POURQUOI ET COMMENT ÉCRIRE LA STRUCTURE D'UN JEU DE DONNÉES GÉOGRAPHIQUE

par Sandrine Balley

Laboratoire COGIT
Institut géographique national,
2-4 avenue Pasteur 94165 Saint-Mandé Cedex
sandrine.balley@ign.fr

Introduction

Les travaux de recherches sur l'utilisabilité des données géographiques [Hunter et al. 2003] témoignent de la nécessité et de la difficulté d'adapter un jeu de données à l'application prévue par un utilisateur. Le contenu, la qualité [Grum et Vasseur 2004], mais aussi la structure du jeu [Balley 2005] sont en cause. Avant d'exploiter les données, l'utilisateur doit cerner la structure de production des données existantes grâce à des métadonnées, identifier la structure que nécessite son application et évaluer la compatibilité de ces structures. Des outils de pré-traitement sont ensuite nécessaires pour adapter la structure du jeu fourni. Si les données et l'application sont accédées sous la forme d'applications Web interopérables, cette étape de pré-traitement doit être transparente et automatique.

La notion de structure est complexe et se formalise difficilement via les modèles de métadonnées existants. De plus, les structures de production et d'application ne sont pas décrites par le même type de métadonnées. Cet article étudie les modèles permettant de décrire une structure et souligne les enrichissements nécessaires dans une optique d'adaptation des structures. Dans la première partie nous proposons une définition de la notion de structure d'un jeu de données. La deuxième partie concerne la description des structures de production, que nous appelons structures source. La troisième partie concerne la description des structures d'application, que nous appelons structures cible. La quatrième partie propose des pistes pour la conception et l'exécution d'un processus automatique d'adaptation de structure précédant l'exploitation des données.

1 Qu'est-ce que la structure d'un jeu de données géographique ?

Nous appelons structure d'un jeu de données l'ensemble des schémas et règles caractérisant un jeu à différents niveaux d'abstraction, détaillés plus loin. La complexité du processus de production des données géographiques n'est pas étrangère à la complexité de leur structure : chaque étape de ce processus obéit à des règles liées au caractère spatial de l'information représentée et aux spécificités techniques des SGBD spatiaux. Ces règles participent à la notion de structure.

La figure 1 décompose le processus de production menant du monde réel aux données géographiques, sans tenir compte des phases instrumentées d'acquisition de données brutes. Chaque étape (représentée dans un cadre aux angles arrondis) est documentée par un ou plusieurs éléments (introduits par une flèche en pointillés et représentés dans un cadre gris).

- L'étape de catégorisation consiste à définir les concepts du monde réel à représenter (ex. le concept de 'route'). Elle est documentée par un *thesaurus* ou une *ontologie* représentant l'univers du discours.
- L'étape de sélection consiste à délimiter, pour chaque concept, les entités du monde réel qui devront figurer dans le jeu. Elle est documentée *par des règles de saisie* (ex. « seulement les routes classées administrativement » ou « les culs-de-sac d'une longueur de 400m au moins »).
- L'étape de modélisation décide de la représentation qui sera faite de chaque concept du monde réel dans les données, indépendamment de toute implémentation. Cette étape est documentée dans un *schéma conceptuel*[1] définissant des classes (ex. la classe

1 ou schéma d'application dans la terminologie normalisée ISO/OGC.

'Tronçon de route'), leurs attributs et leurs relations. Cette étape repose aussi sur *des règles de saisie* spécifiant comment observer les entités du monde réel (ex. « considérer l'axe de la route »), et comment les représenter en tant qu'objets géographiques conformes au schéma conceptuel (ex. « une polyligne sectionnée à chaque croisement et à chaque changement de propriété »). De plus, *des règles d'intégrité* contraignent les objets de la base (ex. « les objets de la classe 'Tronçon de route' doivent former un graphe »).

- L'étape d'implémentation consiste à définir *un schéma logique* dans la plate-forme prévue pour accueillir les données. Il peut s'agir de tables et de colonnes dans le cas d'un SGBD relationnel, de classes Java et de champs dans le cas d'une plate-forme de développement orientée-objet, ou encore d'un schéma XML composé de *featuretypes* dans le cas d'un stockage en GML.

- L'étape d'implémentation donne également lieu à un schéma physique décrivant l'organisation des données en terme de stockage en machine. Il s'agit le plus souvent de systèmes de fichiers.

Chacun des éléments évoqués en italique dans cette section décrit une partie de la structure d'un jeu de données. La section suivante présente une courte synthèse des modèles permettant de formaliser ces éléments de structure.

2 Comment décrire la structure d'un jeu de données géographique ?

2.1 Modèles existants

Les producteurs de données décrivent précisément la structure des jeux qu'ils produisent dans des documents textuels appelés spécifications de produits. Ces documents, quoique très riches, constituent une métadonnée peu utilisable : ils sont parfois confidentiels, trop volumineux et techniques pour des utilisateurs humains peu habitués aux données géographiques, mais insuffisamment formalisés pour une interprétation automatique. La formalisation des spécifications textuelles est étudiée par [Gesbert 2005] et [Abadie et al. 2006]. Malheureusement, il n'existe pas de modèle permettant de formaliser une structure dans son ensemble. Nous présentons dans la suite de cette section les principaux modèles standard et non standard permettant de décrire des éléments de structure. Comme ils sont de plus en plus utilisés, nous insistons sur les modèles standard de la gamme ISO19 100. Nous organisons notre présentation par niveau d'abstraction en traitant successivement les niveaux sémantique, conceptuel, logique et physique.

Il n'existe pas de modèle pour la définition de catégories d'entités du monde réel. La norme ISO 19 131 - Data Product Specifications - propose de rassembler ces catégories et « toute autre information utile » dans un champ en texte libre. La norme ISO19 115 - Metadata (champ Topic Category du package MD – Data Identification) - propose une liste de mots-clés identifiant le contenu général des données. La norme ISO19 110 - Feature Cataloguing - permet de préciser sous forme de texte libre les concepts du monde réel couverts par chaque type d'objet de la base constituant une entrée du catalogue. Il s'agit cependant d'une approche inversée : la description est attachée à des classes de schéma et non pas à des catégories du monde réel. Soulignons qu'une norme en cours de préparation, ISO 19 150 - Ontology - devrait améliorer cette situation dans les mois à venir. Hors du domaine spécifiquement géographique, les modèles formels OWL[2] ou DAML[3] permettent de créer des ontologies.

Les règles de saisie (sélection, observation et représentation des entités du monde réel) ne peuvent être décrites que sous forme textuelle dans les champs des normes ISO 19 131 et ISO19 110. Plus spécifiquement, le champ DPS – Data Capture Information de la norme ISO 19 131 est dédié aux règles d'observation et de représentation. Des modèles non standard concernant ces règles ont été proposés par [Gesbert 2005] [Christensen 2006] dans une optique d'intégration de données. [Gesbert 2005] les formalise comme des relations entre l'univers du discours et le schéma conceptuel.

Le schéma conceptuel peut être formalisé en détails grâce au General Feature Model proposé par la norme ISO19 109 - Application Schema. Les schémas conceptuels UML utilisent notamment les stéréotypes FeatureType (classe), Feature Attribute (attribut) et AssociationType (association). À chaque classe peuvent être attachées des contraintes d'intégrité. Aucun langage de règle n'est cependant pro-

2 http://www.w3.org/TR/owl-features/
3 http://www.daml.org

posé : il est possible d'utiliser du texte ou des modèles existants, comme Object Constraint Language et son extension géographique Spatial OCL [Pinet et al. 2004]. Dans le General Feature Model, les attributs de classes peuvent être de type simple (texte, booléen, entier, etc.), d'un type complexe défini par une autre classe du schéma ou d'un type spatial défini dans la norme ISO 19 107-Spatial Schema (ex. polyligne, surface complexe, arc orienté, etc.). De nombreux formalismes non standards comme MADS [Parent et al. 2006], UML+PVL [Bédard et al. 2004] ou CONGOO [Pantazis et al. 1996] permettent de décrire des schémas conceptuels géographiques. Ils offrent de nombreux avantages : les deux premiers autorisent la représentation multiple et sont supportés par un assistant de génie logiciel. Le dernier définit une gamme complète de contraintes d'intégrité topologiques. Le modèle de métadonnées ISO 19 115 autorise la représentation non standard d'un schéma conceptuel dans le package MD - Application Schema Information.

Il n'existe pas de formalisme générique pour décrire un schéma logique. Chaque plate-forme a son propre modèle qui n'est pas toujours explicite. L'OGC définit un schéma logique standard orienté-objet reposant sur la notion de *feature* (objet géographique). Des implémentations en sont proposées en Java, en SQL pour les SGBD relationnels, et en XML pour le format GML.

Le schéma physique, qu'il est généralement inutile de connaître en détails pour exploiter les données, est le moins décrit. Il est évoqué dans le modèle de métadonnées ISO 19 115 à travers les packages MD – Data Identification et MD - Format, mais aucun formalisme n'y est proposé.

La figure 2 synthétise cette section. Les éléments de structure y sont représentés à gauche et les modèles standard à droite. Une flèche noire indique qu'un élément de structure peut être formalisé, une flèche grise qu'il peut être représenté de façon non formelle. Les traits pointillés concernent les modèles de description en cours d'élaboration. Cette figure met en évidence la difficulté de représenter la structure d'un jeu de données géographiques : un modèle intégré prenant en compte tous les éléments de structure serait utile. C'est l'objet de la section suivante.

2.2 Un modèle intégré pour la description de la structure d'un jeu de données géographique

Comme présenté en section précédente, les producteurs souhaitant documenter la structure de leurs données peuvent soit publier leurs spécifications textuelles, soit construire des descriptions par morceaux en utilisant plusieurs modèles standard ou non-standard. Dans les deux cas, il est difficile de proposer une description cohérente et aisément compréhensible.

Dans le cadre de travaux sur la restructuration [Balley 2007], nous avons proposé un modèle générique dont le noyau est représenté en figure 3. Cette proposition répondait au constat que seul un modèle intégré explicitant les liens entre les différents niveaux de structure permettrait de manipuler une structure tout en préservant sa cohérence. Le modèle lie donc les différents niveaux d'abstraction de la structure d'un jeu de données :

- Les éléments de schéma conceptuel (ex. une classe) correspondent à des concepts du monde réel par le biais des règles de saisie.
- Les éléments de schéma logique (ex. une table relationnelle) correspondent à des éléments de schéma conceptuel via des règles de projection du formalisme conceptuel vers la plate-forme choisie pour l'implémentation.
- Les éléments de schéma physique (ex. un fichier) correspondent aux éléments de schéma logique via des règles d'encodage spécifiques au format de stockage.

Nous ne proposons aucun nouveau formalisme pour décrire ces niveaux de structure : notre modèle de base peut être spécialisé par n'importe quel modèle représentant des concepts du monde réel, des règles ou des schémas. Pour les besoins de notre application, nous avons choisi le General Feature Model pour le niveau conceptuel, le schéma de la plate-forme objet-relationnelle GeOxygene[4] étendant les standards OGC pour le niveau logique, et nous avons spécifié les règles de projection correspondantes.

Dans les parties 1 et 2 de cet article, nous avons défini la notion de structure d'un jeu de données géo-

4 GeOxygene est une plate-forme Java objet-relationelle développée au laboratoire COGIT. Elle est accessible sous la licence LGPL sur http://oxygene-project.sourceforge.net.

graphique et exploré les modèles existants pour sa formalisation. Les utilisateurs potentiels de données géographiques peuvent utiliser ces descriptions pour mesurer la pertinence de la structure d'un jeu diffusé au regard de l'application prévue. En d'autres termes, ils doivent évaluer la distance entre une structure source et une structure cible, notion que nous présentons en troisième partie.

3 Structures cible

3.1 Qu'est-ce qu'une structure cible ?

Nous considérons dans cette section l'application prévue par un utilisateur, par exemple la réalisation d'une carte, la généralisation de données ou une application de navigation. Ces applications requièrent des outils informatiques spécifiques, respectivement un service de représentation de données, une plate-forme complexe de généralisation et un algorithme spécifique de calcul d'itinéraire.

Comme exposé dans [Bucher et Balley 2007], toute application ou outil informatique génère des contraintes sur la structure des données d'entrée. Nous appelons structure cible une structure qui respecte toutes ces contraintes. Nous décrivons ci-dessous les quatre principaux types de contraintes que nous illustrons par l'exemple du calcul d'itinéraire.

Le premier type de contrainte se rapporte au schéma physique : l'outil doit être capable de lire les données d'entrée. Par exemple, l'outil de calcul d'itinéraire doit pouvoir accéder à un répertoire contenant deux fichiers nommés *route.shp et nœud.shp.*

Le deuxième type de contrainte concerne le schéma logique des données source. L'outil doit en effet charger les données dans un « schéma de manipulation » propre à l'application et à la plate-forme. Par exemple, l'outil de calcul d'itinéraire manipule les données sous la forme d'un objet *Java Graphe* construit à partir des routes et des nœuds d'entrée. Pour cela, chaque objet route doit porter un identifiant, une primitive géométrique de type *GM - Linestring* et un attribut de *type float* intitulé *poids*.

Le troisième type de contrainte concerne les règles de représentation et d'intégrité du jeu d'entrée. L'outil de calcul d'itinéraire peut, par exemple, nécessiter, pour la création de l'objet *Graphe*, que les routes et les nœuds d'entrée soient connectés topologiquement sans que cela apparaisse explicitement sous forme d'association dans le schéma conceptuel. De plus, l'outil peut avoir des comportements indésirables dont le développeur lui-même n'est pas conscient. Par exemple, l'algorithme de calcul d'itinéraire peut échouer dès qu'il rencontre un objet ayant un attribut *poids* égal à zéro. Un certain algorithme de généralisation de routes peut fonctionner mais donner de mauvais résultats sur des tronçons très sinueux. Ce type de contrainte est appelé pré-condition.

Le quatrième type de contrainte se rapporte à l'univers du discours : l'application manipule les données en leur assignant des rôles. Par exemple, l'objet *Graphe* manipulé joue le rôle de « réseau routier ». Ce concept doit pouvoir être localisé dans l'ontologie utilisée par le jeu source. De même, l'attribut *poids* des arcs du graphe peut être utilisé par l'algorithme comme un indice de vitesse améliorant l'identification des plus courts chemins. Le résultat du calcul serait incorrect si l'attribut *poids* des données d'entrée représentait le poids maximal en charge des véhicules autorisés à emprunter le tronçon.

Les quatre types de contraintes énumérés dessinent la structure cible d'une application. La section suivante étudie leur description.

3.2 Comment décrire une structure cible ?

Les modèles dédiés à la description de traitements sont mieux adaptés aux structures cible que les modèles dédiés à la description de données. De nombreux outils et applications sont documentés textuellement ou oralement par leur concepteur. C'est souvent le cas pour les applications très complexes et pour celles qui ne sont exploitées qu'en interne dans une organisation [Abd-el-Kader et Bucher 2006]. Néanmoins, tout outil mis en ligne et destiné à être invoqué par n'importe quel utilisateur ou application doit être décrit de façon plus formelle. C'est pourquoi la plupart des modèles évoqués ci-dessous viennent du domaine des services Web.

Le modèle WSDL[5] fournit le cadre de description minimal pour invoquer un service Web. Il définit le nom et le type des variables d'entrée. Des annota-

5 http://www.w3.org/TR/wsdl

tions sur la sémantique des variables peuvent être ajoutées sous la forme d'énoncés WSDL-S en texte libre [Akkiraju et al. 2005]. Ce modèle est cependant insuffisant pour décrire des variables aussi complexes que des jeux de données géographiques.

La spécification WPS (Web Processing Service) de l'Open Geospatial Consortium concerne les services Web de traitement de données géographiques. Le type *Complex Data* permettant de décrire les données d'entrée est spécifié par trois éléments : le format des données, leur type d'encodage et leur schéma. Ce dernier élément référence un schéma logique GML.

Le modèle OWL-S[6] est dédié à la description de tous les services du Web sémantique. Une description comprend une présentation générale ('service profile'), une définition sémantique des entrées, sorties, pré-conditions et effets du service en OWL ('service model'), et des instructions pratiques sur l'invocation du service reposant sur WSDL ('service grounding'). L'élément 'service model' est en cours d'enrichissement dans des buts de découverte et de chaînage de services géographiques, [Lemmens 2006], notamment par la description des rôles des variables d'un traitement.

D'autres initiatives comme WSMO[7] et WSPEL[8] pourraient être détaillées, mais elles concernent davantage l'exécution des services que leur description.

La figure 4 récapitule les principaux modèles présentés et les types de contraintes qu'ils permettent de décrire (à droite). Les flèches et la partie gauche de la figure représentent les niveaux de structure source auxquels ces contraintes doivent être confrontées pour évaluer l'adéquation du jeu à l'application. Ainsi, de même que pour les structures source, il n'y a pas de modèle global pour la description des structures cible. Un utilisateur doit découvrir par lui-même les contraintes implicites d'une application soit en interrogeant son concepteur, soit en testant l'outil sur ses propres données et en interprétant ses résultats et les messages d'erreur éventuels.

4 Conclusion : de la description à l'adaptation des structures

Les sections 1, 2 et 3 de cet article ont introduit les notions de structure source et structure cible. Plusieurs modèles de description standard ont été évoqués. Il n'existe de modèle global ni pour la description des structures source ni pour celle des structures cible. Or l'adaptation d'une structure source à une application nécessite :

- de comprendre les structures source et cible,
- de confronter les descriptions de ces deux structures,
- d'évaluer le processus de restructuration qui permettra aux données source de respecter les contraintes de la structure cible,
- de mener à bien ce processus de restructuration.

Des précédents travaux [Balley 2007] ont mené à la proposition d'un modèle global de description des structures source. Un processus global de restructuration a également été conçu, spécifié par l'utilisateur au niveau du schéma conceptuel et piloté automatiquement grâce à des mécanismes experts et à la description complète de la structure source.

Pour aller plus loin, nous souhaitons concevoir un modèle de description des structures cible, compatible avec le modèle précédent et suffisamment riche pour transcrire les connaissances des développeurs d'applications et d'outils. Notre processus de restructuration pourra alors être exploité en tant qu'outil générique de pré-traitement par les utilisateurs souhaitant adapter leurs données à un traitement documenté.

6 http://www.w3.org/Submission/OWL-S/

7 http://www.wsmo.org/

8 http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

Bibliographie

Abadie N., Gesbert N., Mustière S., 2006, « Création d'une ontologie à partir des spécifications textuelles pour l'intégration des bases de données géographiques (papier court)", dans *Actes de la conférences Ingénierie des Connaissances (IC'2006)*, Nantes.

Abd-el-Kader Y., Bucher B., 2006, "Cataloguing GI Functions provided by Non Web Services Software Resources Within IGN", dans *Actes de la 9e conférence AGILE*, Visegrad, Hongrie.

Akkiraju R., Farrell J., Miller J., Nagarajan M., Schmidt M-Th., Sheth A., Verma K., 2005, Web Service Semantics - WSDL-S, W3C Member Submission.

Balley S., 2005, "Improving geographical datasets usability by interactive schema transformations", dans *Actes de la 8e conférence AGILE,* Estoril, Portugal.

Balley S., 2007, *Aide à la restructuration de données géographiques sur le web. Vers la diffusion à la carte d'information géographique.* Thèse de doctorat en informatique soutenue le 19 septembre 2007, Université Paris Est.

Bédard Y., Larrivée S., Proulx M.J., Nadeau M., 2004, "Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 years of Research and Experimentations on Perceptory", S. Wang et al., ed., *ER Workshops 2004,* LNCS 3289.

Bucher B., Balley S., 2007, "A generic preprocessing service for more usable data processing services", dans *Actes de la 10e conférence AGILE,* Aalborg, Danemark.

Christensen J.V., 2006, "Formalizing Specifications for Geographic Information", dans *Actes de la 9e conference AGILE*, Visegrad, Hongrie.

Gesbert N., 2005, *Étude de la formalisation des spécifications de bases de données géographiques en vue de leur inté - gration*. Thèse de doctorat en Informatique soutenue le 2 décembre 2005, Université de Marne-la-Vallée.

Grum E., Vasseur B., 2004, "How to Select the Best Dataset for a Task ?", dans *Actes de la conférence ISSDQ'2004 (International Symposium on Spatial Data Quality)*, Bruck an der Leihta, Autriche.

Hunter G.J., Wachowicz M., Bregt,A.K., 2003, "Understanding Spatial Data Usability", *Data Science Journal,* vol. 2.

Lemmens R., 2006, "Semantic and syntactic service descriptions at work in geo-service chaining", dans *Actes de la 9e Conférence AGILE*, Visegrad, Hongrie.

Pantzaris D., Donnay J-P., 1996, *La conception de SIG - méthode et formalisme*, Paris Ed. Hermès (Collection Géomatique).

Parent C., Spaccapietra S., Zimányi E., 2006. *Conceptual modeling for traditional and spatio-temporal applications - The MADS approach*, Springer.

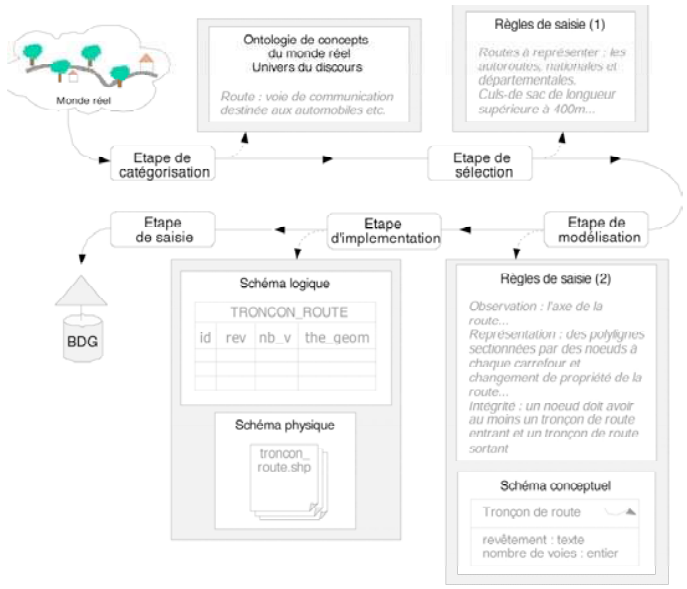

Figure 1 : Etapes du processus menant du monde réel aux données géographiques

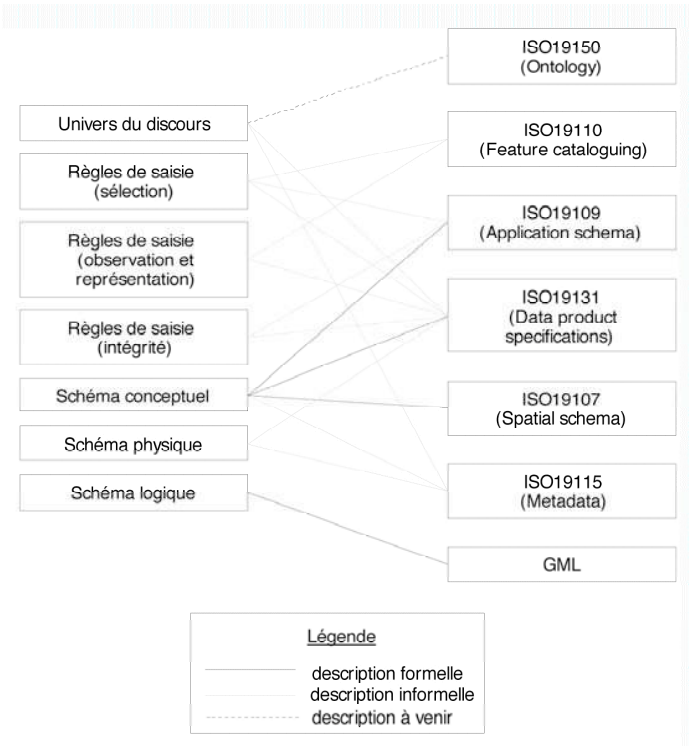

Figure 2 : Description des éléments de structure par des modèles standard

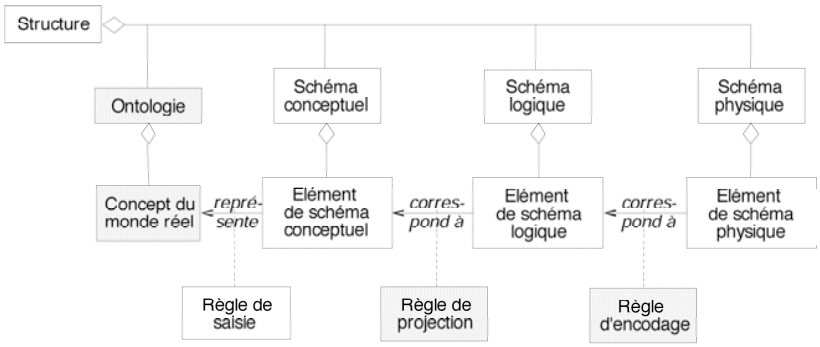

Figure 3 : Modèle intégré pour la description de la structure d"un jeu de données

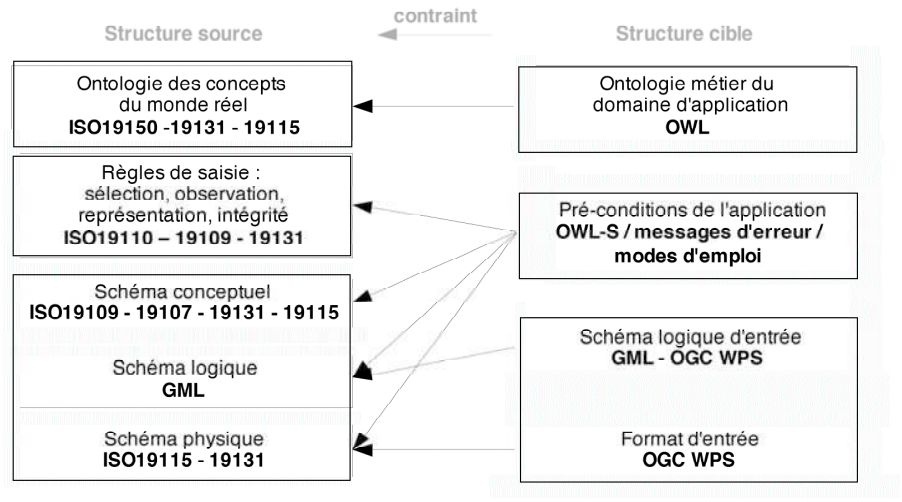

Figure 4 : Relations entre la description de la structure cible d'un traitement (à gauche) et la description de la structure source d'un jeu de données (à droite)