

échanges : les métadonnées seront intégrées à partir des fichiers XML décrits ci-dessus, elles devront pouvoir être récupérées sous la même forme ;

maintenance : saisie et mise à jour des métadonnées en ligne.

Par ailleurs, les demandes de prestation d'archivage et de diffusion sont transmises au prestataire (le Serveur Général) au travers de fichiers XML dont le contenu est lui aussi conforme à la norme ISO 19115 ainsi qu'à la norme ISO 19119 sur les services. Pour l'instant, ces fichiers, une fois utilisés pour effectuer la prestation, sont simplement stockés dans un répertoire. Ces fichiers pourront aussi être conservés au même titre que les métadonnées dans un système permettant leur stockage et leur recherche (date, produit concerné...).

Cette étude a pour but d'évaluer les solutions de stockage de ces données XML au travers de bases de données XML natives ou d'une conservation à l'aide de SGBD Relationnels classiques.

1 la structure des fichiers XML sera conforme soit à la première implémentation de la norme ISO 19115 que le Projet a faite conformément à l'exemple fourni dans cette étude (cf. annexe B), soit, à terme, à la norme ISO 19139 d'implémentation des métadonnées qui est en cours de test avec la cellule normalisation.

2 un niveau hiérarchique intermédiaire pourra être éventuellement rajouté selon la complexité du produit (e.g. produits composites).

3 cf. DIFNUM/201 : « Mise en place de la base de métadonnées pour l'archivage, la consultation et la diffusion des données numériques sur le réseau (Définition Détaillée) ». Version numérique disponible en ligne à l'adresse suivante : http://difnumadm.ign.fr/docs/DD_MDv1.0.pdf

Les bases de données XML natives

Définition

Les bases de données XML natives (en anglais NXD pour « Native XML Database ») sont des bases de données réalisées pour le stockage de données XML. On peut les caractériser par le fait qu'elles offrent à l'utilisateur une vision logique des données en accord avec le modèle XML (organisation hiérarchique des informations...), de la même façon que les bases de données relationnelles présentent une vision logique des données conforme au modèle relationnel (organisation des informations sous forme de tables). Cette vision logique conforme au modèle XML permet d'envisager aisément d'utiliser les standards définis autour de XML (XQuery, XPath, XSLT, XUpdate) pour accéder et traiter les données de la base.

L'initiative XML:DB

Il s'agit d'une initiative visant à fédérer les efforts déployés par des industries ou des organisations autour de la nouvelle technologie représentée par les bases de données XML. Plus concrètement, les buts de cette initiative sont⁴ :

- développer des spécifications techniques pour la manipulation des données au sein des bases de données XML ;
- contribuer au développement d'implémentations de référence de ces spécifications sous une licence Open Source ;
- former une communauté où utilisateurs et distributeurs de bases de données XML peuvent échanger ;
- faire la publicité des produits et technologies bases de données XML sur le marché.

Dans ce but, cette initiative est à l'origine de plusieurs projets dont notamment:

- la mise en place d'une interface de programmation (**XML:DB API**) standard d'accès aux bases de données XML indépendante de tout constructeur (cf <http://www.xmldb.org/xapi/index.html>) ;
- la définition d'un langage de mise à jour de données XML (**XUpdate**), lui même véhiculé sous forme de données XML (cf. <http://www.xmldb.org/xupdate/index.html>) ;
- la définition d'un langage de définition et de modification de données (**SixDML**) pour les base de données XML (cf. <http://www.xmldb.org/sixdml/index.html>) à l'instar de SQL pour les bases de données relationnelles.

L'initiative XML:DB montre un certain ralentissement d'activité (les dernières mises à jour du site de l'initiative XML:DB datent de mai 2003 et les résultats des différents projets semblent arrêtés au stade du « Working Draft », dont le plus récent date de 2002 !), cependant, les projets XUpdate et XML:DB API semblent avoir rempli une part de l'objectif fédérateur en ce sens que de nombreuses bases XML implémentent ou essaient d'implémenter ces spécifications, même à un stade non abouti.

⁴ cette présentation est tirée du site de l'initiative XML:DB : <http://www.xmldb.org/index.html>

Aperçu des bases de données XML existantes

De nombreux produits commerciaux ou sous licence Open Source existent, dans un état de réalisation plus ou moins avancé (voire abandonné!). Un aperçu relativement exhaustif (autant que cela puisse être possible) est donné sur la page internet suivante : <http://www.rpbouret.com/xml/XMLDatabaseProds.htm>

Au niveau de cette étude, on signalera juste que les solutions proposées reposent sur des technologies variées : bases de données relationnelles ou orientées objet offrant des extensions permettant d'intégrer des données XML (« XML-enabled databases »), bases XML natives dont le stockage repose sur un modèle relationnel ou sur une implémentation propre.

Nous nous sommes concentrés ici, sur deux projets open source encore très actifs et ayant atteint un degré de réalisation permettant d'envisager une utilisation industrielle : Xindice et eXist (cf. Annexe A pour les tests effectués).

Solutions de stockage des fichiers XML

Les enjeux du stockage de données XML dans des SGBD-R et dans des bases de données XML natives sont abordés d'une manière théorique et approfondie dans un article de Ronald Bourret, consultable aux adresses suivantes :

texte original : <http://www.rpbouret.com/xml/XMLAndDatabases.htm>

traduction française : <http://peccatte.karefil.com/software/RBouret/xmlBD.htm>

Le Projet Diffusion des données en a réalisé une version PDF, disponible à l'adresse suivante :

<http://difnumadm.ign.fr/docs/xml-et-BDs.pdf>

Dans le contexte des besoins du Projet, on étudiera la pertinence des différentes solutions de stockage aux regard des fonctionnalités énoncées précédemment : stockage, consultation et échanges.

Stockage et mise en ligne

Schéma de données

Le stockage de données dans un SGBD relationnel nécessite la définition au préalable de la structure des données (tables et relations) qui vont y être insérées. Une fois définie, toute donnée non conforme à cette structure ne pourra y être insérée à moins de modifier cette dernière.

Les bases de données XML natives proposent une vision logique des données conforme au modèle XML. La seule contrainte concernant la structure des documents XML stockés dans la base est que ceux-ci soient « bien formés », à savoir qu'ils respectent bien la syntaxe XML. Certaines bases offrent, au travers du parser XML utilisé, la possibilité ou non de valider la syntaxe d'un fichier par rapport à un schéma donné (XSD, DTD...), mais la définition de ce schéma est indépendante de la base : on peut insérer n'importe quel

type de document conforme à n'importe quel schéma sans avoir à modifier la base. L'organisation des données au sein d'une base XML native se fait au niveau des documents, que l'on peut regrouper au sein de collections, exactement de la même manière que l'on regroupe des fichiers dans des répertoires.

Cette souplesse de stockage est intéressante pour la manipulation de documents dont la structure n'est pas définie ou fluctuante, ce qui ne sera pas le cas des métadonnées que l'on souhaite stocker qui seront structurées selon un schéma fixe⁵.

La différence entre les deux solutions pour ce type de données structurées réside sur les transformations qui seront nécessaires pour l'intégration des données dans le SGBD Relationnel : un script de création de la structure des métadonnées sous forme de tables⁶ et une application de transformation des fichiers XML en script d'importation des informations dans la structure, contre une intégration directe du document dans une base de données XML.

Normalisation

La normalisation est une opération qui vise à définir un schéma de données qui minimise la taille des données stockées en supprimant la redondance. Il va de soi qu'une telle opération dépend surtout de la conception du schéma des données à héberger⁷. La solution de stockage va permettre avec plus ou moins de souplesse de mettre en oeuvre un schéma optimisé :

- il s'agit du concept même des bases de données relationnelles, qui permettent d'organiser les données en tables et relations ;
- XML permet de définir des relations entre documents ou des références à des noeuds au travers des standards XLink (liens inter documents), XInclude (inclusion de fragments de documents) et XPointer (références à des fragments documents). Selon les bases, ces standards sont plus ou moins supportés. La base eXist offre une implémentation des liens au travers de XInclude et XPointer (cf. Annexe A - tests sur les métadonnées)

La normalisation induit des règles de cohérence entre les données liées (intégrité référentielle). Les bases de données relationnelles permettent généralement d'assurer cette cohérence, ce qui n'est pas forcément le cas pour les bases XML natives (dans eXist, la disparition de noeuds XML référencés à l'aide de XInclude par d'autres documents de la base se fait sans vérification particulière).

En ce qui concerne les métadonnées que le Projet souhaite stocker, on aura une normalisation à deux niveaux :

1. une normalisation « implicite » du fait de l'organisation hiérarchique des métadonnées : les métadonnées de produit vont contenir des éléments communs à toutes les métadonnées sur les données appartenant à ce produit. Ici, les références se feront au travers des champs des métadonnées liant parent et enfants (éventuellement à l'aide

5 DTD utilisée actuellement par le Projet ou XSD (à venir) issus de la norme d'implémentation ISO 19139.

6 Un tel script a déjà été réalisé pour le Projet par la société Global Géomatics. Il est disponible à l'adresse suivante : <http://difnumadm.ign.fr/docs/MapFusion-script-creation-v5.zip>

7 Comme le fait remarquer Ronald Bouret : « [on peut] concevoir un mauvais stockage des données avec une base XML native aussi facilement qu'avec une base relationnelle. »

d'XPointer), mais ces liens devront être restitués tels quels (en aucun cas ils ne seront instanciés). Ici, l'optimisation viendra d'une intelligente répartition de ces informations entre parents et enfants qui devra se faire au niveau de la création des métadonnées plutôt qu'au niveau du stockage ;

2. certaines informations communes à toutes les données d'un produit devront quand même figurer dans les métadonnées sur les données du fait du caractère obligatoire des champs de la norme auxquels elles correspondent⁸. Dans ce cas, on pourra envisager d'effectuer des regroupements d'informations au niveau du stockage, reposant sur des liens.

Il convient de noter que le volume des informations pouvant être redondantes dans le second cas reste assez négligeable comparé au volume total des métadonnées : la mise en oeuvre d'une gestion de liens ne représentera pas un gain extraordinaire vis à vis de la taille de la base.

Sécurité et administration

Il s'agit ici des fonctions de base permettant d'assurer la sécurité et l'intégrité des données contenues dans la base dans le cadre d'une utilisation comme serveur de données :

- gestion des droits d'accès ;
- gestion d'accès concurrents ;
- gestion de transactions ;
- gestions de sauvegardes.

Si dans leur ensemble, les SGBD transactionnels mettent en oeuvre ces fonctionnalités, les bases de données XML ne les offrent pas toutes :

- eXist offre une gestion des droits d'accès aux documents identique à celle d'un système de fichiers UNIX, gère les accès concurrents par un système de verrouillage des ressources et propose un utilitaire de Backup/Restore de la base mais n'est pas transactionnel ;
- Xindice ne propose aucun système d'authentification pour accéder à la base (accès libre) en déléguant cette responsabilité au serveur http qui diffuse la base ; il n'est pas non plus transactionnel et ne semble pas gérer les accès concurrents. Le système de sauvegarde de la base est sommaire : arrêt du serveur et copie des fichiers de données !

Capacité d'évolution (« Scalabilité »)

Il s'agit ici de la capacité du système de gestion à maintenir de bonnes performances sur de gros volumes de données.

L'étude technique sur la mise en place de la base de métadonnées⁹ proposait un scénario dans lequel la base de métadonnées décrivant l'intégralité des données relatives aux produits tels que la BDDOrtho, la BDPays, les PVA et la BDParcelaire aurait une taille de 1Go. Les tests effectués sur la base eXist montrent que celle-ci réussit à « encaisser » un

⁸ Il s'agit essentiellement des informations de contact : références à des personnes impliquées dans la production ou la distribution des données.

⁹ DIFNUM/202, « Etude technique - Mise en place de la base de métadonnées pour l'archivage, la consultation et la diffusion des données numériques sur le réseau » (22/10/2001). version numérique disponible à l'adresse suivante : http://difnumadm.ign.fr/docs/ET_MDv1.0.pdf

chargement de 45000 documents représentant un volume de données de l'ordre de 1,9Go tout en offrant des temps de réponse corrects à des requêtes simples (1 à 2 sec. sans index et 50ms avec index)¹⁰. Des documents de gros volume (de l'ordre de 5Mo) ont pu être chargés au travers d'une interface réseau comme au travers d'un accès direct à la base. Les tests sur la base Xindice n'ont pas pu aboutir pour l'instant sur ce type de données).

Pour information eXist présente une limitation théorique du nombre de documents stockés dans la base de données à 2³¹ (plus de 2 milliards de documents).

Consultation

Langages de requêtes

Les SGBD relationnels bénéficient d'un langage standard de manipulation de base de données : le SQL. Celui-ci est implémenté par tous les producteurs de SGBD-R dignes de ce nom (et surtout qui souhaitent pouvoir être exploités sur le marché). Le langage SQL offre tout un panel de commandes permettant d'effectuer les opérations nécessaires à la manipulation d'une base de données :

- manipulation de données : insertion, suppression, interrogation (*DML : Data Manipulation Language*)
- définition de données : création, modification, suppression du schéma (DDL : *Data Definition Language*)
- administration : attribution, révocation de droits d'accès (DCL : *Data Control Language*)

La syntaxe SQL est adaptée à des données conformes au modèle relationnel ; en aucun cas au modèle XML. Ce dernier s'accompagne de standards spécifiques permettant la manipulation des informations qu'il véhicule : XPath, XQuery, SiXDML ou XUpdate.

Aucun de ces langages ne couvre l'ensemble des fonctionnalités du SQL. On peut les répartir ainsi selon la classification DML, DDL¹¹ et DCL :

	DML <i>(manipulation de données)</i>	DDL <i>(définition de données)</i>	DCL <i>(administration de données)</i>
XPath	interrogation	non	non
XQuery	interrogation	non	non
SiXDML	oui	oui	non
XUpdate	suppression, mise à jour	non	non

répartition des différents langages de requêtes selon la classification DML, DDL, DCL

Mis à part SiDXML, tous les autres langages ne couvrent qu'un domaine partiel des fonctionnalités d'un langage de manipulation de bases de données. L'accès aux bases

¹⁰ cf. Annexe A : étude des bases XML natives eXist et Xindice

¹¹ Comme il l'a été dit précédemment, la définition de schéma de données est indépendante des bases de données XML ; la notion de DDL pour ces dernières se limite donc à la manipulation de collections (création, destruction, interrogation...)

XML natives se fait donc au travers de plusieurs langages, qui ne seront pas toujours standards :

- l'administration des bases de données XML natives reste complètement propriétaire du produit utilisé¹² ;
- à part XPath qui semble être implémenté par la plupart des bases de données XML, aucun langage ne semble s'être imposé comme LE langage de référence de manipulation de ces bases. A titre d'exemple :
 - eXist utilise XPath, XQuery et XUpdate
 - Xindice utilise XPath, SiXDML et XUpdate

Si SiDXML semble être le plus à même de remplir le rôle d'un SQL pour les bases de données XML, celui-ci ne semble pas avoir été massivement adopté et semble même abandonné. XQuery, du fait de son origine W3C et de sa souplesse d'utilisation, permettant d'effectuer des requêtes d'une complexité comparable à celles effectuées à l'aide de SQL, devrait sans doute devenir un langage de requête de référence pour les bases de données XML.

Indexation

La définition d'index sur les données XML est nécessaire pour satisfaire les besoins énoncés pour l'interrogation des métadonnées dans des délais raisonnables face à une montée en charge de la base :

- les requêtes sémantiques sur des mots-clefs nécessitent la définition d'index textuels sur certains champs des métadonnées. Les bases de données XML comme les SGBD-Relationnels permettent de définir des index. Si cela peut se faire de manière standard pour les SGBDs avec des commandes SQL, la définition d'index pour des bases de données XML reste propriétaire de la solution utilisée¹³ (on les définit à l'aide du fichier de configuration de la base pour eXist alors qu'on les définit en ligne de commande pour Xindice) ;
- les requêtes géographiques (sélection par emprise...) nécessitent la définition d'index géographiques particuliers (de type rtree ou autre). Aucune base de données XML ne semble proposer de telle possibilité alors que les SGBDs relationnels proposent de plus en plus de telles extensions (Oracle, Informix, PostgreSQL, MySQL,...).

Mises à jour

Il s'agit ici de la mise à jour des données d'une base.

Les bases de données XML permettent de faire cela à deux niveaux :

- au niveau du document, en supprimant l'ancien et insérant le nouveau ;
- au niveau de la granularité du modèle XML (mise à jour de noeuds, d'attributs...) à l'aide du langage XUpdate ou SiXDML pour les bases qui les implémentent.

¹² L'API XML:DB propose un système de contrôle d'accès au niveau de la collection, mais celui-ci n'est pas toujours mis en oeuvre (cf. Xindice)

¹³ Seul le langage SiXDML propose des commandes permettant d'indexer des informations, mais comme il a été dit, ce dernier n'a pas abouti à une implémentation massive par les bases de données XML..

Présentation

Il s'agit de la mise en forme des données stockées dans la base.

Ici, les bases de données XML présentent tout leur intérêt : les résultats de requêtes sont directement des noeuds XML que l'on peut aisément transformer à l'aide de XSLT¹⁴.

Le mode de fonctionnement en mode application web adopté par les bases eXist et Xindice rend très aisé le branchement de ces bases sur des applications de publications telles que Cocoon¹⁵, ou à d'autres applications web.

Echanges

« Round-tripping »

Il s'agit de « l'aller-retour » des documents, à savoir la conformité d'un document, inséré puis récupéré dans la base, au document original.

Dans ce domaine, les bases de données XML natives assurent en théorie de meilleurs résultats qu'un stockage des données dans des tables¹⁶, puisqu'elles sauvegardent des éléments spécifiques au modèle XML¹⁷ (commentaires, ordres des noeuds enfants au sein d'un même parent...). D'autre part, la restitution de l'information sous forme de document depuis une base XML native est immédiate alors qu'elle nécessite une étape d'interrogation (jointures entre tables) et de mise en forme des résultats depuis une base de données relationnelle.

Les fichiers de métadonnées qui seront stockés au travers de la base ne nécessiteront pas une conservation à l'identique mais seulement la conservation des informations véhiculées au travers des noeuds textes et des attributs ainsi que la validité de la structure par rapport au schéma de données, ce qui devrait pouvoir être assuré par les deux technologies.

14 A titre d'exemple, l'implémentation de XQuery par eXist propose une extension permettant d'appliquer à la volée des transformations XSLT aux résultats des requêtes.

15 <http://cocoon.apache.org/> : construire des applications web comme avec des « Lego(tm) »...

16 On ne parle pas, bien sûr, d'un stockage « brut » du fichier XML (en tant que BLOB ou CLOB) qui assure, une conservation à l'identique du document.

17 Cela dit, nous avons constaté une modification des sections CDATA après un passage dans la base eXist.

Conclusion

Hormis l'indexation géographique qui n'est actuellement pas disponible ni à l'ordre du jour pour les bases de données XML natives, ces dernières remplissent globalement les fonctionnalités que l'on attend pour un serveur de métadonnées.

Inconvénients des bases XML natives

Les bases de données XML dans leur ensemble souffrent d'un manque de standardisation (dû essentiellement à la jeunesse de cette technologie) et le choix, à l'heure actuelle, d'une telle solution impliquera l'utilisation de technologies propriétaires (notamment pour l'administration, la définition d'index et sans doute pour certaines requêtes) pour sa mise en oeuvre, ce qui rendra plus difficile une éventuelle migration vers une autre base de données XML.

On l'a vu aussi, si certaines bases de données XML peuvent permettre d'effectuer des liens inter-documents (comme eXist), elles n'assureront pas pour autant les règles d'intégrité référentielle. L'utilisation de tels liens pour optimiser le stockage au sein d'une base XML n'est donc pas conseillée à moins de mettre en oeuvre soi-même les mécanismes en assurant l'intégrité.

Avantages des bases XML natives

L'avantage de l'utilisation des bases de données XML réside dans le fait qu'elles manipulent directement le format XML, ce qui permet de mettre rapidement en oeuvre des services web d'accès à de telles bases :

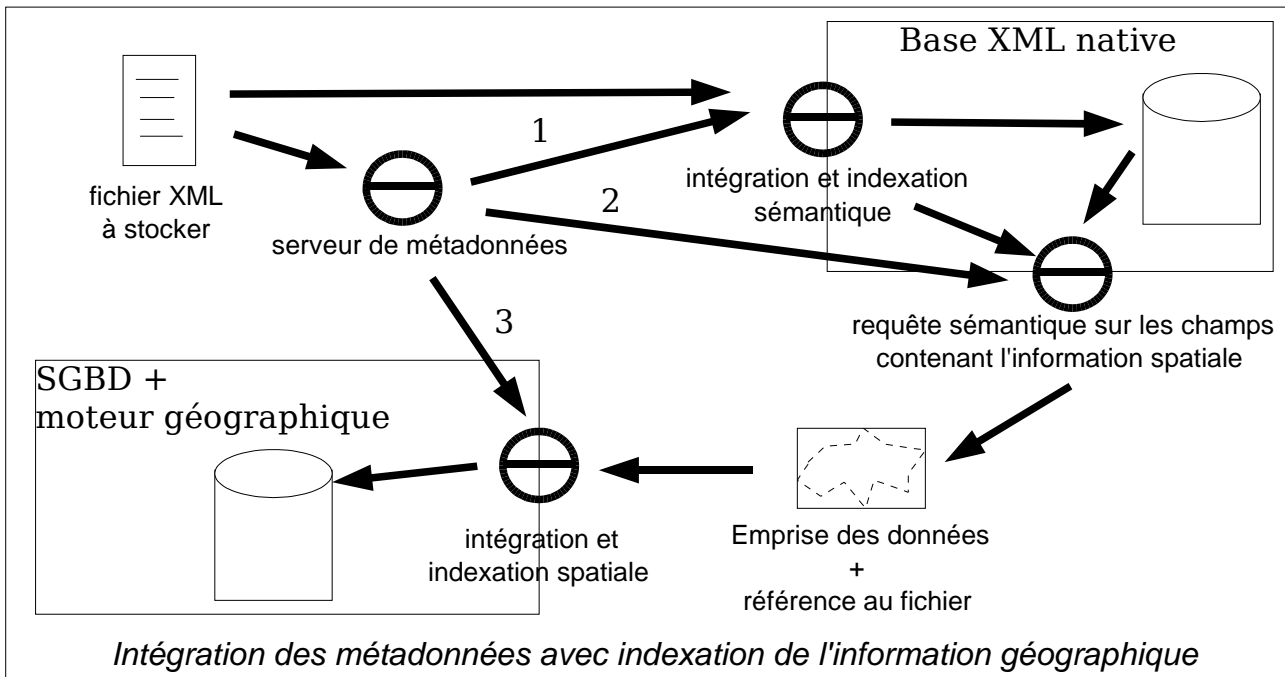
- le stockage des fichiers XML se fait directement (pas besoin de définir et d'écrire des règles de transformations XML<->relationnel) ;
- de même la restitution des documents est immédiate alors qu'il faut le reconstituer à partir des tables d'une base de données relationnelle ;
- la publication des résultats de requêtes au travers du web est très aisée à partir des bases de données XML.

Mise en oeuvre de l'indexation géographique

La mise en oeuvre de l'indexation géographique avec une base XML native implique d'utiliser une solution tierce offrant de telles possibilités telle qu'un SGBD-R avec moteur géographique dans lequel les informations relatives à l'emprise géographique des métadonnées seront insérées et indexées avec une référence au document correspondant.

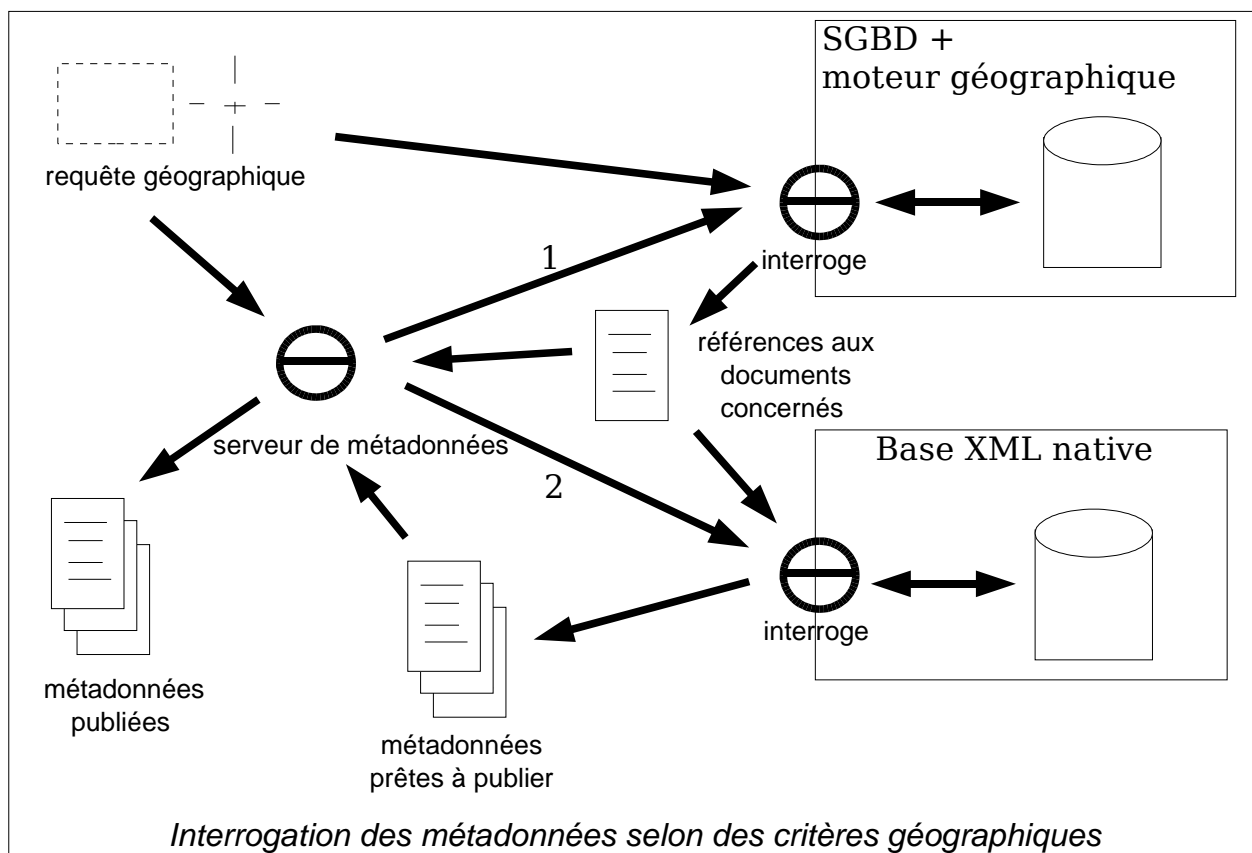
Indexation géographique

L'indexation géographique se ferait lors de l'intégration des documents XML selon le schéma de fonctionnement suivant : intégration du document dans la base XML native (1), récupération et mise en forme de l'information géographique (emprise) à l'aide de requêtes sémantiques sur le document XML (2) puis, indexation de cette information associée à la référence du document à l'aide du SGBD (3).



Requêtes géographiques

L'interrogation de la base de métadonnées selon des critères géographiques peut se dérouler ensuite selon le schéma suivant : récupération à l'aide de l'index géographique des références aux documents répondant aux critères de sélection (1), puis récupération de ces documents depuis la base XML native (2).



Glossaire

- XML** eXtensible Markup Language
version 1.0 (troisième édition) - Recommandation W3C - 4 Février 2004
(<http://www.w3.org/TR/REC-xml>)
version 1.1 - Recommandation W3C - 4 Février 2004
(<http://www.w3.org/TR/xml11>)
« Il s'agit d'un sous-ensemble de **SGML**¹⁸ dont le but est de permettre au SGML générique d'être transmis, reçu et traité sur le Web de la même manière que l'est HTML aujourd'hui. XML a été conçu pour être facile à mettre en oeuvre et interopérable avec SGML et HTML. »
- DTD** Document Type Definition
(<http://www.w3.org/TR/2004/REC-xml-20040204/#dt-doctype>)
La notion de DTD fait partie de la définition de XML. Elle permet de définir la structure d'un document XML en en définissant les balises (type d'élément, liste d'attributs, entités ou notations).
- XSD** XML Schema Definition
Recommandations W3C - 2 Mai 2001
(<http://www.w3.org/TR/xmlschema-0>)
Il s'agit d'un langage permettant de décrire la structure et les contraintes sur le contenu de documents XML. Ce langage est lui-même exprimé à l'aide de XML. Ses possibilités sont beaucoup plus étendues que celles offertes par les DTD.
- XSLT** eXtensible Stylesheet Language Transformations
version 1.0 Recommandations W3C - 16 Novembre 1999
(<http://www.w3.org/TR/xslt>)
Il s'agit d'un langage permettant de transformer des documents XML en d'autres documents XML. Il est exprimé à l'aide de XML.
- XPath** XML Path Language
version 1.0 - Recommandations W3C - 16 Novembre 1999
(<http://www.w3.org/TR/xpath>)
version 2.0 (définition commune avec XQuery 1.0) - Working Draft - 12 Novembre 2003
(<http://www.w3.org/TR/xpath-datamodel/>)
C'est un langage défini par le W3C permettant l'adressage d'informations au sein d'une arborescence XML. Cet adressage peut s'accompagner de prédicats permettant la sélection des informations en fonction de certains critères.

¹⁸ SGML est l'ancêtre (défini par l'ISO) de XML.

XQuery	<p>XML Query version 1.0 - Working Draft - 12 Novembre 2003 (http://www.w3.org/TR/xquery/) C'est un langage de requêtes en cours de définition par le W3C offrant une syntaxe permettant d'extraire des informations depuis des documents XML au travers de critères complexes.</p>
XInclude XLink XPointer	<p>XML Inclusions, XML Linking Language, XML Pointer Language (http://www.w3.org/TR/xinclude/) (http://www.w3.org/XML/Linking) Ces trois standards issus du W3C définissent des manières d'établir des liens entre documents ou fragments XML. Xpointer décrit les manières de référencer des fragments de XML, XInclude et XLink utilisent XPointer et décrivent les relations entre données XML, soit sous forme d'inclusion (XInclude) soit comme simple référence (XLink).</p>
XUpdate	<p>XML Update Language Working Draft du 14/09/2000 (http://www.xmldb.org/xupdate/) C'est un langage défini par l'initiative XML:DB permettant la mise à jour de documents XML. Il est exprimé à l'aide de XML.</p>
SiXDML	<p>Simple XML Data Manipulation Language Implémentation pour Xindice (http://sixdml.sourceforge.net/) Working Draft du 23/06/2002 (http://www.xmldb.org/sixdml/index.html) C'est un langage de requêtes défini par l'initiative XML:DB. Ce langage offre les fonctionnalités d'un DDL (Langage de définition de données) et d'un DML (langage de manipulation de données).</p>

Annexe A : Etude des bases XML natives : eXist et Xindice

logiciels utilisés

Les tests ont été effectués sur un PC Mono processeur cadencé à 1GHz avec 512Mo de RAM et Linux (Mandrake 9.0) comme système d'exploitation.

	eXist	Xindice
version	1.0b1 (2 février 2004)	1.1b3
URL	http://exist.sourceforge.net/	http://xml.apache.org/xindice/
mode de fonctionnement	serveur eXist tournant en mode application WEB derrière un moteur de servlet (Jetty) dédié (aucune autre application ni connection en dehors des tests)	application web derrière moteur de servlet tomcat (4.1.24) non dédié.
client	client fourni avec la distribution eXist lancé sur la même machine	client xindice fourni avec la distribution
protocole de communication	XML-RPC	XML-RPC

Pour tester ces deux bases, nous avons utilisé trois sources de données XML différentes :

1. demandes de prestations au serveur général
2. fichiers issus du logiciel TA (PVA)
3. métadonnées des scans de la série régionale

Tests sur des demandes de prestation

Il s'agit de fichiers de demandes de prestations effectuées au Serveur Général (novembre 2003 à février 2004). Tous conforme à une même DTD.

nombre : 665
taille moyenne : 8ko
taille totale : 5,4 Mo
Intérêt : tests de base (chargement, requêtes sur des fichiers de taille moyenne).

L'évaluation des temps pour eXist est la suivante :

- le temps de chargement des demandes de prestations comprend le temps de lancement et de fin du client (1 à 2 secondes) ;
- les autres temps sont des temps de réponse du serveur (lus dans les logs !).

Pour Xindice, les temps de réponse du serveur n'étant pas disponibles, tous les temps présentés ici tiennent compte du temps de lancement et de fin du client (1 à 2 secondes) ;

Chargement des fichiers dans la base

	eXist	Xindice
Chargement des demandes de prestations (5,4Mo répartis sur 665 fichiers)	1min 5s	3mn 52s

Récupération d'un document par sa clef (son nom) et sérialisation

fichier : dpsg2004-01-13-09-2527150-iso.xml

	eXist	Xindice
récupération document	~30ms	~5000 ms

Requête XPATH (1)

Recherche des demandes de prestations effectuées un jour donné (14/11/2003). Affiche le contenu du noeud `/Metadata/mdFileID` des documents résultats.

```
/Metadata/mdDateSt[text()='14/11/2003']/../mdFileID[text()]
```

Elle est effectuée avec et sans l'index sur le noeud `//mdDateSt`.

Remarque : pour eXist l'utilisation ou la non utilisation de l'index se fait en changeant l'opérateur utilisé dans le prédicat de sélection :

sans index : `[text()='14/11/2003']`

avec index : `[text()&='14/11/2003']`

Pour Xindice, les résultats présentés sont (censés être) obtenus avec un index...

	eXist	Xindice
requête XPATH (1)	380ms (8 résultats)	~45000ms (8 résultats) 45 secondes
requête XPATH (1) avec index	~20ms (0 résultats)	

Avec eXist, la requête XPath (1) avec index ne retourne pas les mêmes résultats (aucun résultat) que sans index (8 résultats). Cela semble être dû au fait que la valeur recherchée est une date ayant la forme 'JJ/MM/AAAA' alors que l'index « full text » de eXist indexe les mots des valeurs textuelles. Le champ date ne semble pas avoir été indexé comme un mot complet. Le comportement de l'indexation (choix des mots et des césures) est paramétrable; mais ce paramétrage n'a pas été testé.

Requête XPATH (2)

Recherche des noeuds `//resTitle` contenant le texte « BDPays »

```
//resTitle[contains(text(),'BDPays')]
```

Remarque : pour eXist l'utilisation ou la non utilisation de l'index se fait en changeant l'opérateur utilisé dans le prédicat de sélection. Exemple, pour la requête XPATH (2), le prédicat s'écrit :

sans index : [contains(text(), 'BDPays')]

avec index : [starts-with(text(), 'BDPays')]

Pour Xindice, les résultats présentés sont (censés être) obtenus avec un index...

	eXist	Xindice
requête XPATH (2)	~850ms (126 résultats)	~38000ms (126 résultats) (38 secondes)
requête XPATH (2) avec index	~360ms (126 résultats)	

Requête XPATH croisée

Recherche des demandes de prestations effectuées un jour donné (03/11/2003) sur un produit donné (Georoute). Affiche le contenu du noeud /Metadata/mdFileID des documents résultats.

```
/Metadata/mdDateSt[text()='03/11/2003']/../svIdInfo/operatesOn/idCitation/resTitle
[starts-with(text(),'Georoute')]/../../../../../../../../mdFileID[text()]
```

	eXist	Xindice
requête XPath croisée	~360ms (5 résultats)	~63000ms(5 résultats) 1mn 3s

Requêtes XQuery

Affiche les noeuds /Metadata/mdFileID et /Metadata/svIdInfo/idAbs des documents dont le noeud /Metadata/mdDateSt a la valeur 14/11/2003

```
let $collection := xmldb:collection($collURI, "gilles", "gilles1"),
    $resources := coll:list-resources($collection)
for $doc in $resources
  where contains(concat("/db/test/dpsg/", $doc)) /
Metadata/mdDateSt, "14/11/2003")
return
  <resultat>
    {document(concat("/db/test/dpsg/", $doc))/Metadata/mdFileID}
    {document(concat("/db/test/dpsg/", $doc))/Metadata/svIdInfo/idAbs}
  </resultat>
```

	eXist	Xindice
requête XQuery	~1800ms (3 résultats)	

Mise à jour au travers d'XUpdate

Insertion d'un commentaire avant le noeud //mdFileID :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xupdate:modifications version="1.0"
  xmlns:xupdate="http://www.xmldb.org/xupdate">
  <xupdate:insert-before select="//mdFileID">
    <xupdate:comment>identifiant des metadonnees...</xupdate:comment>
  </xupdate:insert-before>
</xupdate:modifications>
```

	eXist	Xindice
mise à jour XUpdate sur un document	~400ms	~5000ms 5 secondes

Fichiers issus du logiciel TA

Les fichiers issus du logiciel TA décrivant les missions aériennes de 1922 à 2002 sont répartis sur deux types de fichiers : des fichiers « chantiers » et des fichiers « archives ». Nous avons organisé ces fichiers dans des collections différentes selon leur antériorité (ou non) à l'année 1990, car les plus récents sont les plus volumineux (certains fichiers peuvent avoir une taille de l'ordre de 4 à 5 Mo) :

		fichiers « chantier »	fichiers « archives »
de 1922 à 1989	nombre	19205	19205
	taille totale	694 Mo	350 Mo
	taille moyenne	36 Ko	18 Ko
de 1990 à 2002	nombre	2315	2315
	taille totale	586 Mo	305 Mo
	taille moyenne	253 Ko	131 Ko

Intérêt: tester les capacités d'hébergement sur des gros volumes de données (taille des données et nombre de fichiers) et les temps de réponse sur de telles données.

Chargement dans la base

Le chargement des PVA dans eXist a été testé de deux manières :

- au travers de l'interface XML-RPC sur un serveur eXist derrière un moteur de servlet dédié ;
- en « local » (ou mode « embarqué ») : le client attaque directement la base dont une instance est lancée sur la même machine virtuelle que celle du client.

Les chargements des fichiers de PVA sur Xindice ont échoué pour les fichiers de gros

volume, ce qui correspond à un comportement connu de Xindice.
 Les tests des PVA avec cette base ont donc perdu tout leur intérêt...

L'évaluation des temps pour eXist est la suivante :

- le temps de chargement des PVA comprend le temps de lancement et de fin du client (1 à 2 secondes) ;
- les autres temps sont des temps de réponse du serveur (lus dans les logs !)

	eXist	Xindice
Chargement des archives anciennes (350Mo répartis sur 19205 fichiers)	XML-RPC : 2h 25min 3s Local : 0h 21min 50s	
Chargement des chantiers anciens (694Mo répartis sur 19205 fichiers)	XML-RPC : 2h 57min 24s Local : 0h 58min 53s	
Chargement des archives nouvelles (305Mo répartis sur 2315 fichiers)	XML-RPC : 0h 46min 21s Local : 0h 17min 25s	
Chargement des chantiers nouveaux (586Mo répartis sur 2315 fichiers)	XML-RPC : 1h 21min 32s Local : 0h 48 min 25s	

Le chargement d'un volume important de données se fait beaucoup plus rapidement en mode embarqué qu'au travers de l'interface XML-RPC, ce qui correspond aux préconisations de la documentation de eXist. Cela permet de préfigurer un mode opératoire lors d'une initialisation de la base, où il pourra être intéressant de charger les données en mode embarqué avant de lancer le serveur.

Récupération d'un document par sa clef (son nom) et sérialisation

- un document de taille anodine (30Ko)
- un document de grande taille (4,2Mo)

	eXist	Xindice
récupération d'un document de taille anodine (30Ko)	~100ms	
récupération d'un gros document (4,2Mo)	~20000ms (20 secondes)	

Requêtes XPATH

Recherche des documents de type chantier récent dont le noeud `//commanditaire` contient une certaine valeur. Valeurs testées : `DIREN` (7 résultats), `DISNEYLAND` (1 résultat), `GOURDON` (4 résultats) et `CEVA` (24 résultats). Affiche la valeur du noeud `//commanditaire`

```
//commanditaire[contains(text(),'CEVA')]
```

Remarque : Avec eXist, l'utilisation de l'index nécessite une utilisation d'un opérateur différent de l'opérateur XPATH standard :

sans index : //commanditaire[contains(text(),'CEVA')]

avec index : //commanditaire[text()&='CEVA']

	eXist	Xindice
requête XPATH sur chantiers nouveaux (sans index)	~1600ms	
requête XPATH sur chantiers nouveaux (avec index)	~40ms	

Requêtes XQuery

Recherche, parmi les chantiers récents (collection /db/test/pva/new/chantiers), ceux dont le commanditaire contient le mot 'DIREN'. Affiche pour chaque résultat, les dates des missions aériennes, le nom générique du chantier et sa désignation :

```
let $collection := xmldb:collection($collURI, "gilles", "gilles1"),
    $resources := coll:list-resources($collection)
for $doc in $resources
  where contains(document(concat("/db/test/pva/new/chantier/", $doc))/
    commanditaire, 'DIREN ')
  return
  <resultat>
    {document(concat("/db/test/pva/new/chantier/", $doc))/TA/chantier/vol/mission}
    {document(concat("/db/test/pva/new/chantier/", $doc))/
TA/chantier/nom_generique}
    {document(concat("/db/test/pva/new/chantier/", $doc))/TA/chantier/designation}
  </resultat>
```

	eXist	Xindice
requête XQuery	~6500ms	

Métadonnées des scans de la série régionale

```
nombre : 20
taille moyenne : 8 Ko
taille totale : 156 Ko
Intérêt : tester une organisation des fichiers de métadonnées avec relations données -> produit.
```

Les documents ont été chargés dans une arborescence de collections de type :

```
metadata/
  produits/
    -> sr250k.xml (fichier de métadonnées sur la série rouge)
  data/
    sr250k/
      -> tous les fichiers de métadonnées sur les cartes de la série rouge
```

les fichiers de métadonnées contiennent tous un noeud :

```
<mdContact>
  <rpOrgName>IGN</rpOrgName>
  <role><RoleCd value="originator"></RoleCd></role>
</mdContact>
```

faisant référence au producteur des données. Cette information a été maintenue au sein du fichier de métadonnées de la série rouge (sr250k.xml) et remplacée, pour les autres fichiers de métadonnées par une référence xinclude :

```
<xi:include href="/db/test/metadata/produits/sr250k.xml#xpointer(//mdContact)"/>
```

avec la définition du namespace suivant :

```
xmlns:xi="http://www.w3.org/2001/XInclude"
```

Chargement (avec références Xinclude)

Le chargement de tels documents avec eXist ne pose pas de problèmes, sauf si on demande qu'une validation au regard du schéma soit effectuée lors de l'insertion, auquel cas, eXist ne déploie pas la référence xinclude et le document n'est donc pas considéré comme valide.

Récupération d'un document

Avec eXist, les références Xinclude sont bien déployées si on le souhaite lorsque l'on récupère un document. Par contre, rien ne permet de vérifier que ces références font bien référence à un noeud existant (non existence ou suppression dans la base) : les noeuds référencés ne sont alors remplacés par rien.

Annexe B : exemples de fichiers XML à gérer

Métadonnées (implémentation proposée par le Projet)

Le fichier qui suit est celui des métadonnées de la série régionale selon l'implémentation de la norme ISO 19115 proposée par le Projet.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!--
<!DOCTYPE Metadata SYSTEM "../META-INF/iso19115_DIS.dtd"
-->
<Metadata xmlns="http://interop.ign.fr/iso/metadata/iso19115"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://interop.ign.fr/iso/metadata/iso19115
                             http://interop.ign.fr/iso/metadata/iso19115_DIS.xsd">
<mdFileID>http://foster.ign.fr/chapitre12/metadata/sr250k.xml</mdFileID>
<mdLang><languageCode value="fr"/></mdLang>
<mdChar><CharSetCd value="8859part1"/></mdChar>
<mdHrLv><ScopeCd value="series"></ScopeCd></mdHrLv>
<mdContact><rpOrgName>IGN</rpOrgName><role><RoleCd
value="originator"></RoleCd></role></mdContact>
<mdDateSt>20030610</mdDateSt>
<dataIdInfo>
<idCitation>
<resTitle>
<![CDATA[
<PtFreeText>
  <textGroup>
    <PtGroup>
      <PtLangCode>fr</PtLangCode>
      <PtText>Série Régionale</PtText>
    </PtGroup>
  </textGroup>
  <textGroup>
    <PtGroup>
      <PtLangCode>en</PtLangCode>
      <PtText>"Série Régionale" Map serie</PtText>
    </PtGroup>
  </textGroup>
</PtFreeText>
]]>
</resTitle>
<resAltTitle>
<![CDATA[
<PtFreeText>
  <textGroup>
    <PtGroup>
      <PtLangCode>fr</PtLangCode>
      <PtText>Série Rouge</PtText>
    </PtGroup>
  </textGroup>
  <textGroup>
    <PtGroup>
      <PtLangCode>en</PtLangCode>
      <PtText>Red Map Serie</PtText>
    </PtGroup>
  </textGroup>
</PtFreeText>
]]>
</resAltTitle>
<resRefDate>
```

```

<refDate>20030610</refDate>
<dateType><DateTypCd value="creation"></DateTypCd></dateType>
</resRefDate>
</idCitation>
<idAbs>
<![CDATA[
<PtFreeText>
  <textGroup>
    <PtGroup>
      <PtLangCode>fr</PtLangCode>
      <PtText>18 cartes régionales à l'échelle du 1:250000 (1cm=2,5km) au
1:280000 indispensables pour parcourir en voiture les régions et découvrir
leurs richesses (routes, châteaux, abbayes,...). Idéale pour trouver sa route
la série Régionale propose une version très précise du réseau routier
régional.</PtText>
    </PtGroup>
  </textGroup>
  <textGroup>
    <PtGroup>
      <PtLangCode>en</PtLangCode>
      <PtText>18 maps covering french regions at scales between 1:250000
(1cm=2,5km) and 1:280000, worth crusing theses regions and discovering their
wealth (roads, castels, abbeys,...). Worth using for finding his way the série
Régionale proposes a very precise local road network.</PtText>
    </PtGroup>
  </textGroup>
</PtFreeText>
]]>
</idAbs>
<resConst><LegConsts>
<useLimit>
<![CDATA[
<PtFreeText>
  <textGroup>
    <PtGroup>
      <PtLangCode>fr</PtLangCode>
      <PtText>COPYRIGHT IGN - Paris 2000 Edition 2. Toute reproduction ou
adaptation sous quelque forme et par quelque procédé que ce soit, même
partielle, interdite pour tous pays.</PtText>
    </PtGroup>
  </textGroup>
  <textGroup>
    <PtGroup>
      <PtLangCode>en</PtLangCode>
      <PtText>COPYRIGHT IGN - Paris 2000 second edition. All rights
reserved.</PtText>
    </PtGroup>
  </textGroup>
</PtFreeText>
]]>
</useLimit>
<useConsts><RestrictCd value="copyright"/></useConsts>
</LegConsts></resConst>
<dsFormat>
<formatName>TIFF</formatName>
<formatVer>GEO</formatVer>
</dsFormat>
<graphOver>
<bgFileName>
<![CDATA[

]]>

```

```

</bgFileName>
<bgFileDesc>
<![CDATA[
<PtFreeText>
  <textGroup>
    <PtGroup>
      <PtLangCode>fr</PtLangCode>
      <PtText>Tableau d'assemblage de la série Régionale</PtText>
    </PtGroup>
  </textGroup>
  <textGroup>
    <PtGroup>
      <PtLangCode>en</PtLangCode>
      <PtText>"Série Régionale" Map serie sheets</PtText>
    </PtGroup>
  </textGroup>
</PtFreeText>
]]>
</bgFileDesc>
<bgFileType>image/gif</bgFileType>
</graphOver>
<dataLang><languageCode value="fr"/></dataLang>
<tpCat><TopicCatCd value="biota"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="boundaries"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="elevation"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="inlandWaters"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="location"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="oceans"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="structure"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="transportation"></TopicCatCd></tpCat>
<tpCat><TopicCatCd value="utilitiesCommunication"></TopicCatCd></tpCat>
<geoBox><westBL>-
5.37589</westBL><eastBL>10.51522</eastBL><southBL>41.11495</southBL><northBL>51
.10805</northBL></geoBox>
</dataIdInfo>
<refSysInfo><RefSystem><refSysID>
<identAuth>
<resTitle>EPSG</resTitle>
<resRefDate><refDate>20010101</refDate><dateType><DateTypCd
value="publication"/></dateType></resRefDate>
</identAuth>
<identCode>4326</identCode>
</refSysID></RefSystem></refSysInfo>
</Metadata>

```

Demandes de prestations

Le fichier qui suit est celui d'une demande de prestation au Serveur Général tel qu'il est généré et envoyé à ce dernier par l'application de saisie de demandes de prestations.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Metadata>
<mdFileID>dpsg2004-01-26-15-2668339</mdFileID>
<mdLang>
<languageCode value="fr"></languageCode>
</mdLang>
<mdChar>
<CharSetCd value="8859part1"></CharSetCd>
</mdChar>
<mdContact>
<rpIndName>P. Priet</rpIndName>
<rpOrgName>SBV/PBXD</rpOrgName>
<rpCntInfo>
<cntPhone>
<voiceNum>8151</voiceNum>
</cntPhone>
<cntAddress>
<eMailAdd>pascal.priet@ign.fr</eMailAdd>
</cntAddress>
</rpCntInfo>
<role>
<RoleCd value="pointOfContact"></RoleCd>
</role>
</mdContact>
<mdDateSt>26/01/2004</mdDateSt>
<distInfo>
<distributor>
<distorCont>
<rpOrgName>Serveur G&#233;n&#233;ral</rpOrgName>
<role>
<RoleCd value="processor"></RoleCd>
</role>
</distorCont>
<distorOrdPrc>
<resFees>xto04</resFees>
<planAvDtTm>10/02/2004</planAvDtTm>
</distorOrdPrc>
</distributor>
</distInfo>
<svIdInfo>
<idCitation>
<resTitle>Demande de prestation au Serveur General</resTitle>
<resAltTitle>SBV/04/30</resAltTitle>
<resRefDate>
<refDate>26/01/2004</refDate>
<dateType>
<DateTypCd value="creation"></DateTypCd>
</dateType>
</resRefDate>
<citId>dpsg2004-01-26-15-2668339</citId>
</idCitation>
<idAbs>Demande de prestation numero : dpsg2004-01-26-15-2668339, effectu&#233;e
le : 26/01/2004, par : P. Priet, concerne le produit : BDPays.Operations &#224;
effectuer : obtention des donnees par desarchivage, traduction vers MifMid,
conditionnement, livraison sur support deconnecte</idAbs>
<idPoC>
<rpIndName>M. Leboeuf</rpIndName>
```

```

<role>
<RoleCd value="distributeur"></RoleCd>
</role>
</idPoC>
<serviceType>DiffBDTopoPays</serviceType>
<accessProperties>
<resFees>xto04</resFees>
<planAvDtTm>10/02/2004</planAvDtTm>
<ordInstr>lisezmoi.txt04030 ; Spec V1.2 ; Lambert 2 &#233;tendu ; TRADUIRE
AUSSI EN SHAPEFILE 2D &amp; 3D ET GEOCONCEPT</ordInstr>
</accessProperties>
<containsOperations>
<operName>Desarchivage</operName>
<listDCP>
<DCPlistCd value="WebServices"></DCPlistCd>
</listDCP>
<connectPoint>
<linkage>...</linkage>
</connectPoint>
<parameters>
<name>NomEchFeiv</name>
<optionality>Conditional</optionality>
<repeatability>oui</repeatability>
<prmValue>D08701</prmValue>
</parameters>
</containsOperations>
<containsOperations>
<operName>TradMifMid</operName>
<listDCP>
<DCPlistCd value="WebServices"></DCPlistCd>
</listDCP>
<connectPoint>
<linkage>...</linkage>
</connectPoint>
<parameters>
<name>Theme</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue>toponymie</prmValue>
</parameters>
</containsOperations>
<containsOperations>
<operName>Conditionnement</operName>
<listDCP>
<DCPlistCd value="WebServices"></DCPlistCd>
</listDCP>
<connectPoint>
<linkage>...</linkage>
</connectPoint>
<parameters>
<name>sauvlan</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>oui</prmValue>
</parameters>
<parameters>
<name>joinReadMe</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>oui</prmValue>
</parameters>
</containsOperations>
<containsOperations>

```

```

<operName>Duplication</operName>
<listDCP>
<DCPlistCd value="WebServices"></DCPlistCd>
</listDCP>
<connectPoint>
<linkage>...</linkage>
</connectPoint>
<parameters>
<name>typeSupport</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>cdRom</prmValue>
</parameters>
<parameters>
<name>livrerA</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>demandeur</prmValue>
</parameters>
<parameters>
<name>nbExempl</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>1</prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue>BD TOPO PAYS</prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue>Formats SHAPEFILE,</prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue>MIF/MID, GeoConcept</prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue></prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue>TOPONYMIE 87</prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>
<optionality>Optional</optionality>
<repeatability>oui</repeatability>
<prmValue></prmValue>
</parameters>
<parameters>
<name>ligneLabel</name>

```



```

</parameters>
<parameters>
<name>titreJaquette</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>BD TOPO PAYS</prmValue>
</parameters>
<parameters>
<name>titreTranche</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>BD TOPO PAYSTOPONYMIE 87</prmValue>
</parameters>
<parameters>
<name>docPDF</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>oui</prmValue>
</parameters>
<parameters>
<name>niveauIso</name>
<optionality>Optional</optionality>
<repeatability>non</repeatability>
<prmValue>iso9660deux</prmValue>
</parameters>
</containsOperations>
<operatesOn>
<idCitation>
<resTitle>BDPays</resTitle>
<resRefDate>
<refDate>...</refDate>
<dateType>
<DateTypCd value="publication"></DateTypCd>
</dateType>
</resRefDate>
</idCitation>
<idAbs>...</idAbs>
<dataLang>
<languageCode value="fr"></languageCode>
</dataLang>
<tpCat>
<TopicCatCd value="imageryBaseMapsEarthCover"></TopicCatCd>
</tpCat>
</operatesOn>
</svIdInfo>
</Metadata>

```