THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Spécialité : *Informatique*

Présentée et soutenue par *Christelle Pierkot*
Le *02 Juillet 2008*

Titre : *Gestion de la Mise à Jour de Données Géographiques Répliquées*

JURY

Rapporteurs
C. Claramunt : *Professeur, Institut de recherche de l'Ecole Navale, Brest*
M. Miquel : *Maître de Conférences, HDR, INSA de Lyon*

Examinateurs
F. Sedes : *Professeur, Université Paul Sabatier, Toulouse*
A. Bouju : *Maître de Conférences, HDR , Université de la Rochelle*
A. Hameurlain : *Professeur, Université Paul Sabatier, Toulouse*
S. Mustière : *Responsable de l'action de Recherche BDMUL, IGN, Saint Mandé*

Invités
S. Guyomard : *Directeur du Centre EADS DS, Labège*
F. Morvan : *Maitre de Conférences, HDR, Université Paul Sabatier, Toulouse*

Ecole doctorale : *MITT*
Unité de recherche : *MIG*
Directeurs de thèse : *A. Hameurlain et A. Ruas*
Encadrant : *Sébastien Mustière*
Thèse Cifre financée par la société EADS DS

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Une donnée géographique représente une entité plus ou moins complexe du monde réel comme par exemple une route, une zone industrielle ou encore un aéroport. La science de l'information géographique rassemble les méthodologies et les outils permettant à divers corps de métiers d'accéder et d'utiliser les données spatiales. De nos jours, l'information géographique constitue une ressource incontournable dans un contexte de prise de décision et les données numériques spatiales sont de plus en plus fréquemment exploitées comme support et aide à la décision par de nombreuses organisations. En effet, de nombreux organismes, quel que soit leur domaine d'activité, ont recours aux données spatiales pour planifier, analyser ou encore exécuter leurs projets.

1.1 Contexte

L'institution militaire utilise les données spatiales comme soutien et aide à la décision. A chaque étape d'une mission, des informations géographiques de tous types sont employées (données numériques, cartes papiers, photographies aériennes...) pour aider les unités dans leurs choix stratégiques. Par exemple, pour définir les zones à sécuriser, prévoir les plans de vol, ou encore organiser les déplacements des troupes.

Dans ce contexte, la DGA (Délégation Générale de l'armement) a demandé une étude permettant une meilleure exploitation des données de description de l'environnement, en tenant compte des contraintes propres à chaque système engagé dans une opération. Le projet ENVOL [1], cadre de ce travail, est divisé en deux volets. Le Volet Infrastructure (Envol VI) s'attache à mettre en place des solutions techniques pour garantir la sélection et l'accès aux données spatiales. Le Volet Dynamique et Cohérence (Envol VDC) doit fournir des solutions pour permettre la prise en compte et la diffusion d'évolutions, tout en maintenant la cohérence globale entre les différents systèmes opérationnels participant à une opération. Le travail de cette thèse s'inscrit dans le cadre du Volet Dynamique et Cohérence du projet Envol [Pierkot et Raynal, 2004a], [Pierkot et Raynal, 2004b].

1. Pour ENVironnement On Line

Un scénario de démonstration permettant de valider les choix techniques a été spécifié dans le projet Envol VDC [Leblanc et Villot, 2003]. Il permet de décrire les relations entre les différentes unités participant à une mission militaire. Il détaille en particulier, les différentes étapes de la mission ainsi que les rôles de chacun des acteurs pour la gestion de l'information géographique.

1.1.1 Rôles et acteurs d'une mission militaire

Dans ce scénario, une mission militaire se déroule sur deux sites distincts, le quartier général et le terrain d'action. Les acteurs participant à une mission sont des unités avec leurs personnels et leurs équipements. Ils possèdent des rôles différents qui sont déterminés en fonction des besoins de la mission. On distingue, les fournisseurs, les producteurs, les opérationnels et les utilisateurs d'information géographique (Cf. figure 1.1) .

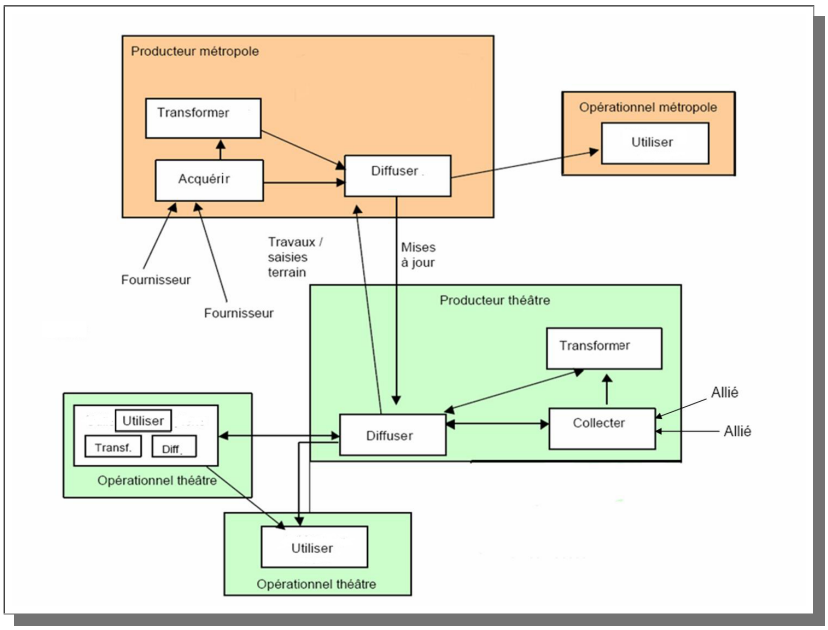

FIGURE 1.1 – Acteurs et rôles d'une mission opérationnelle

- Les fournisseurs sont en général extérieurs à l'opération militaire. Leur rôle est de collecter des données spatiales et de les diffuser aux producteurs de la mission. Les fournisseurs peuvent être des organismes publics comme par exemple l'IGN (Institut Géographique National) ou encore des alliés déjà présents sur la zone d'intervention et ayant préalablement relevé des informations.

- Les producteurs ont pour rôles l'acquisition et la diffusion des données et des évolutions. Lors d'une mission militaire, deux producteurs sont mis à contribution : le producteur métropole qui est situé au quartier général et le producteur théâtre situé sur le terrain d'action. Leurs rôles sont légèrement différents :

  – Le producteur métropole a pour mission de mettre à disposition de tous les autres acteurs des produits géographiques de contenu, résolution et précision

différents, permettant de répondre aux différents besoins opérationnels connus. Son objectif premier est de constituer les jeux de données de la zone d'intervention et de les rendre exploitables à l'ensemble des utilisateurs de données spatiales quel que soit leur corps d'armes (armée de l'air, armée de terre...). Ces jeux de données qui serviront de référence pendant toute la durée de la mission, doivent être dans un format lisible par tous les systèmes utilisateurs, couvrir au mieux la zone d'intérêt, être les plus récents possibles et enfin être cohérents.
Le producteur métropole doit également mettre à jour les jeux de données de référence qu'il a constitué, à partir d'évolutions dont il est informé. Ces évolutions sont des mises à jour ou des nouvelles données et sont collectées par les producteurs de la mission ou récoltées auprès des fournisseurs extérieurs.

- Le producteur théâtre, reçoit les données du producteur métropole et les utilise comme base pour développer son propre système. Sa mission est de compléter l'information de référence mise à disposition par le producteur métropole pour fournir aux opérationnels des produits répondant directement à leurs besoins particuliers. Son but premier est d'évaluer la qualité des jeux de données fournis par le producteur métropole, à partir de relevés terrain ou de mesures d'échantillons sur la zone. Il identifie de cette façon les erreurs figurant dans les jeux de données et en informe le producteur métropole afin que ce dernier puisse répercuter les corrections sur l'ensemble des données de référence. Son deuxième objectif consiste à mettre à jour les jeux de données en effectuant des observations locales (exploitation de données satellitaires, levés terrain ...), en intégrant les mises à jour effectuées par les opérationnels et en récoltant des évolutions auprès des alliés déjà présents sur la zone d'intervention.

- Les opérationnels sont situés au quartier général et sur le terrain d'action. Ils sont constitués de deux types d'acteurs : les opérationnels complexes et les utilisateurs.

  - Les opérationnels complexes sont situés exclusivement sur le terrain d'action. Ils fournissent au producteur théâtre les informations recueillies sur le terrain afin que celui ci enrichisse sa base de données. Ils peuvent transformer l'information de référence et ajouter de nouvelles informations selon le besoin de leur mission.

  - Les utilisateurs sont situés au quartier général et sur le terrain d'action. Ils obtiennent auprès de leur producteur l'information de description de l'environnement dont ils ont besoin pour remplir leur mission. L'information de référence leur sert de fond d'écran sur lequel ils mettent en place des couches d'informations qui leur sont propres, en fonction de leurs objectifs personnels (planification des plans de vols, exécution d'une opération de reconnaissance sur le terrain...). Les utilisateurs ne peuvent néanmoins pas mettre à jour les jeux de données qui leur sont fournis.

La qualité et la quantité des informations fournies aux différents acteurs contribuent pour une grande part au succès de la mission. Lors d'une opération militaire, les données spatiales sont employées en support, par exemple pour gérer le mouvement des troupes sur le terrain ou encore planifier les vols aériens. Les données sont donc dans un premier temps collectées en fonction de l'objectif global de la mission, puis distribuées en tenant compte du besoin particulier de chaque unité. Dans un deuxième temps, des campagnes de mise à jour des données de référence et de collecte de données manquantes sont organisées afin que les jeux de données reflètent au mieux la réalité du terrain.

1.1.2 Déroulement du scénario de démonstration

Le scénario de démonstration utilisé dans le projet Envol VDC, fournit des précisions sur la distribution et le transfert des données entre les acteurs d'une mission militaire [Leblanc et Villot, 2003]. Cinq étapes ont été définies, chacune précisant les modifications effectuées sur les jeux de données des utilisateurs (acquisition de nouvelles données, mise à jour des données existantes) ainsi que les échanges possibles entre les différents acteurs.

Durant la première phase ("Planification", figure 1.2), un plan d'acquisition des données répondant le mieux aux besoins est mis en oeuvre. Ce plan consiste à identifier les produits de référence, à les acquérir auprès de fournisseurs extérieurs et à les importer dans la base de référence du producteur métropole.

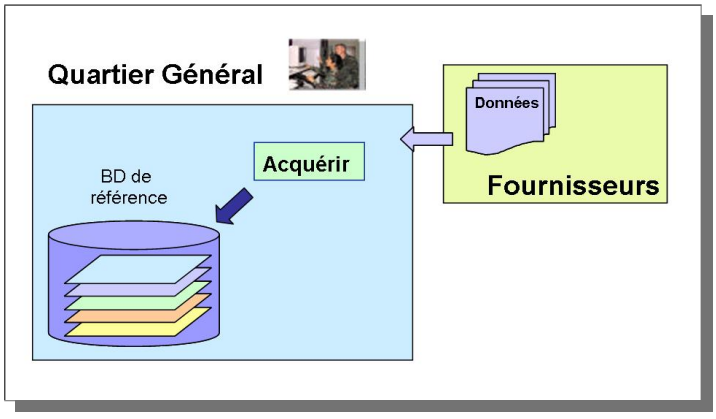

FIGURE 1.2 – Phase de planification lors d'une mission opérationnelle

Dans la seconde phase ("Montée en puissance avant l'engagement", figure 1.3), tous les acteurs susceptibles d'être concernés par l'opération obtiennent les données et les intègrent dans leur système. Un plan de production réactif est également mis en oeuvre au niveau du quartier général, afin de combler au plus vite et en fonction des priorités opérationnelles, les données manquantes identifiées.

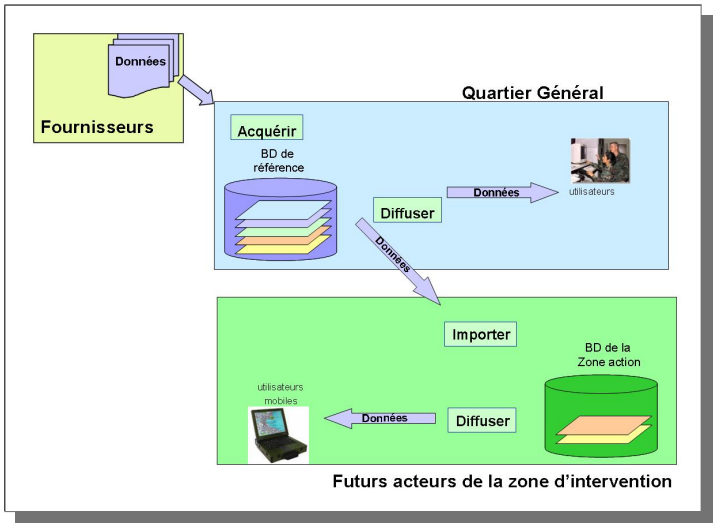

FIGURE 1.3 – Phase de montée en puissance avant l'engagement

Pendant la troisième phase ("Déploiement et équipement de la zone d'opération", figure 1.4), une partie des troupes est envoyée sur la zone d'opération. Ces unités obtiennent par le biais d'armées alliées, des données plus récentes qu'ils intègrent dans leur système. Les données récoltées sont ensuite diffusées au quartier général qui doit les intégrer sans remettre en cause la cohérence de sa base de données. Parallèlement, le plan de production réactif continue d'être mis en oeuvre et les données ainsi produites sont diffusées aux autres acteurs.

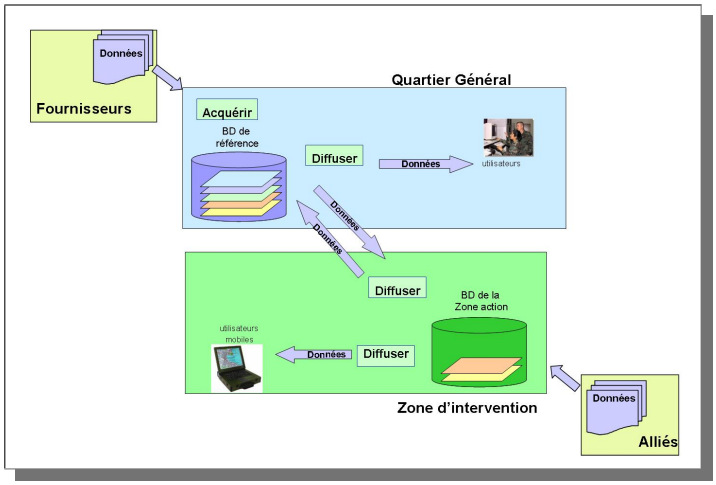

FIGURE 1.4 – Phase de déploiement des unités sur la zone d'intervention

La quatrième phase ("Déploiement de longue durée", figure 1.5) est une phase de stabilisation où les travaux d'enrichissement et de mise à jour sur le terrain continuent d'être effectués selon un plan de priorité établi en fonction du besoin opérationnel. Les informations recueillies sont mises à disposition des autres acteurs

sur le terrain et sont envoyées au quartier général afin d'être intégrées dans la base de référence. Les données sont ensuite envoyées aux unités de commandement (utilisateurs) qui les intègrent sans remettre en cause la cohérence de leur propre jeu de données. Il est primordial que ces derniers aient accès à des données cohérentes avec celles qui sont disponibles sur le terrain d'action.
En parallèle, le quartier général peut décider de répercuter les évolutions qui arrivent de la zone d'intervention, dont la résolution et la précision correspondent en général aux grandes et moyennes échelles, sur les produits à plus petite échelle utilisés pour la conduite et la planification.

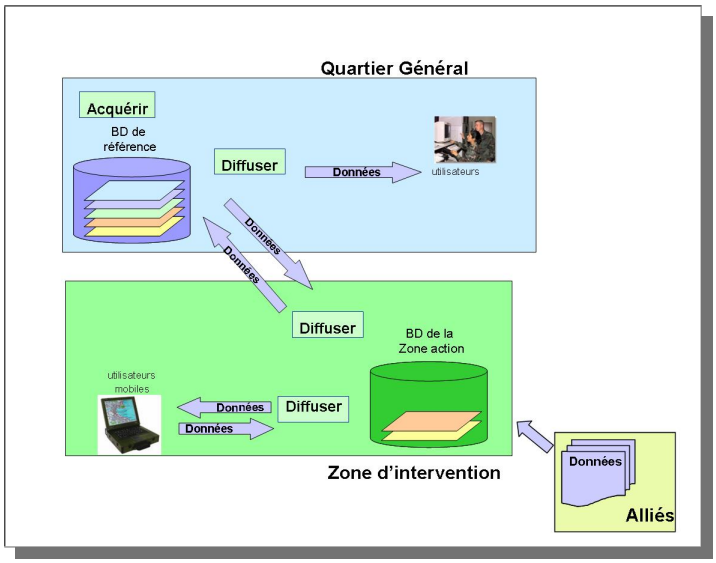

Figure 1.5 – Phase de déploiement longue durée des unités sur la zone d'intervention

Lors de la cinquième et dernière phase ("Désengagement", Figure 1.6), l'unité située sur le terrain, transmet à son retour, la totalité des informations acquises afin de mettre à jour la base de référence de la défense sur la zone du globe concernée.

1.2 Analyse du scénario de démonstration

Ce scénario de démonstration témoigne de la complexité des échanges entre les acteurs participant à une mission opérationnelle et expose la difficulté à gérer la mise à jour des jeux de données spatiaux dans un univers réparti.

1.2.1 Problèmes engendrés par l'échange des mises à jour multi-sources

En effet, en analysant le scénario de démonstration, nous voyons qu'en fonction de leurs besoins, les acteurs d'une mission militaire, soit effectuent directement la mise à jour de leurs jeux de données, soit récupèrent des évolutions auprès des autres acteurs de la mission. Cela a pour effet de **multiplier les échanges de données**

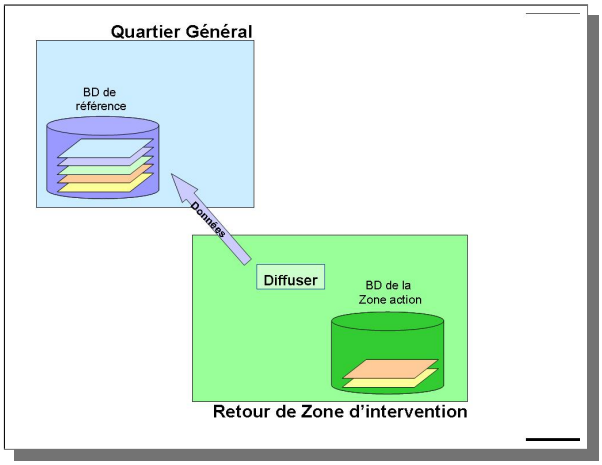

Figure 1.6 – Phase de désengagement des unités de la zone d'intervention

hétérogènes ayant des caractéristiques différentes telles que la qualité, l'actualité, ou encore le niveau d'abstraction. Prenons l'exemple des mises à jour fournies par les alliés déjà en place sur la zone d'intervention. Ces données n'ont pas forcément le même format, ne proviennent peut être pas du même contexte (paix, crise), ne sont peut être pas référencées de la même façon, n'ont peut être pas la même précision, ni la même résolution que les données présentes dans la base. Leur intégration dans la base de données de l'utilisateur provoque des incohérences si aucun traitement n'est appliqué au préalable.
Un autre exemple caractéristique de l'hétérogénéité causée par la mise à jour est celui de la saisie sur le terrain par les opérationnels. Premièrement, leurs systèmes sont moins sophistiqués que ceux du producteur et ne prennent pas forcément en compte la topologie. Ensuite, l'opérationnel sur le terrain peut avoir un besoin rapide d'informations qu'il ne possède pas dans son jeu de données (par exemple, sur une zone où très peu de données ont été produites). L'urgence de la situation impose une saisie rapide, non précise et éventuellement erronée des nouvelles données. Des corrections doivent être appliquées avant toute intégration dans les systèmes producteurs.
Enfin, les mises à jour peuvent être **concurrentes** car saisies plusieurs fois par des utilisateurs distincts.

La diffusion et l'intégration sans ménagement de ces mises à jour peuvent donc provoquer des incohérences et remettre en cause l'intégrité des bases de données des utilisateurs.

1.2.2 Infrastructure militaire et réseau de communication

Afin de limiter les risques évoqués ci-dessus, nous avons proposé de mettre en place une politique d'échange et de diffusion des mises à jour permettant d'assurer une collaboration efficace entre les différents acteurs. Pour cela, nous utilisons une infrastructure militaire dans laquelle des informations telles que le nombre de sites de déploiement, les moyens matériels utilisés, les rôle des acteurs, le type de

coopération ou encore la nature des informations échangées sont spécifiés. Dans cette infrastructure, un réseau de communication est établi entre les utilisateurs de données, ce qui facilite la coopération et permet l'échange d'informations de manière sécurisée. La figure 1.7 s'appuie sur le scénario de démonstration pour présenter un exemple de réseau de communication tel que nous l'avons suggéré et qui peut être mis en place entre deux sites lors d'une mission de défense.

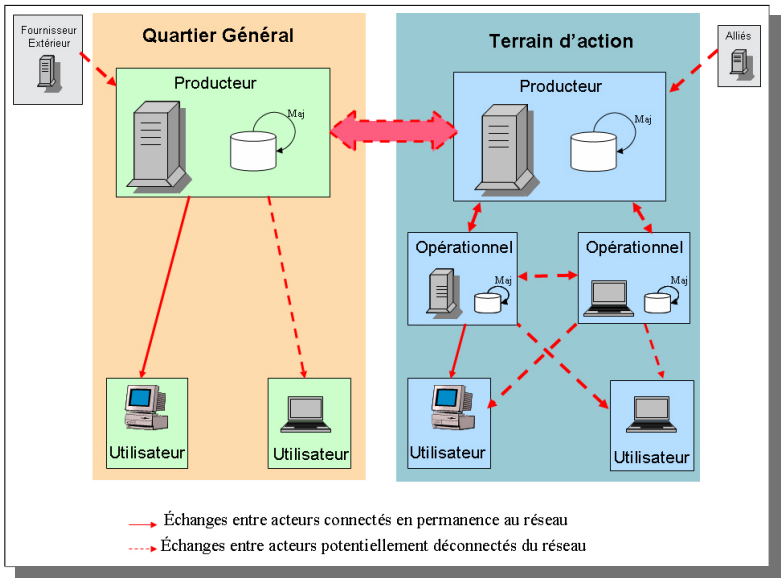

Figure 1.7 – Communication entre les acteurs d'une mission opérationnelle

Nous avons ensuite effectué une analyse détaillée du problème de la collaboration entre plusieurs acteurs échangeant de l'information dans un environnement distribué, analyse que nous présentons ci dessous. Pour chaque point de cette étude, nous exposons les moyens couramment utilisés pour traiter les problèmes liés à l'échange de données réparties, puis nous positionnons ces méthodes par rapport à notre contexte particulier et enfin nous donnons des pistes pour solutionner certains de ces problèmes en considérant les obligations et contraintes militaires.

1.2.3 Réplication des données

L'accès aux données distribuées sur un réseau de communication peut se faire soit par un serveur centralisant l'information sur lequel les différentes applications se connectent, soit par réplication des données sur chaque site. La centralisation suppose d'avoir un serveur accessible par toutes les applications, ce qui implique une connexion permanente à cette machine. La sécurité et la confidentialité des informations constituant une des préoccupations majeures des militaires, il n'est pas envisageable qu'une connexion permanente existe entre deux sites distants. **Les données sont donc répliquées sur chacun des sites**.

Les échanges entre les répliques peuvent être mono (un seul acteur peut fournir ses informations aux autres) ou multidirectionnels (tous les acteurs peuvent

s'échanger librement leurs informations). La réplication est dite asymétrique (ou encore maître/esclaves) lorsque seule la copie d'un site maître peut être mise à jour, les autres répliques doivent alors se connecter à ce site de référence pour rafraîchir leurs données. Elle est en revanche symétrique (ou multimaîtres) lorsque tous les sites sont autorisés à mettre à jour leur copie locale. **L'architecture de réplication** que nous devons mettre en place dans le cadre d'une infrastructure militaire doit être **multimaîtres** car chaque site peut librement mettre à jour ses données en fonction de nouvelles disponibilités ou de relevés sur le terrain et les propager ensuite aux autres répliques.

La réplication multimaîtres permet l'écriture des données par des utilisateurs distincts, ce qui favorise les mises à jour conflictuelles. La cohérence des répliques et la politique de réplication dépendent des besoins de l'application. Si le but est d'avoir à tout instant la même valeur sur toutes les répliques alors il faut utiliser des verrous afin d'avoir une cohérence forte (lecture et écriture non permises pour les autres répliques) [Bernstein *et al.*, 1987]. Si, en revanche, les données peuvent diverger ponctuellement, on choisira une cohérence faible [Saito et Shapiro, 2005].

Les applications permettant la collaboration entre plusieurs utilisateurs sont également caractérisées par le mode de connexion entre les utilisateurs et l'infrastructure de communication. La collaboration synchrone est utilisée lorsqu'un besoin de connaissance rapide des mises à jour est souhaité. La collaboration asynchrone est plutôt employée lorsque les utilisateurs sont distants et peuvent être déconnectés du réseau de communication. Dans le contexte d'une mission opérationnelle, les utilisateurs de données spatiales sont répartis sur des sites distants et peuvent être potentiellement déconnectés du réseau. Cela arrive par exemple, lorsqu'un opérationnel utilise une unité mobile pour faire des relevés sur la zone d'intervention. Nous pensons que **le mode de connexion d'un réseau militaire doit être asynchrone**.

Les protocoles de réplication à cohérence faible sont des protocoles asynchrone qui peuvent être mis en place avec une politique de réplication pessimiste ou optimiste.
Les protocoles de réplication pessimistes résolvent les problèmes de concurrence a priori à l'aide de contraintes et verrous appliqués sur les communications entre répliques (protocole de copie primaire [Kronenberg *et al.*, 1986], [Oracle, 1996], [Dietterich, 1994] ou à base de quorum [Helal *et al.*, 1996]). Ils donnent alors l'illusion à l'utilisateur qu'il n'existe qu'une seule copie [Herlihy et Wing, 1990], [Bernstein et Andgoodman, 1983], [Bernstein *et al.*, 1987] mais ils sont difficilement utilisables en mode déconnecté. En revanche, les protocoles optimistes acceptent toutes les écritures a priori, détectent les conflits lors de la synchronisation et utilisent a posteriori des procédures de réconciliation des écritures concurrentes qui préservent les critères de cohérence [Demers *et al.*, 1994], [Kermarrec *et al.*, 2001], [Saito et Shapiro, 2005], [Oster, 2005]. Lors d'une mission militaire, les unités situées au quartier général ou sur le terrain d'action effectuent des mises à jour de leurs données en parallèle et peuvent travailler en mode déconnecté. La disponibilité des données est indispensable, l'utilisation de verrous n'est pas envisageable et imposer des règles de communications entre les différents sites peut s'avérer très difficile à

mettre en œuvre étant donné la mobilité des unités. Dans un tel contexte, nous devons mettre en place **un protocole de réplication optimiste** et nous devons établir **une politique de réconciliation préservant les critères de cohérence**.

1.2.4 Gestion de la cohérence des données répliquées

La gestion de la cohérence est étroitement liée au type de données qui sont gérées et nécessite l'utilisation d'algorithmes et de protocoles adaptés [Dedieu, 2002]. La nature des données échangées (structurées ou non, volume d'information échangé, relations structurelles ou sémantiques entre les données, données modifiables ...) définit donc également le type d'application à mettre en place ainsi que les critères de cohérence requis. Dans notre contexte, les données spatiales sont utilisées comme support et aide à la décision lors de missions militaires. Nous devons donc considérer des critères de cohérence spécifiques aux propriétés spatiales des données (cohérence structurelle, géométrique, topologique ...) et au contexte applicatif lors de l'élaboration des protocoles et algorithmes de réconciliation.

1.3 Position du problème

Ce travail s'intéresse donc à la mise à jour de données géographiques par différents acteurs répartis sur des sites distincts lors de missions militaires. Les données géographiques sont répliquées sur les différents sites, il n'existe pas de serveur centralisant l'information et la mise à jour est effectuée en parallèle (simultanément) par les acteurs. **Les évolutions ne sont donc pas nécessairement pertinentes pour un utilisateur particulier, sont parfois entachées d'erreurs et des problèmes de cohérence peuvent apparaître lors de leur intégration dans les différents jeux de données.**

La problématique principale concerne donc la gestion dans un contexte militaire, d'une application collaborative permettant la mise à jour asynchrone et symétrique de données géographiques répliquées selon un protocole à cohérence faible optimiste. Cela nécessite de définir un modèle de cohérence approprié au contexte militaire, un mécanisme de détection des mises à jour conflictuelles lié au type de données manipulées et des procédures de réconciliation des écritures divergentes adaptées aux besoins des unités participant à la mission.

La contribution de cette thèse est double. Premièrement, elle s'inscrit dans le domaine de la gestion de la mise à jour des données spatiales, domaine toujours très actif du fait de la complexité et de l'hétérogénéité des données et de la relative "jeunesse" des travaux sur le sujet [Badard, 2000], [J.Pouliot *et al.*, 2001], [Peerbocus *et al.*, 2002], [Kadri-Dahmani, 2005]. Deuxièmement, elle s'inscrit dans le domaine de la gestion de la cohérence des données répliquées selon un protocole optimiste, en spécifiant en particulier, de nouveaux algorithmes pour la détection et la réconciliation de données conflictuelles, dans le domaine applicatif de l'informa-

tion géographique.

Notre objectif final est de proposer des solutions permettant l'intégration cohérente et autant que possible automatique, des mises à jour de données spatiales dans un environnement de réplication optimiste, multimaîtres et asynchrone. Nous proposons une stratégie globale d'intégration des mises à jour spatiales basée sur un contrôle de concurrence couplé à des sessions de mises à jour. L'originalité de cette stratégie réside dans le fait qu'elle s'appuie sur des métadonnées pour fournir des solutions de réconciliation adaptées au contexte particulier d'une mission militaire.

1.4 Plan

Ce mémoire est organisé de la façon suivante.

Le chapitre 2 présente un état de l'art des travaux scientifiques menés en matière de gestion des données réparties et de gestion de la mise à jour des données spatiales. Nous nous intéressons premièrement au mécanisme de la mise à jour en information géographique puis nous étudions les travaux effectués sur la réplication optimiste des données. Nous faisons ensuite une analyse croisée de ces travaux qui permet de préciser notre sujet et d'exposer plus précisément la contribution de ce travail dans chacun des domaines de recherche (réplication optimiste des données spatiales et maintien de la cohérence dans un contexte de mise à jour).

Le chapitre 3 est décomposé en deux parties et constitue notre proposition à ce travail de recherche. La première partie est consacrée à la définition de l'infrastructure militaire que nous avons mise en place, à la politique de gestion des évolutions au sein de cette infrastructure et au modèle de métadonnées que nous allons utiliser dans la stratégie d'intégration des évolutions. En particulier, nous détaillons le profil communautaire de la norme ISO 19115 que nous avons mis en place pour la gestion des évolutions.
La seconde partie de ce chapitre est consacrée à la stratégie d'intégration des mises à jour que nous avons définie afin de traiter les problèmes de cohérence. Nous décrivons dans un premier temps la stratégie globale, puis nous proposons une piste pour effectuer la vérification de la pertinence, ensuite nous détaillons le protocole de vérification de la cohérence et enfin nous discutons de la nécessité d'effectuer des sessions de mises à jour.

Le chapitre 4 détaille la mise en oeuvre de notre approche. Nous exposons ici la nature des données que nous avons utilisées et expliquons les simulations que nous avons effectuées pour tester la faisabilité de notre proposition. Nous concluons ce chapitre avec une évaluation des performances de notre stratégie, en particulier nous analysons les résultats de la vérification de la cohérence.

Le chapitre 5 constitue la conclusion et fournit quelques réflexions sur les éventuelles suites à donner à ce travail.

Chapitre 2

État de l'art : Information géographique et gestion des données réparties

2.1 Cohérence et mise à jour de données spatiales

L'objet principal de la science de l'information géographique est la gestion des données spatiales. Le recours à plusieurs domaines tels que la cartographie, la télédétection, la géodésie, la photogrammétrie, la topométrie ou encore l'informatique est nécessaire pour traiter efficacement ce type de données. Dans cette discipline, les méthodes relatives à l'acquisition, au stockage, au traitement, à la diffusion et à la gestion des données géographiques doivent être utilisées [OQLF, 2004].

Dans cette section, nous présentons les méthodes et techniques utilisées en information géographique pour la gestion des données spatiales. Nous discutons dans un premier temps de la particularité des données spatiales [Scholl *et al.*, 1996]. Nous définissons brièvement les données géographiques, voyons quels sont les différents modes d'acquisition et les différentes sources qui permettent la construction de jeux de données géographiques, listons les différents types de données spatiales utilisés en information géographique et abordons les notions relatives à la topologie.

La deuxième partie parle de la difficulté de la mise à jour des jeux de données géographiques et dresse un inventaire des différents travaux de recherche qui ont été effectués dans ce domaine [Devogele, 1997], [Badard, 2000],[Pouliot *et al.*, 2001]. Nous détaillons ici les différentes étapes de la mise à jour d'un jeu de données géographiques (extraction, classification, diffusion, intégration et propagation).

La troisième partie est consacrée aux infrastructures de données spatiales (ou SDI pour Spatial Data Infrastructure) [Rajabifard et Williamson, 2001], [Nebert, 2004], discipline à part entière de l'information géographique qui s'attache à trouver des solutions pour assurer une meilleure collaboration entre les nombreux utilisateurs des données spatiales. Nous définissons ici les termes consacrés aux SDI, précisons en quoi l'utilisation des métadonnées favorise l'interopérabilité et exposons les différents travaux engagés dans ce domaine

[Benslimane *et al.*, 1999],[Leclercq, 2000],[Brodeur, 2004].

Enfin, nous discutons de la notion de cohérence en information géographique [Sheeren, 2005], [Kadri-Dahmani, 2005]. Nous parlons premièrement de la qualité des données spatiales [Ubeda, 1997], [Puricelli, 2000], [Vasseur, 2004], [Devillers et Jeansoulin, 2005]. Puis nous exposons les sources d'erreurs qui peuvent conduire à un état instable du système et détaillons les principales taxonomies de conflits existantes dans la littérature [Devogele, 1997], [Benslimane *et al.*, 1999], [Defude, 2005]. Enfin, nous définissons plus précisément la cohérence des jeux de données géographiques [Ubeda, 1997], [Puricelli, 2000], [Braun, 2003].

2.1.1 Caractéristiques des données spatiales

Selon l'office québécois de la langue française [OQLF, 2004], les données spatiales comprennent l'ensemble des données géométriques, des données descriptives et des métadonnées portant sur les entités spatiales et leurs relations, dans une application géomatique.
La donnée géométrique renseigne sur la position ou la forme d'une entité géographique selon une référence spatiale, c'est-à-dire une référence fixée dans l'espace par rapport à un système de coordonnées.
Une donnée descriptive est une donnée relative à un des attributs d'une entité ou d'une relation, à l'exclusion de sa position et de sa forme.
Une base de données (BD) est un ensemble cohérent de données structurées qui constitue un modèle de la réalité. **Une base de données géographique** (BDG) est un ensemble cohérent de données géométriques et descriptives représentant une région spatiale.

Cependant, en information géographique, les utilisateurs n'accèdent pas aux données directement par le biais de la base de données mais au travers d'un **système d'information géographique** (SIG). Les systèmes d'information géographiques permettent l'intégration, la gestion, l'interrogation, l'analyse et la restitution des données géographiques [Defude, 2005]. Ils s'appuient d'une part sur des bases de données géographiques pour stocker l'information [Guting, 1994] et d'autre part sur un ensemble d'opérations permettant de manipuler cette information. Ils permettent par conséquent à l'utilisateur d'exploiter de façon transparente les données spatiales.

Les données géographiques sont acquises par le biais de différentes sources comme les images (aériennes, satellites, orthophotographies....), les cartes (papiers, scannées et numériques), les bases de données (routières, topographiques, cartographiques, ...), la télédétection ou encore la géolocalisation (GPS ou relevé sur le terrain). Parmi les techniques les plus répandues pour acquérir de l'information géographique, on trouve les levés topographiques, les relevés GPS, la scanérisation de carte papier, la vectorisation d'images ou encore l'importation de données auprès de producteurs extérieurs. Les données produites par chacune de ces applications ont des propriétés et des qualités différentes. On distingue quatre formes numériques de données : Les données vectorielles, les données raster, les grilles et les données sans information de localisation.

Les données vectorielles sont des éléments qui représentent les objets géographiques par leur géométrie (point, ligne, polygone). Elles sont en général obtenues par les trois moyens suivants : la tablette à digitaliser lorsqu'on dispose d'une carte papier, la saisie sur l'écran lorsqu'on dispose d'une image numérique et enfin automatiquement grâce à un programme de vectorisation. Les erreurs classiques que l'on retrouve suite à une acquisition des données vectorielles sont des décalages ou des problèmes de précision des données.
Les données maillées ou raster sont des images formées par une succession de points portant une information radiométrique ou colorimétrique (le pixel). Chaque point possède des coordonnées intrinsèques par rapport à un des coins de l'image. Les photos numériques, images satellites, orthophotographies et cartes scannées font parties de cet ensemble de données. Ils sont en général utilisés comme image de fond et fournissent ainsi de l'information complémentaire.
Les grilles sont des données utilisées pour décrire un phénomène continu sur une zone donnée. Il existe deux types de grilles : les MNT (Modèles Numériques de Terrain) qui sont des grilles régulières, généralement utilisées pour estimer l'altitude et les TIN (Triangulated Irregular Network) qui sont des grilles irrégulières, plus précises mais plus difficilement utilisables.
Enfin, les données sans information de localisation fournissent les caractéristiques non spatiales des objets géographiques auxquels elles sont attachées.

L'organisation des données vectorielles dépend de la structure topologique utilisée. **La topologie** est une discipline des mathématiques qui étudie les propriétés liées au concept de voisinage. En information géographique c'est la discipline qui s'intéresse aux propriétés géométriques de l'espace tels que la connectivité ou le voisinage. Cela regroupe un ensemble de modèles de données et de traitements qui permettent de mémoriser et d'exploiter les relations de connectivités et de voisinage qui existent entre les données. Deux grandes familles de modèles topologiques (au sens large) ont été définies : les modèles spaghetti et les modèles topologiques (au sens strict).
Les modèles spaghetti permettent uniquement de décrire les primitives géométriques (points, lignes, polygones). Autrement dit, aucune relation existante entre les objets n'est représentée. C'est le modèle le plus simple qui offre l'avantage de permettre un coût faible lors de la saisie des données car aucune transformation n'est appliquée, mais qui en contrepartie n'offrent aucun contrôle au niveau des données, ce qui conduit à de nombreuses incohérences.
Les modèles topologiques (au sens strict), quant à eux, conservent les relations de connexion et d'adjacence entre les objets. Ils sont basés sur la théorie des graphes pour la gestion des connexions. Les objets géographiques sont ici représentés par des points, des noeuds, des arcs et des faces. Ils sont plus riches que les modèles spaghetti et offrent la possibilité d'effectuer des requêtes spatiales sur les données mais l'intégration des données provenant de sources diverses n'est pas aisée car les données doivent être transformées dans le format du modèle.
Le choix du modèle topologique a un impact certain sur la gestion des données. En effet, l'utilisation de modèles topologiques distincts implique un découpage de la géométrie des objets qui diffère dans les jeux de données voire une absence des informations de relation si une des bases de données est structurée selon un modèle

spaghetti.

Finalement, **les données géographiques sont hétérogènes** car elles proviennent souvent de sources diverses, possèdent des types variés et n'ont pas forcément la même structuration topologique. Avant toute utilisation, il faut donc les harmoniser afin de les rendre cohérentes entre elles.

Dans cette thèse, nous nous concentrons uniquement sur les données de type vectorielles. Les problèmes de cohérence que l'on rencontre découlent directement des sources et des méthodes utilisées lors de l'acquisition de ces données dans un contexte de mise à jour.

2.1.2 Mise à jour des bases de données spatiales

Selon [Badard, 2000], **les problèmes** entravant la mise à jour des systèmes d'information géographique ont quatre sources principales : l'utilisateur des données, le producteur des données, la structuration des bases de données et les formats d'échange.
L'utilisateur, en fonction de son besoin, peut être amené à faire des modifications ou à enrichir le jeu de données qui lui sert de référence. Par exemple, un militaire qui reçoit un jeu de données non adapté à son besoin réel (une échelle trop grande ou un schéma très précis ou encore contenant des couches d'informations inutiles) peut décider de simplifier le jeu de données de référence pour ne garder que l'information utile (par généralisation, transformation de schéma ou suppression des couches superflues). A contrario, le militaire sur le terrain qui reçoit un jeu de données ne couvrant pas totalement la zone d'action (pas de topologie ou manque de données dans une thématique particulière) peut vouloir ajouter de l'information supplémentaire pour compléter le jeu de données de référence (couches d'informations supplémentaires, topologie, nouvelles données, ...) (Cf. Figure 2.1). Cependant, les liens entre les données de référence et les données dérivées ne sont généralement pas explicites et une mise à jour peut alors entraîner une perte d'information ou amener à un état incohérent de la base.

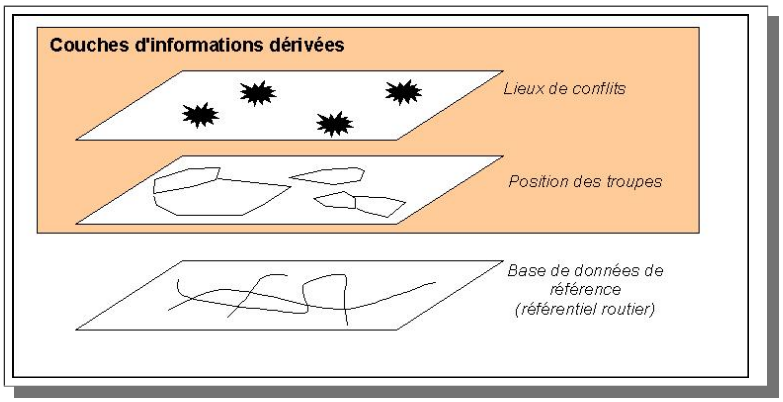

Figure 2.1 – Base de données utilisateur contenant des couches d'informations dérivées

Les problèmes liés au **producteur de données** sont essentiellement dus au changement des spécifications des bases de données (agrégation de classes, modification des paramètres de saisie des objets ...) et à la fréquence des mises à jour qui peut considérablement varier en fonction des différents besoins des utilisateurs. En particulier, l'intégration des évolutions qui est souvent effectuée séparément dans chaque base à partir de sources diverses a pour effet d'augmenter les coûts de maintenance.
Les difficultés dues à **la structuration des bases** viennent du fait que la plupart des bases de données ne possèdent pas de système d'identification des objets qui soit fiable et stable. Cette structuration ne facilite pas la recherche d'objets homologues lors de la mise à jour de la base utilisateur. Il est alors nécessaire d'utiliser un mécanisme d'appariement pour parvenir à retrouver les objets en correspondances [Lemarié, 1996]. Ce processus est encore difficilement automatisable et l'utilisateur doit souvent valider interactivement les liens établis.
Thierry Badard souligne également les problèmes liés à **l'échange de l'information**. Selon lui, les formats d'échange ont été créés pour des données statiques et non pour des informations dynamiques telles que sont les données d'évolution.
Finalement, Thierry Badard constate que des problèmes **d'interopérabilité sémantique** et **d'interopérabilité des systèmes** se posent lorsqu'un producteur veut échanger de l'information d'évolution avec un utilisateur.

Pour l'équipe du projet M@jic [Pouliot *et al.*, 2001], le problème de la mise à jour des données géographiques découle plutôt de la **méthode de transfert** des mises à jour. D'après eux, les producteurs de données procèdent souvent par remplacement global du fichier entier contenant les nouvelles valeurs de données. Cette méthode de transfert ne permet pas d'avoir de liens entre les anciennes et les nouvelles données et donc pas de suivi de l'historique possible. De plus, les données ainsi que les relations géométriques et topologiques avec les couches ajoutées par l'utilisateur sont perdues. Il y a donc perte d'informations entre les données sources et les données dérivées par l'utilisateur à partir des données sources, ce qui occasionne du travail important pour reconstruire les liens.

Thomas Devogele [Devogele, 1997] établit quant à lui, une **taxonomie des conflits** qui sont sources d'erreurs lors du processus d'intégration de plusieurs bases de données géographiques dans une base multi-représentation. Cette liste comprend notamment les conflits liés aux sources (mode de représentation, méthodes de saisie des sources) et les conflits liés à la structuration des bases (définition et spécification des classes, modélisation des données et de la topologie, description sémantique et géométrique). Cette classification peut servir également à décrire les conflits qui interviennent lors de la mise à jour de plusieurs bases de données ayant évoluées différemment.

Plusieurs sources de problèmes **peuvent donc conduire à un état incohérent du système lors de la mise à jour des jeux de données géographiques**, auxquels viennent s'ajouter, dans notre contexte, les problèmes relatifs à la **réplication des données et à la mise à jour simultanée**.

Dans la suite de ce chapitre, nous présentons les différentes phases de la mise à jour des bases de données géographiques et voyons les outils et méthodologies utilisés dans chacune d'entre elles. La première étape de la mise à jour concerne l'extraction des évolutions.

Extraction des mises à jour : Appariement et identification des données

Pour appliquer des mises à jour dans une base de données différente de celle où les évolutions ont été saisies, il faut connaître les changements qui ont eu lieu. Pour cela, on extrait les évolutions entre les deux actualités de la base mise à jour (entre la date correspondant à l'actualité de la base de données avant la mise à jour et la date correspondant à l'actualité de la base de données après la mise à jour). En général, le système qui a généré les mises à jour ne possède pas de mécanismes permettant de créer un journal associé à ces évolutions, **l'extraction des mises à jour** s'effectue alors par comparaison des propriétés des objets que l'on appelle l'**appariement** ou par comparaison **d'identifiants**. L'appariement de données géographiques est rendu obligatoire lorsque la base de données ne possède pas d'identifiants stables sur lesquels s'appuyer pour définir les évolutions.

L'appariement est un processus qui définit les liens entre les objets de différentes bases qui représentent le même phénomène réel. La figure 2.2 montre un exemple d'appariement utilisé pour l'extraction des mises à jour entre deux actualités d'un jeu de données routier.

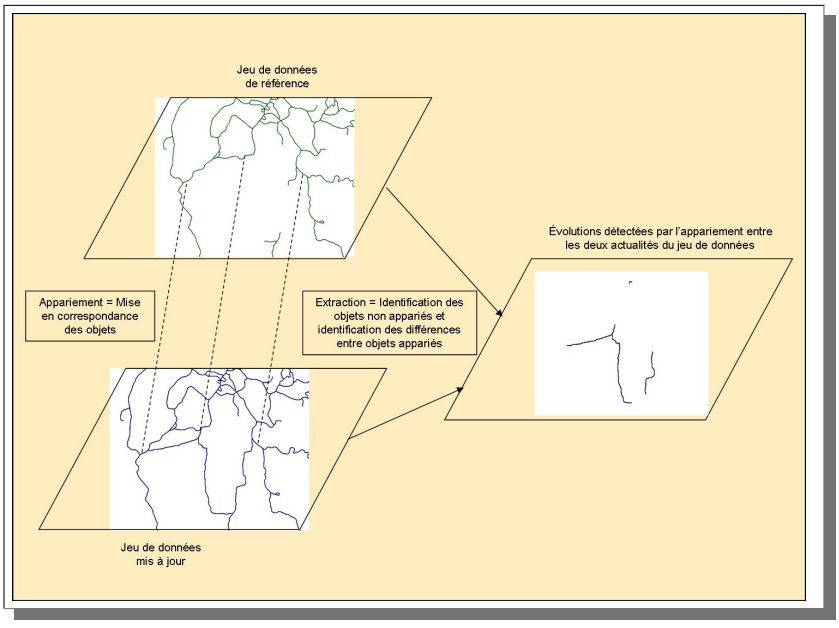

Figure 2.2 – Extraction des évolutions par appariement d'un jeu de données routier

Thomas Devogele [Devogele, 1997] distingue trois outils d'appariement de données géographiques :

– **L'appariement sémantique** qui consiste à mettre en correspondance les objets grâce à la valeur de leurs attributs sémantiques. Cette technique est

dérivée des bases de données classiques.
- **L'appariement géométrique** qui se base sur des notions de distance, de zones et de ressemblances de forme pour comparer la géométrie des objets et retrouver les correspondants.
- **L'appariement topologique** qui utilise les relations de composition et les relations topologiques pour obtenir un ensemble d'objets candidats et limiter les recherches géométriques. C'est un complément des deux autres et ne peut être utilisé seul.

[Devogele, 1997] souligne cependant que les processus d'appariement sont complexes car ils utilisent plusieurs outils pour comparer un grand nombre d'informations. Leur conception est donc difficile mais peut être facilitée par la spécification d'un processus générique et par une interaction avec l'utilisateur dans le choix et le paramétrage des outils.

Selon l'équipe du projet M@jic [Pouliot *et al.*, 2001], **l'appariement géométrique** est largement dépendant de la nature des données géographiques et des seuils utilisés. De plus, il devient extrêmement difficile à utiliser dès lors que les systèmes de coordonnées et les structures diffèrent. L'appariement sémantique exige quant à lui, que les données possèdent le même schéma logique, ce qui est rarement le cas. Finalement, l'appariement combiné est celui qui leur semble le plus intéressant mais il exige l'élaboration d'une stratégie de mise en œuvre des diverses techniques d'appariement. Dans cet article, ils soulignent également que même si l'automatisation des procédures d'appariement peut sembler très alléchante, elle reste limitée. Ils recommandent fortement de toujours procéder à la vérification manuelle de l'appariement.

Selon Atef Bel Hadj Ali [Bel-Hadj-Ali, 2001], l'appariement sémantique ne peut avoir lieu que lorsque les deux bases de données présentent des schémas de données "proches", à défaut, son utilisation s'avère difficile sans une étape préalable de mise en correspondance des schémas des bases de données.

L'appariement géométrique reste celui le plus souvent utilisé pour les données géographiques, mais il est également le plus complexe à mettre en œuvre du fait de plusieurs facteurs :
- Il est difficile de détecter que deux objets occupent la même position dans l'espace sachant que chaque point de l'objet est entaché d'erreurs de localisation plus au moins aléatoires.
- Etant donné qu'un objet d'un jeu de données peut-être apparié avec plusieurs objets de l'autre jeu de données ou même aucun objet, la notion d'objet le plus proche n'est pas suffisante.
- L'appariement doit tenir compte de l'information contextuelle, en analysant la répartition des objets les uns par rapport aux autres dans l'espace géographique.

Bouziani [Bouziani, 2003] pense que le problème le plus difficile à traiter lors de l'appariement est **l'incohérence de position**. L'auteur s'intéresse aux réseaux

routiers et souligne que généralement l'appariement géométrique dans ce contexte est basé sur la recherche de correspondances au niveaux des noeuds, ce qui n'est pas une bonne solution vu que le problème est plutôt basé sur les lignes. En s'appuyant sur le fait qu'un réseau routier est composé de routes, elles-mêmes formées par un ou plusieurs arcs composés d'un ou plusieurs segments et limités par deux noeuds, il décrit alors une stratégie globale pour effectuer l'extraction des mises à jour. Son processus comporte 3 étapes : l'identification des couples d'arcs candidats à l'appariement, l'appariement géométrique de ces arcs et l'extraction des mises à jour. Cette extraction conduit à une classification des évolutions en trois catégories : les données inchangées, les données supprimées et les données ajoutées.

L'imprécision qui entache les données géographiques est une des principales sources d'erreurs que l'on peut rencontrer lorsqu'on utilise un processus d'appariement. En effet, il arrive que le processus détecte un lien entre deux objets qui ne correspondent pas en réalité au même phénomène dans le monde réel. Mais la principale limite du processus d'appariement reste la difficulté à le rendre automatisable. [Mustière *et al.*, 2004] pense qu'un appariement complètement automatique est certainement impossible car certains cas conduisent souvent à des résultats erronés (traitement de nombreux carrefours proches ou d'échangeurs complexes par exemple). En effet dans l'idéal, il faudrait définir un processus informatique capable de retranscrire l'appariement visuel de l'homme (notion intuitive de ressemblance) tout en évitant les conflits. Il faut par conséquent concevoir des outils pour évaluer et corriger interactivement les appariements réalisés automatiquement.

Thierry Badard [Badard, 2000] [Badard, 1998] a défini un processus d'appariement des données géographiques ayant des actualités différentes. Ce processus a été conçu avec un comportement ascendant et est quasi-automatique. Il procède d'abord à l'appariement des primitives géométriques des objets, puis des objets simples pour terminer par les objets complexes (i.e. groupes d'objets). Il interprète ensuite les résultats afin d'établir une typologie des évolutions spatio-temporelles, qui permet de traduire les évolutions subies entre deux actualités d'une même base de données. L'information obtenue est détaillée et proche des évolutions intervenant dans les bases de données géographiques, ce qui facilite ensuite l'intégration des mises à jour dans les bases de données utilisateur.

Bruno Tellez et Sylvie Servigne [Tellez et Servigne, 1998] proposent de mettre à jour automatiquement une base de données provenant du cadastre grâce à des photographies aériennes. Pour trouver l'information de mise à jour d'une base de données Raster et l'apparier aux données de la base de données vectorielle, ils proposent de définir un modèle commun aux deux représentations, en transformant le raster en vecteur. La difficulté est de tenir compte de la qualité et de la quantité d'information disponible. En effet, la photo contient plus d'objets que le cadastre et les mêmes objets ne sont pas forcément extraits dans le même groupe.

Pour pallier à ces difficultés, la notion **d'identifiant unique et pérenne** qui existe dans les bases de données classiques est apparue en information géographique. Le but est de définir pour chaque objet de la base, un identifiant qui soit unique,

qui ne disparaisse pas lorsque la base est transformée et qui ne change pas de valeur lorsque l'objet est mis à jour. A ce jour, le problème réside essentiellement dans la définition d'un identifiant qui soit unique et pérenne, ce qui implique, qu'en pratique, il y a peu d'identifiants qui ont été générés pour les données géographiques actuelles. De ce fait, à notre connaissance, aucune méthode générique n'a encore été proposée dans la littérature mais quelques outils applicatifs ont néanmoins été étudiés.

Le projet M@jic [Pouliot *et al.*, 2001] s'appuie sur des identifiants uniques universels propres à M@jic, pour trouver les concordances entre l'ancienne base et la nouvelle, mise à jour. Néanmoins, pour que cette technique soit automatique et pour obtenir un système fiable et stable, il faut au préalable transformer la base de données de l'utilisateur en base de données dans un format défini par le projet M@jic. Lorsque ce n'est pas possible, un appariement géométrique simple doit être utilisé (il compare la position des points constituants chaque élément à comparer). La description du type des mises à jour (stabilité ou changement) est quant à elle supervisée : le système propose un appariement que l'utilisateur doit confirmer, et selon le type de mise à jour effectué, le système pourra déduire par analyse de concordance des identifiants (s'ils existent) s'il s'agit d'une naissance, d'une mort ou d'une évolution et si ces manipulations correspondent à un effet direct ou en cascade (qui découlent d'autres effets).

Le projet Carto2001 [Lecordix *et al.*, 2005] en place à l'IGN utilise lui aussi des identifiants pour extraire les évolutions entre deux jeux de données. Les auteurs utilisent des empreintes numériques calculées à partir des attributs et de la géométrie de chaque objet de la base. L'empreinte obtenue est une clé produite par l'algorithme MD5 [Rivest, 1992]. Cet algorithme prend en entrée un message de longueur variable et produit en sortie une empreinte numérique de 128 bits. Il permet d'obtenir des signatures pour des documents électroniques avec une très faible probabilité que deux documents distincts aient la même signature. La stabilité est assurée si le calcul est effectué sur deux lots présentés dans le même référentiel et avec la même précision des données. Un objet est considéré comme ayant changé entre deux versions de la base lorsque les deux clés sont différentes. La difficulté est de savoir quels attributs doivent être considérés pour constituer l'empreinte afin qu'il n'y ait pas de recouvrements.

Un des enjeux de la mise à jour d'un jeu de données utilisateur par des évolutions provenant de sources multiples concerne la détection des évolutions conflictuelles. En effet, les mises à jour provenant de sources distinctes peuvent représenter le même phénomène du monde réel mais avoir été saisies dans un contexte, avec une qualité, ou selon un point de vue différents. D'autres conflits interviennent également lorsque deux évolutions sont localisées au même endroit mais ne représentent pas le même objet.

Nous pensons qu'une des issues pour détecter les problèmes de conflits entre évolutions réside dans **l'utilisation des identifiants**. Cela suppose de fournir aux utilisateurs, une base de données de référence dont chaque objet possède un identifiant qui aurait été généré au préalable. Malheureusement, nous verrons que cette technique a aussi ses limites (notamment lorsqu'il s'agit de gérer les évolutions de type création) et il faut alors **utiliser l'appariement en complément**.

Classification et livraison des évolutions

L'étape suivante pour gérer les mises à jour a pour objectif de rendre interprétables les mises à jour fraîchement extraites, par les différents systèmes utilisateurs. Une première phase consiste à définir **une typologie des évolutions** qui soit compréhensible par tous, puis à partir de cette classification, de fournir **un format de livraison** qui soit compatible avec les moyens matériels mis en oeuvre (réseau de communication, SIG, ...).

Ainsi, [Cheylan *et al.*, 1994] proposent de définir quatre types d'entités spatiales : les entités aux contours fixes, les entités modifiables, les entités déformables (les valeurs d'attributs, la forme et l'étendue peuvent varier) et les entités transformables (la position et la forme évoluent au cours du temps) et identifient ensuite les évolutions spatiales et attributaires sur chaque groupe d'entités. Cette classification reflète bien la réalité mais sa mise en oeuvre semble difficile.

Yvan Bedard [Bedard *et al.*, 1997] préfère utiliser uniquement les deux opérations de création et de destruction qui peuvent traduire à elles seules, tous les changements intervenus dans la base. Cette méthode est très simple à mettre en oeuvre mais il est alors impossible de connaître la nature réelle du changement subi par l'objet.

Christophe Claramunt [Claramunt *et al.*, 1994] définit une typologie plus riche que les précédentes. Elle se décompose en un ensemble de processus de base (apparition, disparition et stabilité), de processus de transformation (expansion, contraction et déformation) et de mouvements applicables aux objets (déplacement ou rotation). Cependant, elle ne permet pas de décrire aisément les évolutions de certains objets ou groupes d'objets comme par exemple la transformation d'un rond point en carrefour simple.

Katleen Hornsby et Max Egenhofer [Hornsby et Engenhofer, 2000] définissent un langage de description visuel sous forme d'icônes pour décrire les trois états possibles de l'identité d'un objet (cf Figure. 2.3 partie haute) : Existant (a), Non Existant sans Historique (b) et Non Existant avec Historique (c). La transition d'un état à un autre conduit à neufs combinaisons d'évolutions possibles (cf Figure. 2.3 partie basse) : Persistance de la non-existence sans historique (a), Création (b), Rappel (c), Destruction (d), Persistance de l'existence (e), Élimination (f), Oubli (g), Réincarnation (h) et Persistance de la non-existence avec historique (i). Cette représentation est intéressante pour distinguer la sémantique des différents chan-

gements qui sont intervenus au cours du temps et permet de décrire des scénarios d'évolution.

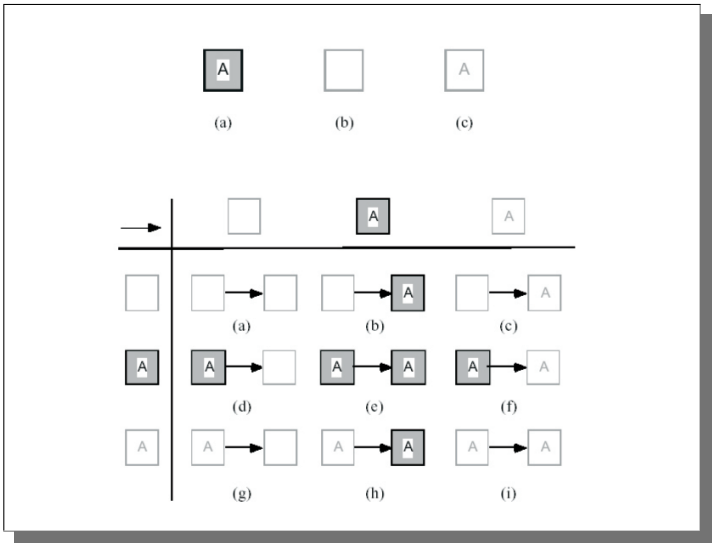

Figure 2.3 – Classification des évolutions selon [Hornsby et Engenhofer, 2000]

Thierry Badard [Badard, 2000] s'inspire de la classification proposée par C. Claramunt pour définir une typologie proche des évolutions qui peuvent intervenir dans les bases de données géographiques. Il identifie huit types d'évolutions : la création, la destruction, la scission, la fusion, l'agrégation, la modification géométrique, la modification sémantique et la modification mixte (cf Figure. 2.4). Cette typologie permet de traduire les évolutions subies par une ou un ensemble d'entités spatiales et de prendre en compte tous les types d'objets géographiques, ainsi que les modifications attributaires.

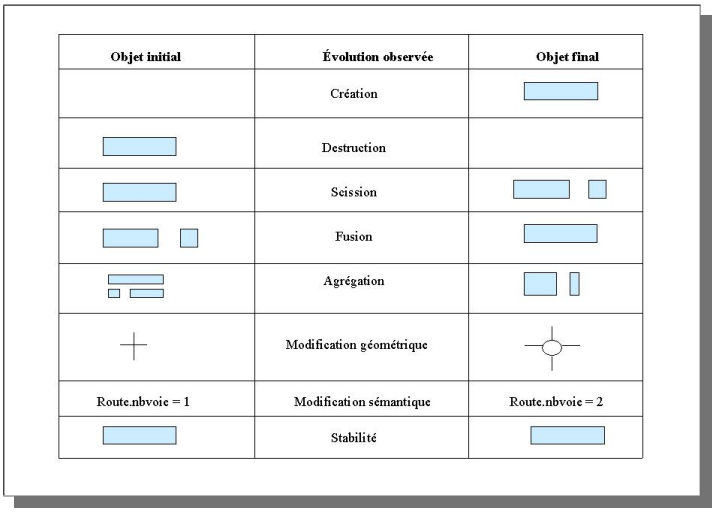

Figure 2.4 – Classification des évolutions selon [Badard, 2000]

La transmission des évolutions aux autres acteurs est l'étape qui fait suite à

l'extraction et à la classification. La diffusion des mises à jour peut s'effectuer selon trois formes de livraison :

- La diffusion globale qui consiste à transmettre le jeux de données mis à jour complet.
- La diffusion partielle d'un patch qui consiste à fournir uniquement la partie du jeu de données qui a été mise à jour (sur une zone ou un thème particulier).
- La diffusion incrémentale qui consiste à ne fournir que les changements survenus au niveau des objets, structurés sous la forme d'ordres élémentaires.

La diffusion entière de la base de données mise à jour pose premièrement des problèmes de coût et d'espace du fait de la grande quantité de données à fournir. Il semble donc difficile d'envoyer les informations de mise à jour sur un réseau ayant un faible débit et comme l'a souligné l'équipe du projet M@jic [Pouliot *et al.*, 2001], la plupart du temps la base entière est renvoyée par le biais d'un média tel que le CDRom. La diffusion globale engendre également une perte des liens avec les données ajoutées par l'utilisateur (perte de lien sémantique avec les données descriptives ajoutées ou perte de relations géométriques et topologiques avec les couches ajoutées par l'utilisateur), une perte de cohérence entre les données de référence et les données dérivées et enfin, ne permet par la construction d'un historique des modifications (inexistence de liens entre les anciennes et nouvelles données). Cela occasionne une perte de temps pour l'utilisateur qui doit de ce fait, reconstruire toutes les relations perdues avec le nouveau jeu de données.

La diffusion d'un patch de remplacement partiel des données semble particulièrement adaptée aux données Raster lorsqu'une partie de l'image qui a été fournie à l'utilisateur possède une actualité plus récente que l'image dans sa globalité. Une substitution du morceau d'image peut être effectuée avec la partie plus récente. En revanche, la diffusion d'une partie de mises à jour sur un jeu de données de type vectoriel va poser les mêmes problèmes que la diffusion globale mais les problèmes seront restreints à la zone couverte par le patch. Cela n'est néanmoins pas satisfaisant dans un contexte de prise de décision rapide où l'effort fourni pour le traitement des mises à jour doit être limité.

La diffusion incrémentale permet l'envoi des seules données ayant évoluées. Elle permet d'obtenir un gain de coût lors de l'envoi ce qui s'avère très intéressant sur un réseau où le débit n'est pas très élevé. En revanche, pour que les évolutions puissent être intégrées sans remettre en cause la cohérence du jeu de données, il faut que les données et les mises à jour soient dans un schéma commun. Un effort de transformation des mises à jour doit donc être effectué afin que les évolutions fournies soient rendues compatibles avec les données du jeu de données utilisateur. Le principal problème est qu'il n'existe pas de format standard permettant la diffusion incrémentale des évolutions. Cependant, quelques recherches ont permis d'aboutir à des formats ad-hoc :

[Ding *et al.*, 2004] définissent un format de livraison générique pour faciliter le transfert des évolutions entre l'utilisateur et le fournisseur de données qui peuvent avoir des plates-formes différentes. Trois types de mises à jour peuvent être échangées

entre ces deux utilisateurs : la mise à jour (correspondant à la modification), l'insertion et la création. Les évolutions structurées selon ce format sont ensuite transférées dans un fichier texte.

Le projet de recherche ActMAP [Otto *et al.*, 2004] avait pour but de développer des solutions pour la livraison de mises à jour incrémentales de cartes utilisées sur des applications dédiées aux véhicules. Les auteurs ont constaté que les formats des données des producteurs sont souvent différents de ceux des utilisateurs, du fait de la diversité des systèmes utilisés, ce qui pose de nombreux problèmes lors de la mise à jour des cartes. Pour permettre l'échange des mises à jour entre les fournisseurs et les utilisateurs, ils proposent alors de créer un format intermédiaire de description des mises à jour. Pour ce faire, ils ont défini un format standard ouvert d'échange des mises à jour basé sur XML qui permet d'avoir un format générique pour la livraison des évolutions. La figure 2.5 montre le contexte d'utilisation de ce format. Les données provenant des fournisseurs sont transformées puis stockées dans un magasin virtuel, permettant ainsi aux utilisateurs de rechercher les informations de mise à jour. Les évolutions trouvées sont ensuite livrées dans le format d'échange puis transformées dans le format propriétaire des utilisateurs. Un des inconvénients de ce système est que les données en entrée doivent être traduites dans le format interne pour que le système ActMAP puisse exécuter les opérations internes dessus et ensuite retraduites dans la représentation propriétaire du véhicule afin de pouvoir être traitées par l'utilisateur. Cette méthode reste néanmoins très intéressante au vu des contraintes qu'implique la mise à jour de systèmes embarqués dans les véhicules.

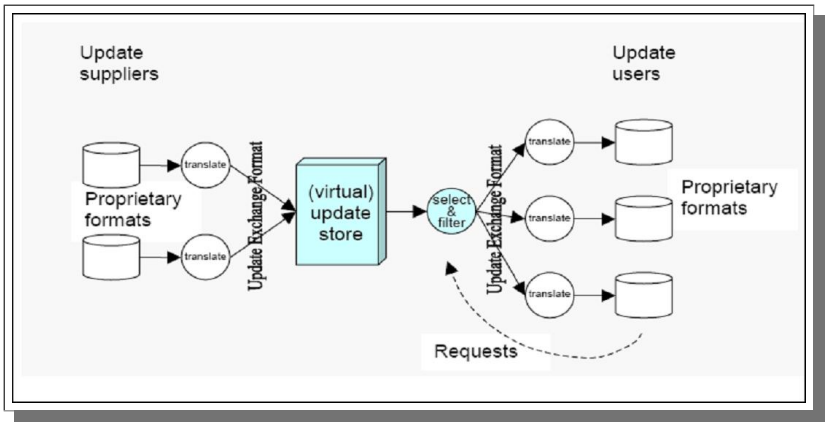

Figure 2.5 – Contexte d'utilisation du format de livraison défini dans [Otto *et al.*, 2004]

Thierry Badard définit, quant à lui, **les lots différentiels** [Badard, 1998]. Ceux-ci sont fondés sur la différence d'état des objets entre deux actualités d'une base de données géographique (cf Figure. 2.6). Deux fichiers composent les lots différentiels :

- Un fichier contenant les couples d'objets anciens et nouveaux livrés dans un format SIG propriétaire.
- Un fichier contenant les mises à jour qu'a subies chaque objet suivant la typologie qu'il a définie.

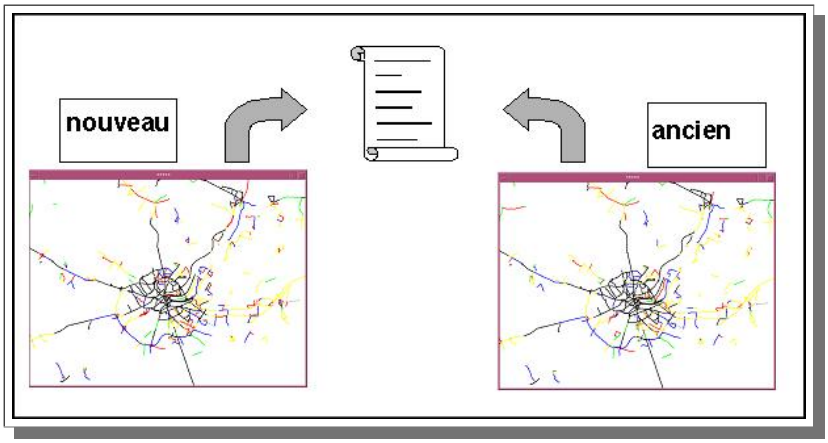

Figure 2.6 – Différentiel d'état

Des travaux ont été entrepris à l'IGN en vu de quantifier le volume représentatif des données de mise à jour. L'étude a été menée sur deux bases de données de référence, Géoroute et BDPays, sur le département de l'Isère. Les données sont au format *shape* et seul le thème routier a été mis à jour. Les bases de données résultantes de la mise à jour pèsent 106Mo pour BDPays et 79Mo pour Géoroute. [Braun, 2003] montre que le volume de données tombe à 26 Mo pour la BD Pays et 16 Mo pour Géoroute si les évolutions sont structurées en lots différentiels, ce qui représente en moyenne 20% des jeux initiaux.
Ce format de livraison a donc fait ses preuves, notamment à l'IGN où il a été testé sur plusieurs bases de données mais le volume de données qui transite reste quand même très important et l'information délivrée n'est pas toujours minimale.

[Badard et Richard, 2001] définissent ensuite **les lots d'évolution** qui se basent sur la description des changements d'états entre des ensembles d'objets (cf Figure. 2.7). Dans ce format, seule l'information permettant de définir ce qui a évolué sur l'objet est transmise. Les lots d'évolutions sont décomposés en 3 parties :

- Les évolutions proprement dites qui sont décrites grâce à la typologie définie dans [Badard, 1998].
- Les métadonnées qui fournissent toutes les informations de contexte décrivant les deux versions livrées.
- Les références aux objets, valeurs d'attributs, relations et primitives géométriques qui ont été modifiés entre les deux versions.

La livraison par lots d'évolution s'appuie sur le format XML (eXtensible Markup Language). Ce langage à base de méta-balises, solution quasi incontournable pour l'échange des évolutions entre SIG, permet de structurer plus fortement l'information d'évolution que dans les lots différentiels et permet de garder une certaine interopérabilité entre les systèmes. Néanmoins, les lots d'évolution restent difficiles à intégrer car ils doivent être interprétés pour pouvoir être incorporés.

```
?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- dep22evol4.xml -->
<!DOCTYPE EvolElem SYSTEM "Evolutions.dtd">
<EvolElem ID="_4">
 <Evol.Inst TYPE="Objet">
 <Evol.DestInst>
  <Evol.Evolution TYPE="Detruire" OPER="Identifier">
   <Evol.Sem>
    <Evol.RefInst xlink:title="Modification d'instance: Destruction d'un objet.">
     <Evol.link xlink:href="dep22rte.xml|ROUTES94_ROUTE_21866" />
    </Evol.RefInst>
   </Evol.Sem>
  </Evol.Evolution>
 </Evol.DestInst>
 </Evol.Inst>
</EvolElem>
```

Figure 2.7 – Exemple de lot d'évolutions défini par [Badard et Richard, 2001]

Les lots d'évolution sont à ce jour **les plus prometteurs** en matière de format de livraison des données géographiques vectorielles. En effet, **l'information délivrée est minimale** car seuls les attributs modifiés sont transmis au travers d'un message. Elle est donc moins redondante que dans les lots différentiels.

Intégration et propagation des évolutions

L'utilisateur ayant reçu les mises à jour doit ensuite intégrer et répercuter les changements dans son jeu de données. **L'intégration** consiste à transformer les données provenant de sources hétérogènes afin qu'elles forment un tout cohérent et homogène [OQLF, 2004], [Sheeren, 2005]. Beaucoup d'auteurs s'entendent pour dire que l'intégration est une des tâches les plus complexes que l'on rencontre en information géographique, du fait de la diversité des conflits (au niveau géographique, sémantique ou encore temporel) qui peuvent provoquer des incohérences conséquentes entre les données [J.Pouliot *et al.*, 2001]).

Selon [Sheeren, 2005], ce qui rend le problème d'intégration particulièrement complexe, c'est l'existence de **l'hétérogénéité des bases**. Les bases de données dans lesquelles les données sont intégrées peuvent être différentes à plusieurs niveaux (SGBD [1], format, modélisation ou interprétation différents). Cela est d'autant plus important pour les bases de données géographiques pour lesquelles s'ajoutent les différences liés à la composante géométrique telles que le mode de représentation des données (vecteur/raster), le niveau de détail ou encore la qualité des informations contenues dans la base de données. Le problème s'accentue encore lorsque l'intégration concerne des données de mises à jour qui proviennent de sources diverses et hétérogènes.

Pour remédier à cela, Thomas Devogèle [Devogele, 1997] propose l'extension d'une méthode d'intégration déclarative utilisée dans les bases de données classiques (BD), définie par Spaccapietra et Parent [Spaccapietra et C.Parent, 1996]

1. Systèmes de Gestion de Bases de Données

[Spaccapietra et al., 1992] (cf Figure. 2.8). Ce processus est composé de trois phases : **la pré-intégration, l'identification des correspondances et l'intégration**. Ces trois phases sont enrichies afin que le processus puisse être adapté aux bases de données géographiques (BDG). L'extension est fondée sur une taxonomie des conflits d'intégration entre BDG et sur l'ajout d'un processus d'appariement géométrique et topologique.

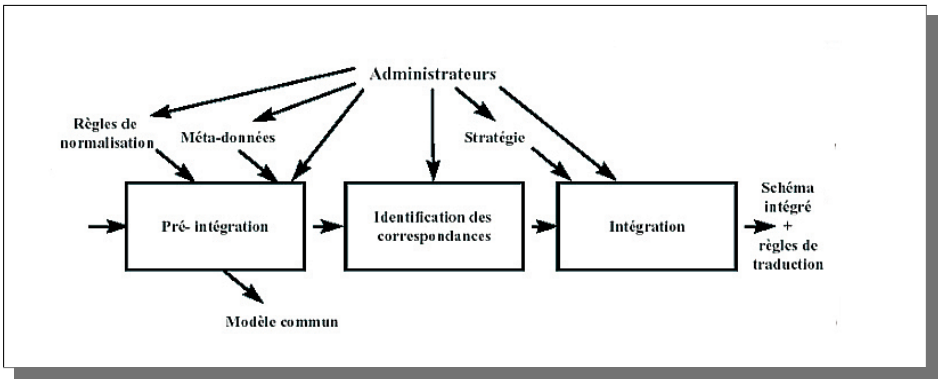

Figure 2.8 – Processus d'intégration des mises à jour selon S.Spaccapietra

Thierry Badard et Cécile Lemarié [Badard et Lemarié, 1999] intègrent, quant à eux, séparément chaque évolution préalablement filtrée (le filtrage consiste à sélectionner les évolutions qui ne concernent que la base de données dérivée), puis propagent ces évolutions aux objets corrélés. Ils gèrent ensuite les conflits et vérifient en dernier lieu la cohérence.
Les différentes étapes et les éléments nécessaires à ce processus sont indiqués dans la figure 2.9 :

- **Le filtrage** est partiellement automatisé et consiste à ne sélectionner que les évolutions concernant la BD dérivée. Pour cela, trois types d'informations sont utilisés : Les métadonnées disponibles sur la base de données dérivées pour déterminer si la modification est pertinente en comparaison avec la résolution de la base, le processus de dérivation pour vérifier si une modification de la base de référence a un impact dans la base dérivée, et les relations de correspondances établies préalablement entre les deux bases, pour déterminer si l'objet mis à jour dans la base de référence est présent dans la base dérivée.
- Chaque évolution préalablement filtrée est **intégrée** séparément dans la base de données dérivée. La manière d'intégrer les mises à jour dépend du type des évolutions, de la cardinalité des relations de correspondances et du processus de dérivation.
- La phase de **propagation** se déroule en deux étapes : retrouver les objets corrélés avec ceux qui viennent d'être mis à jour et propager les mises à jour automatiquement si le processus de dérivation est parfaitement défini, sinon c'est à l'utilisateur de choisir s'il veut effectuer la mise à jour des objets corrélés ou non.
- **Les conflits** sont détectés pendant l'intégration et/ou la propagation. Ils surviennent lorsque les contraintes d'intégrité spatiales sont violées ou lorsque des

mises à jour n'ont pas été effectuées. L'utilisateur doit alors, soit résoudre ces conflits interactivement, soit enrichir la base de règles qui permet de traiter les conflits automatiquement et relancer les deux processus précédents.

– Le processus de **vérification de la cohérence** peut entraîner des retours en arrière ou des reprises de certaines étapes après enrichissement de la base de règles. Les différents tests mis en place pour cette phase sont la vérification de la topologie de la base et la mise en conformité avec les métadonnées qui ont été mises à jour.

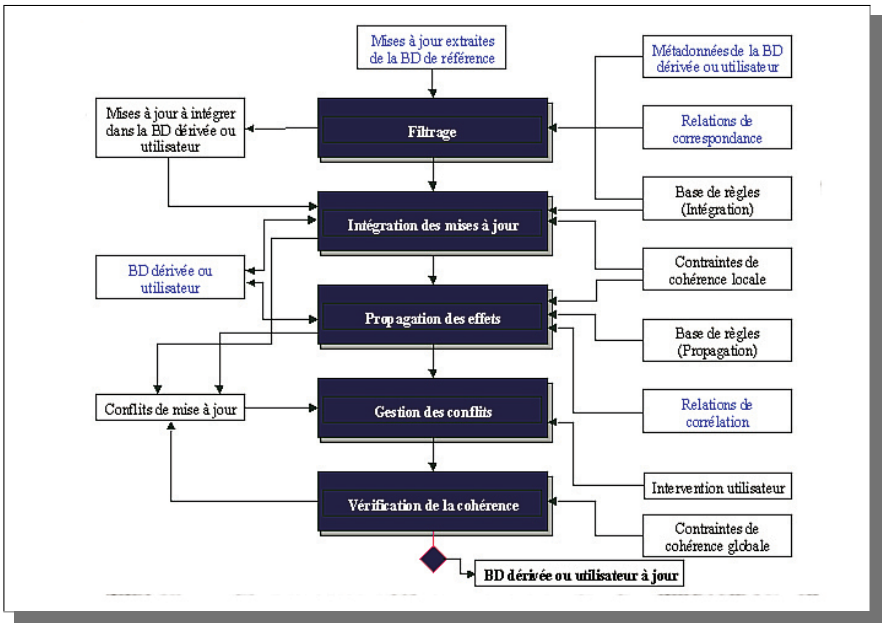

Figure 2.9 – Processus d'intégration des mises à jour selon [Badard et Lemarié, 1999]

Cette méthode n'est pas entièrement satisfaisante car elle nécessite l'aide fréquente de l'expert. De plus, elle s'avère compliquée à mettre en œuvre car elle est essentiellement basée sur les appariements et elle demande de nombreux retours-arrières.
Néanmoins elle offre de bons résultats en matière de cohérence et l'idée de faire un pré-filtrage est intéressante pour "épurer" toutes les données ne concernant pas directement la base dérivée.

Alors que [Badard, 2000] propose l'utilisation d'outils d'appariement pour détecter les liens de correspondance entre les différents thèmes des jeux de données et entre les données à différentes échelles pour faciliter la propagation des évolutions, d'autres proposent l'utilisation de **bases de données multi-représentations**. Ces bases de données permettent de représenter plusieurs visions du même espace, soit à différentes échelles, soit selon différents points de vue, dans une seule et même base de données. Le but étant de passer d'une représentation à l'autre le plus facilement possible [Vangenot *et al.*, 2002]. Ainsi, [Kilpelainen, 2000] propose d'établir des liens bidirectionnels entre les différents objets représentés dans de telles bases

de données. L'idée étant qu'en conservant tous ces liens, il est plus aisé de retrouver un objet correspondant à une évolution quel que soit le niveau de représentation. Cela permet également de propager automatiquement, sur le jeu de données dérivé, les mises à jour qui ont été saisies depuis la base de données de référence.

Une autre solution est l'utilisation des **bases de données fédérées**. Une base de donnée fédérée est une vue commune de plusieurs bases de données qui permet une coopération entre ces bases. Il faut donc créer un schéma commun à toutes les bases de données, ainsi que les règles de passage permettant à un schéma particulier d'accéder à celui de la base fédérée. [Christensen, 2001] propose en particulier de créer une base de données géographique fédérée permettant de regrouper les caractéristiques communes à certains objets provenant de différentes collectes.

Dans notre contexte, nous ne pouvons malheureusement pas utiliser les bases de données fédérées car les données sont répliquées sur chaque site. En effet, aucun serveur centralisant l'information ne peut être mis en place et les données doivent être disponibles à tout moment.

2.1.3 Infrastructures de données spatiales, interopérabilité et métadonnées

La mise à jour peut être facilitée par la mise en place d'une infrastructure dans laquelle une politique d'échange des évolutions peut être mise en oeuvre. Dans cette partie, nous définissons en premier lieu la notion **d'infrastructure de données spatiales** et voyons en quoi leur utilisation favorise **l'interopérabilité**. En particulier, nous nous intéressons aux **métadonnées** et aux différents standards qui existent pour la gestion des données spatiales.

Infrastructure de données spatiales, pour qui, pourquoi, comment ?

De nos jours, de nombreux acteurs sont impliqués dans la collecte et la distribution des données spatiales. Cela entraîne une multiplication des données ayant des types, des formats et des qualités différents. Cette redondance d'information pose le problème de l'accès, l'échange et de l'utilisation des données provenant d'organisations multiples. En effet, les données collectées auprès de divers fournisseurs ne peuvent généralement pas être intégrées sans avoir subi de nombreuses transformations qui les rendent interprétables par le système de l'utilisateur. Ce travail de transformation entraîne souvent une mise en oeuvre et un coût important, non souhaitables dans un contexte de prise de décision rapide. Une solution consiste à regrouper, dans une infrastructure, les acteurs ayant un objectif analogue ou appartenant à une même organisation, afin que des décisions communes puissent être prises, notamment en matière de formatage, de partage et d'échange de données spatiales.
[Vögele et Schlieder, 2002] soulignent d'ailleurs que le rôle important que jouent les données spatiales dans un certain nombre d'applications implique un besoin grandissant d'avoir des infrastructures qui savent gérer l'accès et l'échange de ces données.

Une infrastructure de données spatiales (SDI) [2] est une initiative prévue pour créer un environnement dans lequel tous les acteurs peuvent coopérer et interagir les uns avec les autres pour mieux atteindre leurs objectifs à différents niveaux politiques et administratifs [Rajabifard et Williamson, 2001]. En particulier, l'utilisation d'une SDI facilite et coordonne l'accès, l'échange et le partage de données géographiques entre plusieurs acteurs, tout en utilisant un ensemble minimum de standards, protocoles et spécifications [Nebert, 2004].

> Une SDI permet donc de déterminer la façon dont les données vont être utilisées à l'intérieur d'une organisation, de spécifier les politiques d'échanges qui peuvent être appliquées entre les communautés et surtout d'obtenir un gain de temps lors de la génération et de la maintenance des données.

Les principales composantes d'une infrastructure de données spatiales sont les acteurs (fournisseurs ou utilisateurs de données), le mode de communication établie entre les acteurs, les jeux de données de référence, les politiques appliquées à l'intérieur de l'infrastructure et les métadonnées et standards associés [Coleman et Nebert, 1999] (Cf. Figure 2.10).

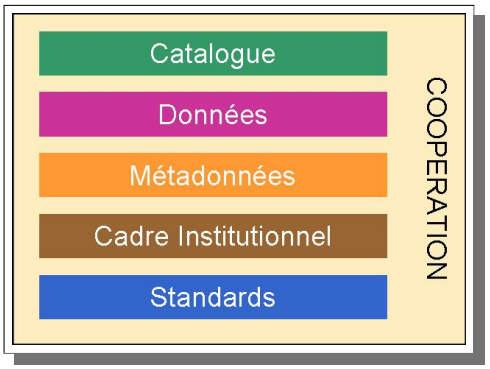

Figure 2.10 – Principaux composants d'un SDI selon [Nebert, 2004]

[Nebert, 2004] souligne d'ailleurs qu'en fixant **les rôles et responsabilités des acteurs**, on est alors capable d'identifier les fonctions de chaque utilisateur dans l'infrastructure et particulièrement de définir les autorisations en matière de transformation, mise à jour, diffusion et intégration des données spatiales.

Le mode de coordination entre les acteurs permet, quant à lui, de définir l'architecture du réseau de communication et le type de collaboration entre les différents acteurs de l'infrastructure.

Les jeux de données sont des ensembles spécifiques d'informations spatiales ou des collections de données fournies par un producteur ou saisies directement depuis un système d'information géographique. Dans une SDI, les données de référence

2. On utilise souvent l'acronyme SDI correspondant au terme anglais Spatial Data Infrastructure pour désigner une infrastructure de données spatiales

sont constituées par un ensemble de jeux de données nécessaires à l'intérêt commun.

Un des avantages majeur d'une SDI est la possibilité de déterminer **un cadre institutionnel** visant à accroître l'accès, le partage et l'utilisation efficace de l'information géographique [Craglia et Johnston, 2004]. Ainsi, **des stratégies** peuvent être établies à différents niveaux de l'infrastructure et dépendre du rôle attribué aux acteurs, être fonction du réseau de collaboration, ou encore être attaché au type d'information utilisé. Elles ont en outre l'intérêt de restreindre les actions non souhaitées et limitent par conséquent les éventuelles incohérences entre les données échangées, particulièrement dans un contexte de mise à jour.

Les métadonnées forment un ensemble formel de propriétés descriptives qui peuvent être partagées par une communauté. En particulier elles renseignent sur la nature et les caractéristiques des données. L'utilisation des métadonnées permet dès lors une gestion pertinente des données car elles fournissent à l'utilisateur certaines informations capitales telles que la disponibilité, la qualité, ou encore la localisation [Gilgen, 1999].

Des standards de description des métadonnées ont été développés suivant un processus consultatif et fournissent de ce fait une base commune aux utilisateurs [CEN, 1998], [FGDC, 1998], [ISO19115, 2003]. L'utilisation de métadonnées normalisées dans l'infrastructure **favorise donc l'interopérabilité** entre les différents acteurs et systèmes.

Par ailleurs, [Chan et Williamson, 1999] ont défini **un modèle hiérarchique** de SDI composé de SDI interconnectés à différents niveaux (local, national, régional, global). Ce modèle permet ainsi aux organisations d'être en relation à différents niveaux politiques et administratifs (Cf. figure 2.11).

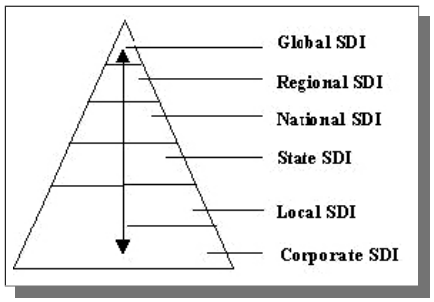

Figure 2.11 – Modèle hiérarchique d'un SDI défini par [Chan et Williamson, 1999]

Basées sur ce modèle hiérarchique, plusieurs initiatives d'infrastructures de données spatiales ont vu le jour ces dernières années, dans un premier temps, au niveau national ([FGDC, 1997], [Masser, 1998], [ANZLIC, 1998], puis aux niveaux régional et global [Rhind, 2001], [INSPIRE, 2007]).

Ainsi, l'Australie et la Nouvelle Zélande ont défini une SDI nationale qui comprend quatre composants [ANZLIC, 1998] :

- Un cadre institutionnel pour définir les arrangements politiques et administratifs permettant de construire, mettre à jour et utiliser les standards et données.
- Des standards pour définir les caractéristiques techniques des jeux de données de référence.
- Des jeux de données fondamentaux produits grâce au cadre institutionnel et conformes aux standards techniques.
- Et enfin des réseaux d'entrepôts permettant à la communauté d'accéder aux jeux de données.

Le comité fédéral de données géographiques américain [FGDC, 1997] a constitué une SDI pour faciliter le transfert et l'utilisation des données spatiales entre les diverses organisations nationales. L'infrastructure est composée de politiques, standards et procédures qui permettent aux organisations nationales d'interagir pour une meilleure utilisation, gestion et production des données.
L'infrastructure de données spatiales nationale mise en place en Allemagne (RAVI) [Masser, 1998] est constituée d'une collection de politiques, jeux de données, standards, technologies et connaissances permettant de fournir à l'utilisateur l'information géographique dont il a besoin pour réussir une tâche.

Plus proche de nos préoccupations, l'initiative européenne [INSPIRE, 2007] (www.ec-gis.org/inspire), a été lancée à la fin de l'année 2001 avec l'intention de rendre l'information géographique accessible, pertinente, harmonisée et de qualité. L'objectif était de parvenir à un cadre juridique européen, en se concentrant d'abord sur les besoins de la politique de l'environnement, pour être ensuite étendu à d'autres domaines d'intérêt communautaire tels que l'agriculture, la politique régionale, et les transports. L'initiative INSPIRE vise à créer des services d'information spatiale, basés sur un réseau distribué de bases de données, dans lequel les jeux de données peuvent être combinés avec différentes sources et partagés entre de nombreux utilisateurs et applications, favorisant ainsi l'interopérabilité entre les différents systèmes européens.

Interoperabilité des Systèmes

L'organisation internationale de normalisation (ISO) définit **l'interopérabilité** comme étant un ensemble de capacités. Premièrement, c'est l'aptitude à trouver des informations et des outils de traitement, quand on en a besoin, peu importe où ils se trouvent physiquement. Secondement, c'est la faculté de comprendre et d'utiliser l'information et les outils quelle que soit la plate-forme sur laquelle ils reposent. Enfin, c'est la possibilité de participer à un marché où les biens et les services sont adaptés aux besoins des utilisateurs.

Selon [Bishr, 1997], l'interopérabilité se traduit par la capacité d'un système ou des composants d'un système à partager ses données et ses fonctions avec d'autres systèmes. Plus précisément, l'interopérabilité est assurée si des messages et des requêtes peuvent être échangés entre deux systèmes et s'il est possible de les faire opérer comme une unité pour réaliser une tâche commune [Solar et Doucet, 2002]

[Danko, 2005] souligne l'importance de considérer l'interopérabilité par le fait que de nos jours plus personne ne travaille seul et que les efforts doivent être mutualisés au sein d'organisations. Selon l'auteur, l'interopérabilité est la clé pour améliorer la communication, l'efficacité et la qualité des échanges entre toutes ces organisations qui coopèrent.

[Leclercq *et al.*, 1998] distinguent l'interopérabilité des systèmes d'information classiques de celle que l'on trouve dans les systèmes d'information géographiques (SIG). Ils soulignent en particulier qu'en information géographique, les problèmes d'hétérogénéité sont causés également par la géométrie et que le caractère spatial des données accentue les conflits de classification ou de fragmentation. Ils définissent ensuite trois niveaux d'interopérabilité des modèles que l'on rencontre dans les SIG :

- **L'interopérabilité syntaxique** qui s'attache à unifier les structures des données.
- **L'interopérabilité schématique** qui a pour but de résoudre les différences au niveau des schémas.
- **L'interopérabilité sémantique** qui doit s'assurer que les échanges ont un sens, c'est à dire que l'utilisateur et le fournisseur ont une compréhension commune de la signification des éléments qu'ils partagent.

Jean Brodeur [Brodeur, 2004] constate que depuis les années 1990, de nombreuses recherches ont été effectuées pour développer l'interopérabilité des systèmes d'informations géographiques afin de résoudre les problèmes d'échanges et d'intégration des données [Bishr, 1997] [Laurini, 1999] [Sheth, 1999]. L'auteur remarque que des progrès considérables ont été faits en ce qui concerne l'hétérogénéité syntaxique et schématique [Egenhofer, 1999] [Rodriguez, 2000], néanmoins, il souligne que l'hétérogénéité sémantique doit également être prise en considération dans la solution pour qu'une complète interopérabilité des systèmes d'information géographiques soit assurée.

Pour Bruno Defude [Defude, 2005], les deux formes d'hétérogénéités qui entravent essentiellement l'interopérabilité des systèmes sont l'hétérogénéité syntaxique et l'hétérogénéité sémantique. L'hétérogénéité syntaxique souvent étudiée dans la littérature pose le problème de l'accès uniforme à un ensemble de sources hétérogènes. L'hétérogénéité sémantique est bien plus complexe car il s'agit de comparer et reconnaître les similitudes et les différences entre les concepts utilisés dans les différents systèmes.
Dans le cas particulier des SIG, l'auteur différencie l'interopérabilité technique qui se traduit par la capacité à échanger des requêtes et des données entre les différents SIG, de l'interopérabilité sémantique qui se traduit par la capacité à manipuler de manière cohérente les informations provenant des différents SIG (par exemple reconnaître qu'un segment quelconque d'un premier SIG correspond en fait à une route dans un second).
L'auteur propose donc une architecture répartie de type fédération comprenant la définition d'un modèle canonique et une base de connaissances des conflits spatiaux afin d'aider le processus d'intégration et un modèle sémantique permettant de

définir des données spatiales, numériques et cartographiques ainsi que les transformations qui manipulent ces données afin de traiter l'hétérogénéité sémantique dans les systèmes d'informations géographiques [Branki et Defude, 1997].

Le projet de recherche ISIS (pour Interopérabilité des Systèmes d'Information Spatiale) a pour but de trouver une solution pour réaliser l'interopérabilité des SIG dans des environnements ouverts et évolutifs. La thèse de [Leclercq, 2000] effectuée dans le cadre de ce projet a pour objectif principal de fournir des accès transparents à des systèmes d'information géographiques hétérogènes en exploitant la sémantique des informations spatiales. L'auteur utilise pour cela une approche basée sur la médiation de contextes (contexte de référence, contextes de coopération et outils de transformation) qu'il définit grâce à des ontologies.

Plus récemment, [Gesbert, 2005] a proposé de traiter l'interoperabilité sémantique grâce aux ontologies. L'objectif de ce travail est de déterminer les correspondances entre les schémas des différentes bases de données géographiques en s'appuyant sur la sémantique contenue dans les spécifications, afin d'aider l'intégration et de favoriser l'interopérabilité. Pour cela, il propose de définir un modèle formel de représentation des spécifications basé sur une ontologie du domaine qui facilite la compréhension de la sémantique.

Les métadonnées, composant indispensable d'une SDI

Les **métadonnées** constituent un des composants principaux des infrastructures de données spatiales. Le terme *métadonnée* vient du mot grec *meta* qui signifie « à propos » et du mot latin *data* qui signifie « information ». Souvent définies simplement comme étant des « données sur les données », elles ont en fait un rôle essentiel d'information sur la nature d'autres données permettant ainsi leur utilisation pertinente [OQLF, 2004]. Les métadonnées sont par conséquent, utilisées dans tous les domaines où de l'information complémentaire est nécessaire pour exploiter les données à bon escient.

[Bucher, 2002] précise qu'il existe trois niveaux d'utilisation de métadonnées : **les métadonnées de découverte** qui permettent de retrouver les données correspondant à une recherche, **les métadonnées d'exploration** qui permettent de savoir si les données conviennent à l'analyse qui va être faite et **les métadonnées d'exploitation** qui renseignent sur les moyens d'obtenir et d'utiliser les données. [Libourel, 2003] indique que les métadonnées peuvent être utilisées à des fins diverses, depuis l'aide à la structuration et à la recherche d'information jusqu'à des fonctionnalités plus complexes dans le cadre d'applications interopérables. [Tsou, 2002] exploite quant à lui les métadonnées pour aider les utilisateurs de systèmes informatiques à accéder, archiver et utiliser les informations réparties. En particulier, il utilise des métadonnées opérationnelles pour développer des services autonomes pour la description et la gestion des objets géographiques dans un environnement distribué.

En information géographique, les métadonnées sont plus complexes que celles

utilisées dans les autres domaines du fait de la nature spécifique des informations qu'elles renseignent. En effet, elles sont composées à la fois d'éléments descriptifs et d'éléments spatiaux et apportent différents renseignements tels que l'identification, l'étendue, la qualité, les schémas spatiaux et temporels ou encore la distribution des données spatiales [Danko, 2005].

Selon [Nebert, 2004], les métadonnées aident les utilisateurs de données spatiales à trouver les données dont ils ont besoin et à déterminer comment les employer au mieux. En particulier, elles permettent d'organiser et de maintenir l'investissement effectué par une communauté afin de fournir suffisamment d'information pour aider le transfert de données entre les différents acteurs de l'infrastructure [Nogueras-Iso *et al.*, 2005].

[Danko, 2005] insiste sur **l'importance croissante de l'utilisation des métadonnées** par le fait que l'information géographique est de plus en plus utilisée dans diverses organisations notamment à des fins de prise de décision. Cela entraîne une prolifération des données spatiales et un accroissement d'utilisateurs de données non experts. Sachant que les données géographiques sont imparfaites et incomplètes (précision, exhaustivité), les utilisateurs doivent être en mesure de connaître leur qualité avant toute utilisation. De plus, le producteur, qui doit favoriser la réutilisation des données existantes du fait de l'importance des coûts de production d'un jeu de données, ne connaît pas a priori l'utilisation qui sera finalement faite des données.

Les métadonnées permettent donc de **faciliter la recherche, la gestion et la réutilisation** des données géographiques.

Mais pour fournir une connaissance partagée et cohérente des données entre les différentes communautés, il faut **normaliser** les métadonnées [Luzet, 1998], [Günther et Voisard, 1997], [Spéry et Libourel, 1998]. En conséquence, plusieurs normes de métadonnées consacrées aux données spatiales ont été définies pour assurer l'interopérabilité entre des utilisateurs manipulant de l'information géographique [FGDC, 1998], [CEN, 1998], [ISO19115, 2003].

Ainsi, le comité européen de normalisation [CEN, 1998] a établi une norme expérimentale pour manipuler plus efficacement l'information géographique (CEN/TC287). Aux Etats-Unis, le standard FGDC/CSDGM (Content Standard for Digital Geospatial Metadata) a été développé par le comité fédéral des données géographiques, dans le but d'être utilisé par une infrastructure de données spatiale nationale (NSDI) [FGDC, 1998]. Plus récemment, le comité international de normalisation (ISO) a lui aussi proposé une norme pour la gestion et l'échange des données spatiales [ISO19115, 2003].

Ces standards ont en commun la structuration sous forme de couple (champ, valeur) et l'organisation des informations en sections contenant des caractéristiques similaires (identification, systèmes de référence, qualité, ...). Cependant, la norme qui retient le plus l'attention à ce jour est [ISO19115, 2003] qui spécifie un cadre

formel pour décrire l'information géographique et les services associés. Cette norme fournit beaucoup d'informations sur les jeux de données spatiaux et permet par conséquent la description de nombreuses ressources mais elle est difficile à exploiter du fait du très grand nombre d'éléments à gérer (plus de 350 éléments). Cependant, l'ISO 19115 offre la possibilité de créer **des profils** par extension et par restriction de la norme, ce qui amène de plus en plus d'organismes tels que l'armée française à l'utiliser pour créer leur profil communautaire [METAFOR, 2005].

Néanmoins, les métadonnées définies dans les standards s'attachent surtout à fournir des informations sur la description du produit. Elles sont développées du point de vue du producteur et contiennent peu de description du sens des données [Herdorfer et Bianchin, 1998], ni d'information exploitable par l'utilisateur sur l'utilisation qui peut en être faite [Bucher, 2002]. Cela pose en particulier le problème de **la pertinence des données pour l'utilisateur final**, notamment en ce qui concerne l'adéquation aux besoins.

2.1.4 Cohérence et qualité des bases de données spatiales

En information géographique, la notion de cohérence est étroitement liée à la qualité des jeux de données et peut être remise en cause par de nombreuses sources d'erreurs. Dans un contexte de mise à jour, c'est la capacité à satisfaire un ensemble de critères lors de l'intégration et/ou la propagation des nouvelles données et évolutions.
Dans ce paragraphe, nous abordons dans un premier temps les concepts relatifs à la qualité des bases de données géographiques, puis nous évoquons les différentes sources d'erreurs qui peuvent conduire à des incohérences locales dans les ensembles de données et enfin nous terminons par un inventaire des moyens qui ont été mis en oeuvre pour gérer la cohérence globale entre plusieurs jeux de données, notamment les mécanismes de versionnement.

Qualité des jeux de données spatiaux

Une définition générale de **la qualité** a été fournie par l'Organisation Internationale de Normalisation par « l'ensemble des propriétés et caractéristiques d'un produit ou service qui lui confère l'aptitude à satisfaire des besoins exprimés ou implicites » [ISO8402, 1994], [ISO9000, 2000].
Dans son travail de thèse, [Bel-Hadj-Ali, 2001] souligne que la qualité des données géographiques est tellement complexe qu'il est impossible d'utiliser une mesure globale et qu'il faut par conséquent recourir à plusieurs composantes pour la déterminer.

La définition de l'ISO a été spécialisée par [David et Fasquel, 1997] pour définir la qualité en information géographique. Ainsi, les auteurs distinguent deux types de qualité pour les données spatiales :

- La **qualité interne** est l'ensemble des propriétés et caractéristiques d'un produit ou service qui lui confère l'aptitude à satisfaire aux spécifications de contenu de ce produit ou de ce service.

– La **qualité externe** est définie comme étant l'adéquation des spécifications aux besoins de l'utilisateur.

La figure 2.12 illustre ces deux types de qualité. En information géographique, le monde est représenté par une abstraction de la réalité appelé **terrain nominal**, entité qui sert de base à l'élaboration de la base de données (grâce aux spécifications de production ou depuis les exigences des utilisateurs). Évaluer la qualité des données correspond à estimer l'écart en terme de distance entre la base de données et le terrain nominal. Plus précisément, la qualité interne se mesure par la différence entre les données qui devraient être produites et les données qui ont effectivement été produites. Elle est liée aux spécifications (et en particulier aux erreurs qui peuvent être commises lors de production des données) et est évaluée en fonction du producteur. La qualité externe se mesure quant à elle, par la différence entre les données souhaitées par l'utilisateur et les données effectivement produites. Elle est liée aux besoins des utilisateurs et varie donc d'un utilisateur à l'autre.
Finalement, en simplifiant, l'évaluation de la qualité revient à vérifier la conformité entre les données de la base de données et les données considérées comme justes (du point de vue du producteur ou de l'utilisateur).

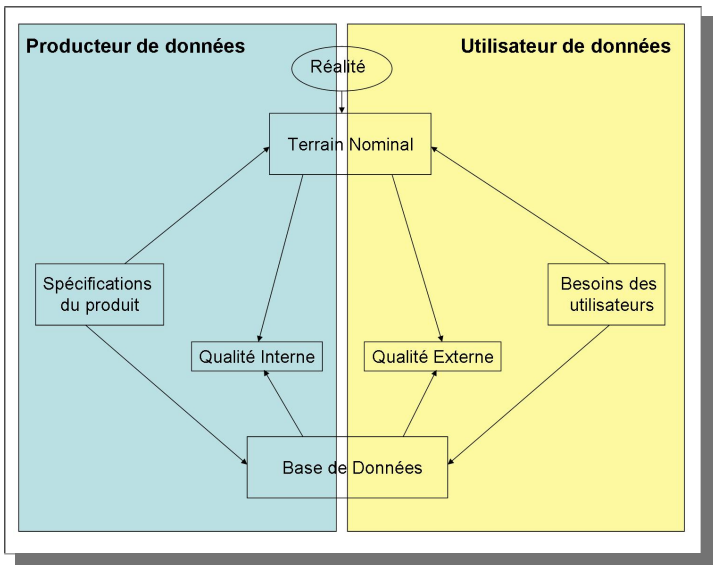

Figure 2.12 – Qualités interne et externe des jeux de données spatiaux

Cela nécessite de mettre en place des outils basés sur des indicateurs. Dans ce sens, [Moellering, 1987] a spécifié cinq critères pour définir la qualité interne d'un jeu de données spatial :

✓ **La généalogie** regroupe l'historique des données, les indications sur les sources, les opérations de saisie, les transformations effectuées sur les données [Clarke et Clark, 1995].

✓ **La précision géométrique** (ou exactitude spatiale) donne les écarts de position entre les objets de la base et ceux du monde réel. Des méthodes utilisées pour déterminer cette précision sont décrites dans [Faïz, 1996] et [Drummond, 1995].

- ✓ **La précision sémantique** (ou exactitude des attributs) est la différence entre la valeur d'un attribut du jeu de données et sa valeur dans le monde réel [Goodchild *et al.*, 1992] [Bicking, 1994].
- ✓ **L'exhaustivité** indique si les objets du monde réel sont tous représentés dans le jeu de données. [Brassel *et al.*, 1995] en distinguent deux types : l'exhaustivité des données et l'exhaustivité du modèle.
- ✓ **La cohérence logique** définit le degré de cohérence interne des données selon les règles de spécifications et de modélisation du jeu de données. Elle inclut la cohérence géométrique et la cohérence topologique des données spatiales. Plusieurs travaux concernant ces différents types de cohérence ont été abordés dans la littérature [Kainz, 1995], [Jeansoulin, 1997], [Ubeda, 1997] et [Puricelli, 2000].

Des composantes supplémentaires ont été ajoutées par d'autres auteurs :

- ✓ **L'actualité** (ou précision temporelle) qui détermine les dates de la dernière mise à jour et de la validité des données [Guptill, 1995].
- ✓ **La fidélité textuelle** qui est une mesure de l'exactitude de l'orthographe des informations écrites.
- ✓ **La cohérence sémantique** qui fait référence à la qualité avec laquelle les objets géographiques sont décrits [Salgé, 1995]

Tous ces critères ont été largement expérimentés et sont aujourd'hui utilisés dans de nombreux travaux de normalisation tels que ceux réalisés par l'organisation internationale de normalisation (ISO) ou le centre européen de normalisation (CEN). Ils forment un support pour évaluer la qualité d'un jeu de données géographiques. Cependant, si la qualité est jugée mauvaise, il faut pouvoir l'améliorer. Deux directions apparaissent dans la littérature pour y parvenir. Les travaux axés sur la recherche et la correction d'erreurs spatiales et plus particulièrement la vérification des contraintes géométriques et topologiques [Laurini et Raffort, 1994], [Ubeda et Servigne, 1996], [Ubeda, 1997], [Puricelli, 2000] et les travaux basés sur la superposition de cartes d'origines ou d'échelles différentes grâce à des techniques basées sur l'appariement [Chrisman, 1989], [Veregin, 1989] et [Matos. *et al.*, 1997].

La **pertinence des données** est une notion que l'on peut lier au concept **d'adéquation aux besoins** (fitness for use) et en particulier à la qualité dite externe [Dassonville *et al.*, 2002]. La notion d'adéquation aux besoins a été définie dans les années 70 par [Juran *et al.*, 1974]. [Chrisman, 1983] a ensuite mis en évidence la nécessité d'avoir une qualité qui sert de base à l'évaluation de l'adéquation de l'utilisation de l'information géographique pour un besoin précis.
[Wang et Strong, 1996] définissent alors quatre dimensions de la qualité externe des données spatiales :

- ✓ La qualité intrinsèque détermine la crédibilité, la précision, l'objectivité et la réputation que l'on peut accorder aux données.
- ✓ La qualité contextuelle s'attache plutôt à vérifier si les données sont appropriées (pertinence, valeur ajoutée) et suffisantes (complétude, volume de données) pour l'usage qui doit en être fait.
- ✓ La qualité représentationnelle aborde les notions d'interopérabilité et de compréhension des données.

✓ Enfin, le dernier critère concerne l'accessibilité et la sécurité liées aux données.

Par ailleurs, ces dernières années plusieurs travaux de recherche ont été entrepris pour mieux prendre en compte la qualité externe [Bruin *et al.*, 2001], [ReV !Gis, 2004] [Vasseur, 2004], [Devillers, 2004], [Devillers et Jeansoulin, 2005]. Deux grandes approches ont été proposées, l'une basée sur **l'évaluation du risque** encouru par l'utilisation de données non adéquates et l'autre sur **l'utilisation des métadonnées** pour analyser la similarité entre les données produites et les besoins des utilisateurs. Ces méthodes sont différentes dans leur élaboration mais aboutissent toutes deux à apprécier la qualité des données en fonction de l'utilisation qui doit en être faite. Elles permettent ainsi **des interprétations mieux ciblées et des prises de décision moins risquées**.

[Agumya et Hunter, 1998] utilisent une approche basée sur la gestion des risques pour évaluer a posteriori la qualité des données utilisées pour une certaine application. Ce processus est composé de plusieurs étapes permettant d'identifier, d'analyser, d'évaluer, de mesurer l'exposition et d'estimer le risque afin de pouvoir lui fournir une réponse. Les auteurs proposent ainsi une méthode permettant de déterminer le niveau acceptable de l'incertitude en analysant les risques potentiels dus à une prise de décision basée sur l'utilisation de données.

[Bruin *et al.*, 2001] proposent également une méthode basée sur une évaluation du risque. Leur approche utilise la technique des arbres de décision et le concept de valeur de contrôle pour estimer le coût qu'engendrerait une décision basée sur des données incorrectes. Cette méthode permet de sélectionner le jeu de données qui limitera le risque mais suppose de savoir quantifier le risque, ce qui peut s'avérer être une tâche difficile.

D'autres approches basées quant à elles sur les métadonnées ont été proposées. Selon [Frank, 1998], les descriptions actuelles de la qualité offertes par les métadonnées suivent les orientations du producteur de données et sont généralement fournies à des fins de traitement de données mais n'informent pas l'utilisateur sur l'utilisation qui peut en être faite. L'auteur suggère que la qualité soit indépendante du processus de production et utilisée dans une procédure formelle permettant d'obtenir des résultats quantitatifs de telle sorte à ce qu'elle soit opérationnelle et interprétable par l'utilisateur. Pour ce faire, il propose un métamodèle permettant de fusionner les points de vue du producteur et des utilisateurs afin d'analyser a priori la similarité entre les données produites et les besoins des utilisateurs.

[Hunter, 2001] précise quant à lui, que l'utilisateur voudrait avoir la possibilité d'utiliser techniquement l'information de qualité associée à l'information géographique et connaître le risque lié à une utilisation de données incertaines. Il soulève également un autre problème concernant la granularité des informations de qualité qui selon lui se trouve à un niveau très général, ne permettant pas à l'utilisateur d'obtenir une information utile. Les métadonnées fournissent des informations sur le jeu de données dans sa globalité mais il n'est pas possible d'obtenir l'information pour un sous-ensemble ou encore un objet particulier. L'auteur pense donc

que les travaux de recherche à venir doivent prendre en compte ces deux points c'est-à-dire l'évaluation de la qualité externe et une meilleure identification de la granularité dans les métadonnées de qualité.

[Devillers, 2004] part du constat qu'une mauvaise utilisation ou interprétation des données peut avoir des conséquences importantes lors des prises de décision. Pour réduire ce risque, les utilisateurs doivent être en mesure d'évaluer l'adéquation des données à leur utilisation c'est-à-dire leur qualité externe. Mais les techniques d'évaluation sont difficiles voire impossibles à utiliser pour des utilisateurs non experts. Pour y remédier, l'auteur propose une série d'outils permettant de structurer et de communiquer l'information de qualité à des utilisateurs experts ou à des experts en qualité afin que ceux-ci soient à même de conseiller des utilisateurs non experts dans leur utilisation des données. L'auteur utilise des indicateurs visuels pour représenter l'information de qualité en support à la prise de décision. Ces indicateurs fournissent des informations brutes ou agrégées sur la qualité des données géospatiales et sont présentés sur un tableau de bord faisant partie de l'interface du SIG.

Le projet européen ReV !Gis, terminé en juin 2004, est né du constat que lorsqu'on manipule des données géographiques, on ne peut ignorer les imprécisions de mesure, les ambiguïtés de définition et les erreurs d'observation ou d'interprétation [ReV !Gis, 2004]. La qualité apparaît donc comme l'écart entre le contenu de l'information archivée et l'information qui est souhaitée pour un usage mais cette définition est difficile à mettre en pratique. Ainsi, pour surmonter en partie ces difficultés, le développement de techniques de représentation et de raisonnement basées sur la révision des connaissances spatiales et leur intégration à des applications commerciales et scientifiques a été mis en œuvre dans le cadre de ce projet.

[Vasseur, 2004] a consacré sa thèse à la modélisation de la qualité dans les applications géographiques et s'est plus particulièrement intéressée à la qualité externe. Elle établit des relations entre l'application et les données afin d'évaluer la qualité externe de l'application. Elle s'appuie sur deux types d'ontologies définis par [Jeansoulin et Wilson, 2002] : les ontologies du problème et les ontologies du produit. **L'ontologie du produit** décrit une partie du monde réel telle que représentée par le producteur de données (spécifications, qualité interne, schéma des données), **l'ontologie de problème** décrit quant à elle l'interprétation de cette partie du monde réel par l'utilisateur en fonction du problème à résoudre (connaissance de l'utilisateur, prise de décision, qualité externe). L'auteur crée ensuite deux matrices (**matrice de qualité réelle** issue de l'ontologie du produit et **matrice de qualité attendue** issue de l'ontologie du problème) dans un référentiel commun [Brodeur *et al.*, 2003], puis par comparaison, agrégation et normalisation, définit la matrice de qualité de l'application qui lui permet d'évaluer l'adéquation aux besoins (Cf. figure 2.13). Si la qualité est jugée insuffisante alors un retour en arrière est effectué, soit par reformulation de l'ontologie du problème, soit par analyse des données pour obtenir une matrice de qualité améliorée. Lorsque la qualité est jugée satisfaisante, la matrice d'application permet de déterminer si le jeu de données est adéquat grâce au calcul de l'utilité. **L'utilité** est une mesure quantitative qui provient de la comparaison

entre les attentes de l'utilisateur et les données disponibles dans le jeu de données [Frank *et al.*, 2004]

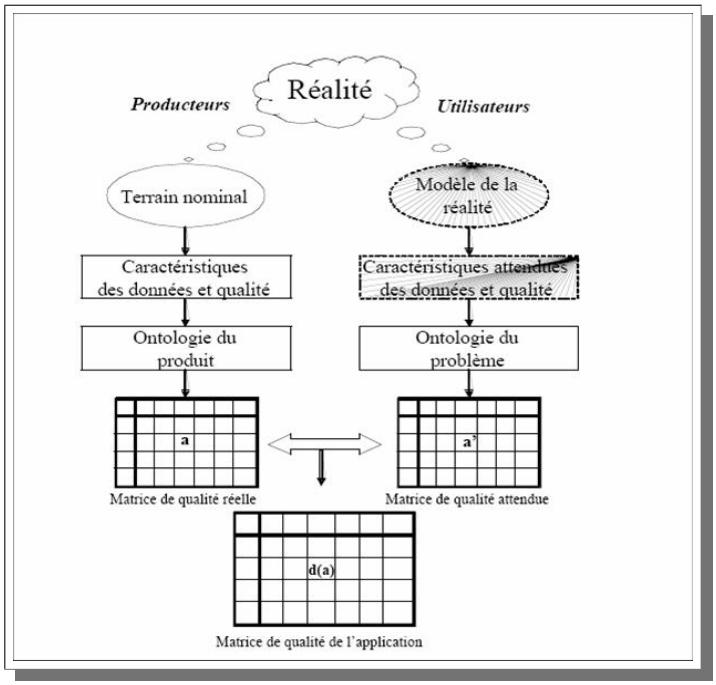

Figure 2.13 – Evaluation de la qualité externe selon [Vasseur, 2004]

L'utilité de la qualité dépend donc de l'utilisation qui va être faite de l'information qu'elle qualifie, en particulier lors de prises de décisions. La fonction d'utilité de la qualité se mesure donc par son pouvoir de réduction de l'incertitude dans un processus de décision [Devillers et Jeansoulin, 2005]. [Harding, 2005] signale par ailleurs que l'information géographique étant souvent utilisée à des fins de résolution de problèmes ou de prise de décision, la fiabilité des résultats est par conséquent en partie dépendante de l'utilisation prévue des données ainsi que de leur interopérabilité avec d'autres sources de données.

Finalement nous pouvons assurer à la vue de ces travaux que **l'évaluation de la qualité** d'un jeu de données géographiques **est une chose délicate** qu'il ne faut pas à prendre à la légère tant une base de données de mauvaise qualité peut provoquer de nombreuses erreurs et incompréhensions qui peuvent mettre en péril la cohérence des données et des systèmes dans l'infrastructure. Un autre aspect qui nous paraît essentiel en conclusion de ce paragraphe est qu'**il ne faut pas négliger le point de vue de l'utilisateur** lors de l'évaluation de la qualité d'un jeu de données et par conséquent considérer la qualité externe des données spatiales. Cela est d'autant plus vrai dans un contexte de prise de décision où les données servent de support à l'action des acteurs.

Sources d'erreur et incohérence des données

L'information géographique représente un modèle de la réalité [Longley *et al.*, 2005]. Cependant, comme le souligne [Box, 1976], tous les modèles sont faux même si certains sont utiles. Cela a pour effet de dénaturer l'information fournie par rapport à ce qu'elle est censée représenter, provoquant ainsi des erreurs, des imprécisions, des ambiguïtés et finalement des incohérences dans le jeu de données.

[Sheeren, 2005] pense que le manque de cohérence entre les données provient principalement de deux faits : une différence dans **l'actualité** des bases de données et l'existence **d'erreurs** de saisie. Il souligne en particulier que la politique de maintenance des jeux de données n'est pas organisée à l'identique et que le rythme des mises à jour est souvent différent, ce qui provoque des **incohérences** entre les jeux de données.

Selon [Heuvelink, 2005], l'erreur peut être définie par la **différence** entre une réalité et la représentation de cette réalité. L'auteur précise que les erreurs proviennent essentiellement de trois facteurs : le processus de modélisation des données (par exemple les algorithmes qui sont utilisés), les moyens technologiques (comme la précision des appareils de mesure) et le facteur humain (principalement lors de la saisie des données). En conséquence, l'ensemble des erreurs influence directement la qualité interne des données produites.

Selon [Devogele, 1997], la modélisation du monde réel et de ses phénomènes n'étant pas la même d'un concepteur à l'autre, des différences (appelées aussi conflits d'intégration) interviennent dans la structure et la population des classes de deux bases de données. [Kadri-Dahmani, 2005] ajoute en ce sens, que les conflits perturbent la **cohérence** de la base de données géographiques et pénalisent les analyses spatiales, privant de ce fait l'utilisateur qui obtient finalement une **information non souhaitée voire erronée**.

Pour limiter les problèmes de cohérence, certains auteurs de la littérature ont effectué un recensement des différents types de **conflits** que l'on retrouve généralement lors de l'intégration des données [Devogele, 1997], [Benslimane *et al.*, 1999], [Defude, 2005]. Ces inventaires ont conduit à la définition de taxonomies plus ou moins détaillées.

Ainsi, [Devogele, 1997] a recensé un grand nombre de conflits et a proposé une classification très complète. Il indique en outre le type et l'origine des conflits, les causes et les effets qu'ils peuvent provoquer et fournit quelques solutions permettant de les résoudre. [Devogele, 1997] regroupe les conflits en six classes :

- Les conflits liés aux différentes sources de données utilisées pour constituer les bases de données géographiques : levés de terrain, photographies aériennes, images satellitaires, cartes scannées, ... [Kavouras *et al.*, 1995].
- Les conflits d'hétérogénéité qui sont eux-mêmes regroupés en six catégories :
  – Les conflits de modèles : modèles issus de la CAO, modèles hy-

brides, modèles relationnels étendus ou modèles objets [Rouet, 1991], [Querzola et Billout, 1995].
  - Les conflits de positionnements : ellipsoïde, système de projection, point de référence différents [Shepherd, 1992], [Laurini, 1996], [Devogele, 1997].
  - Les conflits de gestion de modélisation de la troisième dimension : modélisation de l'altitude en 2.5D ou en 3D ou abstraction différente d'un même phénomène du monde réel.
  - Les conflits de représentation de la géométrie, en particulier le choix entre le mode raster et le mode vecteur.
  - Les conflits de métadonnées géométriques qui regroupent les notions de précision (unités de mesure différentes), résolution (critères différents) et exactitude (processus de saisie différents) [Shepherd, 1992].
  - Les conflits de modélisation de la topologie : modèle spaghetti ou topologique.
- Les conflits de classification (regroupement et résolution différents), de spécification (sélection et décomposition des objets) et de fragmentation (segmentation et granularité des bases de données) qui portent sur la définition des classes et des relations et sur leurs instanciations.
- Les conflits liés à la structure utilisée pour représenter les éléments.
- Les conflits portant sur la description sémantique et géométrique des éléments [Larson *et al.*, 1989], [Kim *et al.*, 1993].
- Les conflits liés au données, provoqués par certains des conflits évoqués ci dessus ou par des opérations de généralisation.

[Benslimane *et al.*, 1999] ont proposé une taxonomie des conflits de données qui se décompose selon trois niveaux : les conflits de modèle, les conflits structurels et les conflits sémantiques.

[Defude, 2005] s'inspire quant à lui, de la classification définie par [Devogele, 1997] et propose une taxonomie simplifiée à quatre niveaux : les conflits de modélisation, les conflits de schémas, les conflits de métadonnées et les conflits de données.

Nous nous appuyons sur la taxonomie de [Devogele, 1997] pour synthétiser dans le tableau 2.2 les différents types de conflits, leurs effets et les solutions proposées pour les contourner.

Type de conflit	**Effets**	**Solution proposée**
Conflits de sources	Engendrent d'autres conflits tels que des conflits de gestion de la troisième dimension, des conflits de résolution et de précision ou encore des conflits d'intégration des données	Compléter la base de données avec des métadonnées et traiter les conflits lors de la résolution des conflits qu'ils engendrent.
Conflits de modèles	Impossibilité d'intégrer les données directement dans la base de données sans transformation des modèles	Définir un modèle commun dans lequel les modèles en conflit sont convertis [Shepherd, 1992]

Type de conflit	Effets	Solution proposée
Conflits de positionnement	Impossibilité d'intégrer les données directement dans la base de données sans transformation des modèles de positionnement	Transferer les données d'un système à un autre grâce à des outils de recalage [Fagan et Soehngen, 1987], [Rouet, 1991]
Conflits modélisation de la troisième dimension	Impossibilité d'intégrer les données directement dans la base de données sans transformation des modèles d'altitude ou d'abstraction	Compléter la base de données avec des métadonnées décrivant l'abstraction utilisée [de Cambray, 1994]
Conflits de mode de représentation de la géométrie	Impossibilité d'intégrer les données directement dans les bases de données	– Utiliser un algorithme de conversion raster/vecteur [Peuquet, 1981] – Conserver les deux modes pour le même objet [Günter, 1989] – Définir un modèle canonique englobant les deux modes [Ramirez, 1997], [Egenhofer *et al.*, 1989]
Conflits de métadonnées géométriques	Engendrent des conflits de données	Conflits à traiter lors de la résolution des conflits qu'ils entrainent
Conflits de la topologie	Impossibilité d'intégrer les données directement dans la base de données sans transformation des modèles topologiques	– Ajout de relations topologiques dans la base la moins riche [Ubeda, 1997] – Developpement de modèles génériques où des données avec des topologies différentes peuvent être stockées [David, 1991] – Developpement d'un modèle permettant d'exprimer la topologie selon différentes résolutions [Bertolotto *et al.*, 1994], [Puppo et Dettori, 1995]
Conflits de classification	Impossibilité d'intégrer les données dans la base de données sans tranformation préalable	Fusionner, généraliser ou partitionner les classes [Larson *et al.*, 1989], [Gotthard *et al.*, 1992], [Dupont, 1995]
Conflits de spécification	Engendrent des conflits de données, de classification et de fragmentation	Conflits à traiter lors de la résolution des conflits qu'ils entrainent

Type de conflit	**Effets**	**Solution proposée**
Conflits de fragmentation	Impossibilité d'intégrer les données dans la base de données sans tranformation préalable	– Faire une segmentation dynamique [Maguire *et al.*, 1992] – Pas de solution pour les conflits de granularité – Créer des relations entre les objets en conflits de décomposition [Dupont, 1995]
Conflits de structure	Pas de correspondance explicite entre les éléments	Choisir parmi les structures en conflit celle qui est la moins contrainte [Spaccapietra et Parent, 1991]
Conflits de description sémantique et géométrique	Impossibilité de déterminer des équivalences entre les objets en conflits	– Pour les conflits sémantiques : – Déclarer des fonctions de correspondances [Larson *et al.*, 1989] – Définir des attributs virtuels [Dupont, 1995] – Ajouter des relations entre les objets – Pour les conflits géométriques : – Définir des métadonnées permettant de connaître la dimension de la géométrie [Stephan *et al.*, 1993] – Définir une structure permettantde relier les géométries à différentes échelles [Puppo et Dettori, 1995], [Timpf et Frank, 1995]
Conflits de données	Impossibilité de retrouver les objets en correpondances	Méthodes d'appariement [Lemarié, 1996]

Table 2.2 – Taxonomie des conflits proposée par [Devogele, 1997]

Ces différentes taxonomies montrent qu'une quantité non négligeable de conflits très diverses peut se produire lors de l'intégration des données dans un jeu de données géographiques. Par ailleurs, certains conflits sont spécifiques au caractère spatial des données et viennent s'ajouter aux conflits classiques que nous rencontrons habituellement dans les bases de données. **Leur traitement** nécessite la mise en place de **méthodes spécialisées** prenant en compte la géométrie et la topologie des objets géographiques. **La gestion de ces conflits** est donc une étape importante du processus d'intégration qu'il ne faut pas sous-estimer, d'autant plus dans un contexte de mise à jour où les évolutions proviennent de sources distinctes.
Plus précisément, **la résolution des conflits** d'intégration permet de gérer la **cohérence interne** d'un jeu de données. En effet, comme le précise [Defude, 2005], pour pouvoir intégrer correctement des données (et donc maintenir la cohérence de la base de données), il faut être capable de détecter les conflits résultant de l'hétérogénéité des données.

Le **contrôle de cohérence** a été largement étudié dans le domaine des bases de données classiques [Gardarin, 1999], [Gançarski, 1994], [Doucet *et al.*, 1996], [Medeiros et Jomier, 1994] et a fait l'objet de quelques recherches significatives dans les bases de données géographiques que nous présentons maintenant.

[Braun, 2003] voit la vérification de la cohérence comme la justification que les données ne produisent pas une vue aberrante du monde réel. Cela implique de spécifier **un ensemble de contraintes** qui doivent assurer la cohérence interne des données selon des règles de modélisation et les règles inhérentes à la spécification du jeu de données. Il distingue quatre types de contraintes :

- Les contraintes de cohérence sémantique qui concernent les attributs non géométriques.
- Les contraintes de cohérence spatiale qui vérifient la cohérence logique de la base du point de vue géométrique (par exemple, une route ne traverse pas une rivière sauf si elle passe sur un pont).
- Les contraintes d'intégrité topologique qui contrôlent la cohérence topologique de la base.
- Les contraintes d'intégrité référentielle qui examinent la cohérence logique de la base.

Ubeda [Ubeda, 1997] définit la cohérence spatiale des données comme étant le respect **d'un ensemble de règles spatiales** qui s'appliquent aux données géographiques. Ces règles peuvent être conceptuelles (respect du modèle de données), géométriques (respect des règles mathématiques de définition de la forme des objets) ou sémantiques (respect des spécifications).
Il s'appuie ensuite sur une description des entités géographiques pour affiner cette définition et propose alors trois types de cohérence :

- La cohérence structurelle qui définit l'adéquation entre les structures de

données de stockage et le modèle conceptuel spatial de données.
- La cohérence géométrique qui spécifie l'adéquation entre le modèle conceptuel spatial des données et les modèles mathématiques et logiques du monde réel
- La cohérence topo-sémantique qui caractérise l'adéquation entre les relations spatiales des objets géographiques de la base et leur sémantique.

Puricelli [Puricelli, 2000] reprend les travaux de [Ubeda, 1997] et ajoute deux autres types de cohérence permettant ainsi de maintenir une cohérence plus large, notamment entre différentes bases de données :
- La cohérence inter-couches qui est la cohérence topologique entre plusieurs objets appartenant à différentes couches d'une base.
- La cohérence inter-bases qui est la cohérence topologique entre plusieurs objets appartenant à des bases différentes.

Ces travaux sont intéressants car il définissent précisément la cohérence des bases de données géographiques mais ont le plus souvent été initiés pour juger la qualité des bases de données géographiques et non comme une étape essentielle du processus de mise à jour. [Kadri-Dahmani, 2005] propose donc une approche à base de contraintes d'intégrité pour enrichir les SIG de la notion d'évolution afin d'automatiser au maximum l'opération de mise à jour en garantissant l'intégrité de ses données. Cette approche fournit quelques avancées à la gestion de la cohérence lors de la mise à jour d'un jeu de données géographiques mais se place d'une part du point de vue du producteur de données qui veut réviser et maintenir ses jeux de données à moindre coût et d'autre part suppose la réorganisation des bases de données afin que la notion d'évolution soit prise en compte dès leur conception. Elle ne prend donc pas en considération **le caractère distribué** des jeux de données qui entraîne **la réplication des données** dans un contexte de **mise à jour simultanée**, qui est le coeur de notre problème.

2.2 Gestion des données réparties et réplication

Le besoin croissant des applications à accéder simultanément à des données réparties sur plusieurs sites a conduit la communauté informatique à s'intéresser à l'étude de la **réplication des données**. En vulgarisant, la réplication consiste à « stocker » plusieurs copies d'une donnée sur des sites distincts de l'environnement réparti afin qu'elle soit plus facilement accessible.
L'utilisation de la réplication comporte plusieurs avantages, notamment celui de rendre disponibles les données lorsque la donnée de référence n'est pas libre (probabilité de panne plus faible), ou encore de favoriser le partage et le parallélisme (équilibrage des charges, meilleur temps de réponse, diminution du temps de transfert des données) et d'améliorer de ce fait les performances d'accès aux données réparties. En contrepartie, la réplication implique une gestion plus complexe des mises à jour et nécessite une vérification de la cohérence entre les données répliquées et la donnée de référence (échange de messages entre les différents sites).

Il existe deux catégories de protocoles de réplication :

- Les **protocoles à cohérence forte** (ou synchrones) garantissent la cohérence des copies à tout moment et sur tous les réplicats [Bernstein *et al.*, 1987]. On utilise un **protocole synchrone** lorsque l'on veut que toutes les données réparties aient à chaque instant la même valeur. Cela implique de nombreux échanges entre les répliques et nécessite l'utilisation de mécanismes de verrouillage [Dedieu, 2002].
- Les **protocoles à cohérence faible** (ou asynchrones) ne cherchent pas à assurer la cohérence immédiate des copies et permettent la divergence entre les répliques. On utilise un **protocole asynchrone** lorsque l'on veut pouvoir, à tout instant, accéder, créer, modifier ou encore supprimer une donnée, quel que soit le site sur lequel elle se trouve.
  Les protocoles de réplication à cohérence faible sont de deux types :
  - Les **protocoles pessimistes** permettent la divergence « ponctuelle » des copies en acceptant que les données soient différentes pendant un temps limité. Les données peuvent être modifiées localement mais un contrôle des incohérences est effectué a priori avant toute intégration des mises à jour sur les autres répliques. On utilise pour cela des contraintes de communication entre les sites (protocole de copie primaire [Kronenberg *et al.*, 1986], [Oracle, 1996], [Dietterich, 1994] ou à base de quorum [Helal *et al.*, 1996]). Ils donnent alors l'illusion à l'utilisateur qu'il n'existe qu'une seule copie [Herlihy et Wing, 1990], [Bernstein et Andgoodman, 1983], [Bernstein *et al.*, 1987] mais ils sont difficilement utilisables en mode déconnecté.
  - Les **protocoles optimistes** permettent également la divergence des copies et n'imposent pas de contraintes entre les sites [Demers *et al.*, 1994], [Kermarrec *et al.*, 2001], [Saito et Shapiro, 2005], [Oster, 2005]. Chaque réplique peut mettre à jour librement sa copie, les mises à jour sont ensuite envoyées aux autres répliques qui les intègrent directement. Dans ce cas, le **contrôle de la cohérence** est effectué a posteriori sur chacune des répliques. Des algorithmes de **détection de conflits** et de **réconciliation** des écritures divergentes doivent être mis en place afin de corriger les incohérences entre les répliques. Idéalement les protocoles asynchrones optimistes doivent assurer la **cohérence des répliques à terme** et la **convergence**. Cela suppose que lorsque le système est au repos, toutes les données répliquées doivent converger vers une même valeur et ne pas remettre en cause la cohérence des copies. Nous verrons dans la suite de ce paragraphe que la réconciliation des données conflictuelles est un des points critiques des travaux de recherche en réplication optimiste.

Le choix entre ces différents protocoles dépend avant tout des besoins et objectifs de l'application qui doit être mise en place. De surcroît, d'autres caractéristiques sont également à prendre en compte lors de la définition d'un protocole de réplication [Saito et Shapiro, 2005] :

- Quel est le **type** des objets répliqués ?
- La réplication des données est elle **totale** ou **partielle** ?
- Quelle est la **topologie du réseau** (en anneau, en étoile, ad-hoc, ...) ?

- Quelles sont les **opérations** permises (insertion, suppression, modification, requête, ...) ?
- Existe t'il un **journal** des écritures ?
- Comment est gérée la mise à jour des répliques ?
  - **Mono-maître** lorsque la mise à jour d'une donnée est effectuée par un site de référence qui diffuse ensuite les évolutions aux autres sites.
  - **Multi-maîtres** lorsque chaque site peut mettre à jour sa copie localement et diffuser ensuite ses évolutions aux autres sites.
- Comment est déclenchée **la propagation** des évolutions (à la demande, dès que possible, périodiquement, ou est elle propre à l'application) ?
- Selon quel mode est effectuée la **diffusion** des évolutions ?
  - En mode push lorsque les mises à jour sont envoyées par le site où elles ont été effectuées.
  - En mode pull lorsque les mises à jour sont demandées par les autres sites.
  - En mode mixte c'est-à-dire l'un ou l'autre mode indifféremment.
- Comment s'effectue le **transfert** des évolutions (par état courant ou par opérations) ?

Dans la suite de ce paragraphe, nous abordons en premier lieu les études menées dans les systèmes de gestion de base de données réparties pour gérer la réplication. Puis, nous présentons un aperçu des différents travaux en matière de réplication optimiste des données. Enfin, nous discutons des solutions mises en place en information géographique, en particulier des possibilités qu'elles offrent pour la gestion des accès concurrents et des transactions longues.

2.2.1 Réplication dans les SGBD

Une **base de données** (BD) est une collection cohérente de données structurées. Une entité de base de données est une paire (nom, valeur) sur laquelle des **opérations de lecture et d'écriture** peuvent être appliquées. Un **système de gestion de base de données** (SGBD) est un ensemble de logiciels permettant de gérer et manipuler de manière efficace une base de données. Il assure la structuration, la maintenance, la mise à jour et la consultation des données [Gardarin, 1999]. Une base de données est dite **cohérente** si elle satisfait un certain nombre de contraintes. Une **transaction** est une unité de travail définie par [Gray, 1980] regroupant une suite d'opérations qui doivent être exécutées sur la base de données. Une transaction est considérée comme **valide** par le SGBD si elle ne remet pas en cause la cohérence de la base de données. Pour être valide, une transaction doit présenter les propriétés ACID c'est à dire :

- **Atomicité** : Elle assure qu'en cas de succès, toutes les opérations sont validées et qu'en cas d'échec aucune opération n'est appliquée.
- **Cohérence** : La transaction ne doit pas remettre en cause la cohérence de la base de données.
- **Isolation** : Une transaction n'a pas d'effet visible tant qu'elle n'a pas été validée.
- **Durabilité** : Une transaction validée ne peut pas être remise en cause. Les

conséquences sur la base sont persistantes et ne peuvent pas être supprimées. L'application d'une transaction valide modifie de ce fait l'état de la base de données.

Une **base de données répartie** est constituée d'un ensemble de bases de données logiquement reliées et distribuées sur un réseau [Arcangeli *et al.*, 2004]. Un **SGBD réparti** permet aux utilisateurs de gérer une base de données répartie de manière transparente comme s'ils avaient accès à une seule base de données [Özsu et Valduriez, 1999]. Chaque site contient alors soit une copie de la base de données répartie (réplication totale) ou une partie de la base de données répartie (réplication partielle).
Dans un SGBD réparti, la mise à jour de la base de données répartie s'effectue soit selon une approche dite de **copie-primaire** (la mise à jour est centralisée sur un site puis propagée aux autres sites), ou selon une **approche distribuée** (la mise à jour peut être effectuée sur n'importe quel site). L'utilisation d'une copie primaire permet d'éviter les transactions concurrentes mais pose le problème de la tolérance aux pannes. Autoriser la mise à jour sur tous les sites permet d'obtenir l'information de mise à jour plus rapidement mais nécessite un ordonnancement des transactions provenant des différentes copies. Par ailleurs, la propagation des mises à jour est gérée par le SGBD réparti et peut être effectuée de manière synchrone ou asynchrone.

La **propagation synchrone** (cf. figure 2.14) consiste à valider une transaction après l'avoir propagée aux autres répliques. Cette approche assure la cohérence des données de manière simple et garantit la propriété d'atomicité [Gray, 1978] mais le retardement de la validation des transactions provoque un surcoût des temps de réponse et engendre de ce fait des problèmes de performance et de disponibilité des données.

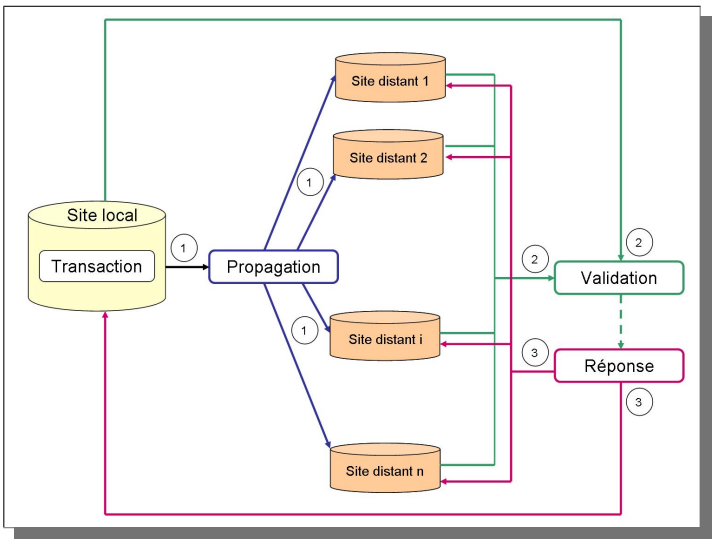

Figure 2.14 – Propagation synchrone des transactions dans un SGBD réparti

La **propagation asynchrone** (cf. figure 2.15) consiste à valider localement la transaction, avant de la propager aux autres répliques [Bernstein et Newcomer, 1997], [Breitbart et Korth, 1997]. Cette validation locale permet aux utilisateurs de pou-

voir continuer à travailler en mode déconnecté [Bernard *et al.*, 2003]. Une validation définitive est alors effectuée à la reconnexion lors d'une étape de **synchronisation**. Cependant, les opérations effectuées sur le site local pendant la déconnexion peuvent être en conflit avec d'autres opérations qui ont déjà été validées par le SGBD réparti, il faut alors vérifier la **cohérence mutuelle** des bases de données. La cohérence mutuelle est définie dans [LePape, 2005] comme étant le fait qu'à un instant donné, toutes les copies d'une donnée ont la même valeur.
Un autre inconvénient de la propagation asynchrone est que la propriété d'atomicité ne peut être garantie [Kemme, 2000]. En effet, si une panne se produit avant qu'une transaction validée localement puisse être propagée aux autres sites alors cette transaction est perdue.

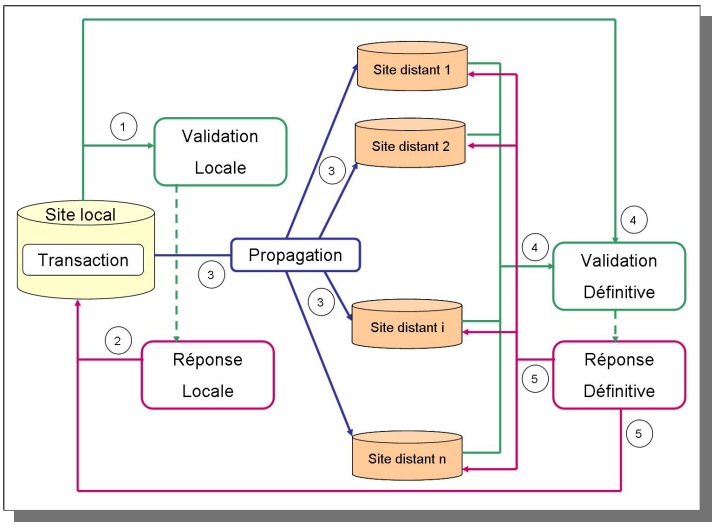

Figure 2.15 – Propagation asynchrone des transactions dans un SGBD réparti

Une **base de données répliquée** est composée d'un ensemble de répliques stockées sur différents noeuds d'un SGBD réparti, chaque réplique étant la copie d'une **donnée de référence** [Pacitti *et al.*, 1999]. La réplication des données dans un SGBD réparti est un problème qui soulève encore beaucoup d'interrogations dans la communauté scientifique et qui fait toujours l'objet de nombreux travaux de recherches [Gançarski, 2006], [LePape, 2005], [Kemme, 2000], [Monnet, 2006], notamment en ce qui concerne la qualité des bases de données répliquées.
Selon [Pacitti et Valduriez, 1998], la qualité d'une base de données répliquée dépend avant tout de la fraîcheur des données et de la cohérence. [Pacitti *et al.*, 1999] proposent deux stratégies de propagation des mises à jour (propagation des mises à jour retardée et propagation des mises à jour immédiate) permettant d'améliorer la fraîcheur des données, mais l'étude est basée sur un système de réplication mono-maître. [Gançarski, 2006] élargit la notion de qualité en spécifiant deux objectifs à atteindre : la cohérence et la performance. La performance est ici liée à la latence, le débit transactionnel, la disponibilité des mises à jour, ou encore la tolérance aux pannes.
La qualité des bases de données répliquées est donc étroitement liée à la cohérence. Par cohérence on sous-entend, cohérence des données répliquées par rapport

aux données de référence [Pacitti *et al.*, 2001]. Cependant, comme le souligne [Gançarski *et al.*, 2002], la notion de cohérence est subjective et dépend du point de vue, de la granularité du point de vue, de l'application ou encore des besoins particuliers. Une taxonomie des différentes problématiques de la cohérence a cependant été proposée dans [Ramamritham et Chrysanthis, 1996]. Plus récemment, une analyse des travaux en matière de divergence des copies lors de la réplication des bases de données a été fournie dans [LePape, 2005].

Du côté des systèmes commerciaux, les méthodes de réplication optimiste ont également inspirées les concepteurs de base de données relationnelles car ils sont aussi confrontés au problème de l'échange des données sur des systèmes répartis. Néanmoins, la réplication optimiste multimaîtres ne constitue pas le premier choix des concepteurs de bases de données relationnelles car elle s'avère difficile à mettre en œuvre. Les éditeurs d'Oracle [Oracle®, 2003] recommandent d'ailleurs d'utiliser un site de référence pour résoudre les conflits et de n'employer la réplication optimiste que si nécessaire.
Dans la suite de ce paragraphe nous présentons quelques solutions proposées par les éditeurs de SGBD en matière de gestion des données répliquées. Pour chaque SGBD, nous donnons une description des outils mis en place, exposons le type de collaboration utilisé et soulignons le degré d'automatisation des techniques de détection et de résolution de conflits.

Oracle [Oracle®, 2003] a prévu des outils permettant la réplication d'un ensemble de tables sur plusieurs serveurs. La collaboration dans Oracle est asynchrone multimaîtres. Oracle détecte automatiquement les mises à jour concurrentes, propose des routines de résolution de conflits et renvoie une erreur en cas d'échec. Seuls les conflits portant sur les données d'une même table sont détectés, il n'y a pas de vérification au niveau des relations entre tables. Les méthodes prédéfinies pour résoudre les conflits d'unicité permettent uniquement d'échapper à la contrainte afin de pouvoir appliquer la mise à jour, elles ne garantissent de fait pas la convergence des données. Aucune procédure de résolution n'est définie pour les conflits de suppression du fait de la non journalisation des écritures.

PostgreSQL gère le contrôle de concurrence multiversions et la journalisation des écritures. Initialement la réplication n'est pas prévue mais des travaux permettant l'ajout d'un module ont depuis été entrepris. eRServer (the enterprise Replication Server project) fut le premier projet de serveur de réplication asynchrone dédié à PostgreSQL. Ce projet est maintenant abandonné en faveur de Slony1 qui est un serveur de réplication asynchrone asymétrique basé sur un réseau de nœuds et sur un unique site maître. Slony2, serveur de réplication synchrone multimaîtres est actuellement en cours de développement. Aucun module de collaboration asynchrone multimaîtres n'a été élaboré mais PostgreSQL étant un logiciel Open Source, il semble envisageable de le concevoir.

La réplication dans MySQL [MySQL, 2003] est prise en compte en interne. Tous les objets d'une base de données sont répliqués. La topologie est constituée d'un serveur maître et d'esclaves qui peuvent éventuellement travailler en mode déconnecté.

Le maître garde un journal de toutes les écritures effectuées sur les répliques depuis un point fixe dans le temps. La gestion des erreurs est laissée à la charge de l'utilisateur. La collaboration dans MySql est donc asynchrone asymétrique. MySql étant lui aussi OpenSource, des routines de résolutions de conflits peuvent être implémentées.

SQL Server [Seshadri et Garrett, 2000] gère également la réplication des données (tables, index, vues ...). Différents rôles sont attribués aux serveurs : L'éditeur (ou serveur de publication) contient les données répliquées et les met à la disposition de la réplication vers les autres serveurs. Le distributeur héberge la base de données et stocke l'historique des données (il peut être couplé à l'éditeur). Enfin, les abonnés reçoivent les données répliquées, les utilisent et propagent les modifications soit au distributeur, soit aux autres abonnés. La synchronisation s'effectue indifféremment en mode push (déclenchée par l'éditeur) ou pull (déclenchée par les abonnés). SQL Server détecte les conflits côté abonnés mais ne gère pas leur résolution. Les informations de conflit sont passées au serveur de publication afin que les conflits soient résolus au cours de la prochaine synchronisation. Pour résoudre les conflits, les résolveurs peuvent utiliser la source de la modification des données, ou bien la valeur de priorité du serveur de publication. La collaboration dans SQL Server est donc asynchrone multimaîtres.

Le tableau 2.4 présente une synthèse des principales caractéristiques de ces protocoles de réplication.

Protocole	**Type de collaboration**	**Opérations**	**Journalisation**	**Gestion des conflits**
Oracle	Asynchrone monomaître ou multimaîtres en fonction de la configuration préétablie	Mise à jour, Insertion, Suppression	Non	Automatique pour les conflits de mise à jour, manuel sinon
SQL Server	Asynchrone Multimaîtres	Mise à jour, Insertion, Suppression	Oui	Automatique mais la résolution s'effectue uniquement sur le serveur de publication.
MySQL	Asynchrone Monomaître	Mise à jour, Insertion, Suppression	Oui	Laissée à la charge de l'utilisateur
PostgreSQl + Slony1	Asynchrone Monomaître	Mise à jour, Insertion, Suppression	Oui	Laissée à la charge de l'utilisateur

Protocole	Type de collaboration	Opérations	Journalisation	Gestion des conflits
PostgreSQl + Slony2	Synchrone Multimaîtres	Mise à jour, Insertion, Suppression	Oui	Conflits résolus par l'utilisateur ou par des règles prédéfinies

Table 2.4 – Synthèse des travaux de réplication dans les bases de données

La gestion de la réplication dans les bases de données est devenue nécessaire avec la propagation d'ordinateurs portables sur lesquels les utilisateurs doivent pouvoir accéder aux données en mode déconnecté et avec la création de réseaux d'entreprises où chaque employé doit avoir une visibilité des données stockées. Cependant, comme le souligne [Gançarski, 2006], le problème le plus important reste **la question de la cohérence mutuelle des répliques en cas de mise à jour**. Les travaux cités ci-dessus montrent d'ailleurs bien la difficulté qu'ont les éditeurs de SGBDR à concevoir une réelle application de réplication optimiste asynchrone multimaîtres. En effet, contrairement aux applications généralistes, les transactions dans les bases de données sont fréquentes et les écritures conflictuelles ne sont pas rares. La difficulté réside donc dans le **maintien de la cohérence** des bases de données et notamment dans la **résolution des conflits**, souvent laissée **à la charge des utilisateurs** qui ne sont pas forcément qualifiés pour effectuer cela.

2.2.2 Réplication optimiste asynchrone

Nous présentons dans cette partie quelques travaux effectués dans le domaine particulier de la réplication des données asynchrone, optimiste et multimaîtres.

Le recours à des méthodes de réplication optimiste s'est dans un premier temps fait sentir pour couvrir des besoins particuliers en matière d'échange de données réparties, notamment concernant l'échange de messages entre plusieurs utilisateurs (groupware [Kawell *et al.*, 1988], Usenet [Lidl *et al.*, 1994]), la gestion des systèmes de fichiers répartis (Coda [Kumar et Satyanarayanan, 1993], Ficus [Reiher *et al.*, 1994], Roam [Ratner, 1998]) ou encore le travail collaboratif (Lotus-Note [Ellis *et al.*, 1991], CVS [Berliner, 1990], Clearcase [Allen *et al.*, 1995]). Des protocoles plus généralistes ont ensuite été proposés pour couvrir des besoins plus larges (Bayou [Demers *et al.*, 1994], IceCube [Kermarrec *et al.*, 2001]).
Nous détaillons ici quelques unes de ces méthodes en précisant pour chacune le besoin cible, le type de collaboration mis en place et le degré d'automatisation des techniques de détection des conflits et de réconciliation.

Le protocole Usenet assure la gestion des messages envoyés par des utilisateurs répartis sur des sites distincts et ponctuellement déconnectés [Lidl *et al.*, 1994], [Spencer et Lawrence, 1998], [Saito *et al.*, 1998]. La collaboration dans Usenet est

asynchrone multimaîtres basée sur l'échange de messages. Il n'y a pas à proprement parler d'écritures conflictuelles car la règle de Thomas [Thomas, 1979] spécifiant que le dernier écrivain gagne est utilisée. Usenet ne permet pas l'utilisation d'opérations de modification et se révèle être au final un protocole ni optimiste, ni pessimiste.

La mobilité et la décentralisation des services ont accru le besoin d'utiliser la réplication optimiste notamment pour gérer les systèmes de fichiers répartis (Coda [Kumar et Satyanarayanan, 1993], Ficus [Reiher *et al.*, 1994] ou Roam [Ratner, 1998]) ou encore en matière de travail collaboratif (Lotus Note [Kawell *et al.*, 1988], [Ellis *et al.*, 1991], [Moore, 1995], CVS [Berliner, 1990] [Cederqvist *et al.*, 2001], Clearcase [Allen *et al.*, 1995]).
Coda [Kistler et Satyanarayanan, 1992] est un système de fichiers mobile destiné aux ordinateurs portables [Mummert *et al.*, 1995], [Kumar et Satyanarayanan, 1993]. La collaboration dans Coda est asynchrone multimaîtres et utilise un modèle de cohérence à terme (lorsque le système est au repos, les données sont cohérentes entre elles). La détection des conflits est automatique et la résolution spécifique à l'application. Le modèle client/serveur sur lequel repose Coda nécessite de définir à l'avance qui sera client et qui sera serveur. La synchronisation client/client n'est pas possible.
Ficus [Reiher *et al.*, 1994] et ROAM [Ratner, 1998] sont également des systèmes de fichiers répliqués destinés à des machines ponctuellement connectées. La collaboration dans Ficus et Roam est asynchrone multimaîtres et utilise un modèle de cohérence à terme. La détection des conflits est automatique et la résolution dépend de l'application et du type d'objets répliqués. La synchronisation par états courants ne permet pas de connaître les écritures qui ont fait diverger les répliques. La non journalisation des écritures pose des problèmes avec les créations/destructions car ne permet pas de savoir quelle politique de résolution appliquer lorsqu'un fichier est présent sur une réplique et absent sur une autre.

Lotus Note [Kawell *et al.*, 1988], [Ellis *et al.*, 1991], [Moore, 1995] est un logiciel de groupe de travail permettant le partage de l'information sur un support numérique à un groupe engagé dans un travail collaboratif. La collaboration dans Lotus Note est asynchrone multimaîtres. La détection des conflits est automatique et la résolution est spécifique à l'application ou laissée à la charge de l'utilisateur. La procédure de résolution de conflits automatique ne peut être modifiée par l'utilisateur si celui-ci souhaite appliquer une autre politique. Lotus est un système propriétaire et fermé.
CVS [Cederqvist *et al.*, 2001], [Vesperman, 2003] est un logiciel de gestion de configuration issu d'un projet Open Source qui permet l'édition collaborative de documents et qui retrouve les anciennes versions à la demande. La collaboration dans CVS est asynchrone multimaîtres et utilise un modèle de cohérence à terme. La détection des conflits est automatique mais concerne uniquement les erreurs syntaxiques, la résolution est laissée à la charge de l'utilisateur. Le système est fortement couplé au serveur central ce qui nécessite de le définir au préalable et peut poser problème en cas de panne.

Enfin depuis quelques années, des systèmes plus généralistes tels que Bayou

[Terry *et al.*, 1995] ou encore IceCube [Kermarrec *et al.*, 2001] sont apparus.
Bayou est un protocole généraliste de réplication optimiste de bases de données dans un environnement mobile [Demers *et al.*, 1994], [Terry *et al.*, 1995], [Petersen *et al.*, 1997], [Edwards *et al.*, 1997]. La collaboration dans Bayou est asynchrone multimaîtres et utilise un modèle de cohérence à terme. Toutes les répliques peuvent propager leurs écritures et voir rapidement les changements des autres sites mais un site primaire est désigné pour finalement décider de l'ordonnancement et de la résolution à appliquer en cas de conflit (basé sur un commit final). Les procédures de fusion s'avèrent difficiles à établir même pour les cas les plus simples [Terry *et al.*, 2000]
IceCube [Kermarrec *et al.*, 2001], [Preguiça *et al.*, 2001] est un protocole généraliste de réplication optimiste qui part de l'idée que la résolution de conflit si elle n'est pas automatique conduit toujours à supprimer des actions. La détection des conflits et la réconciliation sont basées sur des contraintes qui peuvent exprimer une intention de l'utilisateur ou un invariant sémantique. Deux types de contraintes sont utilisées : des contraintes statiques qui limitent l'espace de recherche (contrainte de type *Order* et *MustHave*) et des contraintes dynamiques qui sont constituées de préconditions et de postconditions des actions. La collaboration dans IceCube est asynchrone, multimaîtres et utilise un modèle de cohérence à terme. La détection et la résolution de conflits s'effectuent par le biais d'un site central et les contraintes doivent être préalablement établies. Le calcul de toutes les réconciliations possibles peut créer une explosion combinatoire.

Le tableau 2.6 présente une synthèse des principales caractéristiques de ces protocoles de réplication.

Protocole	**Topologie**	**Objets répliqués**	**Opérations**	**Diffusion**	**Journalisation**	**Détection des conflits**	**Réconciliation**
Usenet	Arbitraire et modifiable	Articles et Messages	Poster et Annuler	Par états	Non gérée	Non géré	Non géré
lotus Note		Enregistrements	Création, Modification, Suppression	Par états	Non gérée	Automatique	Application spécifique ou laissée à la charge de l'utilisateur
Coda	Client-serveur, en étoile	Fichiers et Répertoires	Création, Modification, Suppression	Par états et par opérations	Oui	Automatique	Application spécifique

Protocole	Topologie	Objets répliqués	Opérations	Diffusion	Journalisation	Détection des conflits	Réconciliation
Ficus	Etoile sans distinction entre clients et serveurs	Fichiers et Répertoires	Création, Modification, Suppression	Par états	Non gérée	Automatique	Application spécifique
CVS	Etoile autour d'un serveur central	Documents	Insertion, Suppression, Modification de lignes	Par opérations	Oui	Automatique mais seulement pour les erreurs syntaxiques	A la charge de l'utilisateur
Bayou	Arbitraire et modifiable mais utilisation d'un site primaire	Enregistrements de base de données	Requêtes SQL	Par opérations	Oui	Préconditions attachées à l'opération et définies par l'utilisateur ou l'application	Procédure de fusion attachée à l'opération
IceCube	Ad hoc autour d'un serveur central	Connexes à l'application	Spécifiées par l'application	Par opérations	Oui	Automatique grâce à des contraintes sémantiques définies par l'utilisateur	Application spécifique en fonction des contraintes

Table 2.6 – Synthèse des travaux en réplication optimiste

Tous les protocoles exposés ci-dessus sont donc asynchrones et multimaîtres et utilisent un modèle de cohérence à terme. Ils diffèrent principalement par le type d'objet répliqué (dépendant de l'application finale), la topologie du réseau (en étoile, en anneau ou ad hoc), la méthode de propagation des écritures (pull, push, par état ou par opérations) et la détection (basée sur des horloges ou sur la sémantique des opérations) et résolution des conflits (automatique, manuelle, basée sur des contraintes ou préconditions).

2.2.3 Quelles solutions en information géographique ?

Nous avons vu précédemment que la particularité des bases de données géographiques réside dans la nature complexe des données qui sont utilisées. En effet, lors de la création d'un système de gestion et de manipulation des données géographiques, il faut considérer la composante spatiale (point, ligne, polygone...), la représentation multiple (multi-thématiques ou multi-échelles), les différents systèmes de projection et la gestion des contraintes topologiques (non recouvrement ...). Par ailleurs, de nombreuses informations ne sont pas décrites explicitement dans la base de données géographiques et se déduisent par analyse de la géométrie. Ces particularités vont contribuer à influencer les performances des logiciels en matière de stockage, de chargement, de calcul, d'affichage et d'analyse des données. Une autre particularité directement liée à l'information géographique est la notion de longues transactions qui peuvent parfois s'étendre jusqu'à des semaines lors des campagnes de mise à jour des données. Enfin, les systèmes d'information géographique doivent permettent l'accès simultané aux données par plusieurs utilisateurs.

Il n'existe pas encore à ce jour de protocole dédié à la réplication optimiste asynchrone des données géographiques. Le problème est dû au fait qu'un des points critiques de la réplication optimiste reste l'avortement d'une transaction longue qui conduit quasi-systématiquement à l'annulation des mises à jour [Gray *et al.*, 1996]. Deux méthodes sont donc principalement utilisées pour gérer les transactions longues et l'accès multi-utilisateurs dans les SIG : la méthode dite de **Check in, Check out** et le **versionnement**.
Dans la première [ESRI, 2004], un extrait de la base est stocké dans un fichier dédié à des fins de mise à jour (Check in). Les données sont alors bloquées pour les autres utilisateurs qui ne peuvent plus faire de mise à jour jusqu'à ce que elles soient réinjectées dans la base (Check out). La collaboration est donc asymétrique.
Dans la technique de versionnement [le Roux, 2003], [Cellary et Jomier, 1990], les utilisateurs possèdent leur propre version de base de données issue d'une base de données de référence. Dès qu'une mise à jour est effectuée, une copie de l'objet initial est conservée et une nouvelle version est enregistrée. A la fin de la transaction, les versions sont comparées avec la version de référence et les conflits sont résolus la plupart du temps manuellement. La collaboration ici est asynchrone multimaîtres mais basée sur un serveur central contenant les données et s'avère être finalement un protocole pessimiste.

2.3 Analyse des travaux

L'état de l'art effectué ci-dessus montre qu'il existe plusieurs techniques lorsque l'on veut rendre disponibles des données à plusieurs utilisateurs simultanément. Dans notre étude, nous devons gérer des jeux de données spatiaux dans un contexte de prise de décision militaire où les utilisateurs sont répartis sur des sites distincts (le quartier général et le terrain d'action), doivent être autonomes (mises à jour sur chaque site, possibilité de travail en mode déconnecté...) et où ils échangent

régulièrement leurs informations (nouvelles données et mises à jour). Nous avons vu dans notre première analyse du problème (Cf. §1.2) que les données doivent être répliquées sur chaque système et que les mises à jour peuvent être saisies par plusieurs utilisateurs. Cela entraîne l'apparition d'incohérences lorsqu'un utilisateur veut intégrer les évolutions dans son jeu de données personnel et cela suppose de gérer la concurrence d'une part entre toutes les mises à jour disponibles et d'autre part entre les mises à jour et les données de l'utilisateur.

Une rapide analyse de ces travaux montre que plusieurs protocoles ont été définis dans les communautés systèmes ([Cederqvist *et al.*, 2001] ; [Kermarrec *et al.*, 2001]) et bases de données ([Oracle®, 2003] ; [Seshadri et Garrett, 2000]) pour gérer la réplication des données. Cependant, les solutions apportées sont souvent fonctions du besoin spécifique de l'application et ne sont donc pas réutilisables dans un contexte différent, ou supposent l'existence d'un serveur de référence centralisant les données.
Les mécanismes employés en information géographique pour gérer les données et les mises à jour ne sont pas non plus appropriés a notre étude car ils supposent que les données soient verrouillées aux autres utilisateurs jusqu'à ce que les mises à jour aient été intégrées (approche check in-check out [ESRI, 2004]), ou utilisent un serveur centralisé contenant les données de référence (versionnement [Cellary et Jomier, 1990]).

Une analyse plus poussée de cet état de l'art permet d'examiner **les différents aspects des systèmes de réplication**. Nous voyons en outre que la topologie du réseau, le type d'objets répliqués, les opérations de mise à jour et le mode de transfert **dépendent en général du contexte** dans lequel le système de réplication doit être installé. En effet, les caractéristiques des systèmes proposés dans la littérature (type d'objets répliqués, opérations de mise à jour ...) sont en règle générale **spécifiques à l'application** ([Kawell *et al.*, 1988] ; [Ratner, 1998] ; [Vesperman, 2003]) et le transfert des mises à jour s'effectue souvent par état, ce qui fait qu'**il n'existe pas de journal des opérations** ([Spencer et Lawrence, 1998] ;[Reiher *et al.*, 1994] ;[Moore, 1995]).
Dans notre étude, le type d'objet répliqué est une donnée géographique vectorielle stockée dans une base de données spatiale. La topologie du réseau se veut spécifique au contexte militaire et la propagation des écritures s'effectue indifféremment en mode push ou pull car les acteurs peuvent échanger l'information dont ils disposent. Les opérations de mises à jour et la structure des fichiers d'évolutions dépendent de la stratégie qui a été mise en place dans l'infrastructure de données spatiales (nous spécifierons explicitement, dans le prochain chapitre, le type des opérations permises ainsi que la structure des ensembles d'évolutions). Nous soulignons également l'intérêt d'avoir un journal des modifications qui permet de traiter les conflits particuliers dus à la mise à jour simultanée des données tels que la suppression d'un objet par un utilisateur et la modification de ce même objet par un autre par exemple.

Plus précisément, nous constatons que les principaux objectifs des algorithmes de réconciliation asynchrones optimistes sont d'assurer la **convergence des données** (lorsque le système est au repos, toutes les répliques ont les mêmes valeurs) et la **cohérence à terme** (lorsque le système est au repos, les données sont cohérentes

entre elles).
Dans notre contexte, **la convergence ne peut être conservée** car les jeux de données sont façonnés en fonction des besoins des unités. En effet, les besoins en terme de données des unités situées au quartier général ne sont pas les mêmes que celles sur le terrain d'action (plan de vol des avions VS plan de route des blindés). Les bases de données dédiées à chaque unité sont en fait dérivées (extraction d'une partie des données ou généralisation des données) d'une base de données de référence et peuvent couvrir des zones différentes (extraction de deux parties distinctes), être à des niveaux de détail ou à des échelles différents et contenir uniquement des couches thématiques spécifiques (par exemple, uniquement les routes) mais peuvent tout aussi se chevaucher ou encore se correspondre totalement. Pour garantir un certain niveau de collaboration, toutes les mises à jour effectuées au cours de la mission doivent être transmises aux unités mais certaines d'entre elles peuvent n'avoir aucun impact sur le jeu de données d'une autre unité ou pire provoquer des incohérences du fait du surplus d'informations inutiles pour l'utilisateur final.
En revanche, **la cohérence des jeux de données à terme est indispensable** dans le cadre d'une mission militaire. En effet, on imagine très mal l'utilisation de données erronées, mal situées ou encore non actualisées dans un contexte de prise de décision militaire. Mais, en information géographique, on ne peut pas comme pour les bases de données classiques définir des règles car l'élaboration d'un jeu de données géographique dépend de différents points de vue et abstractions. Une mise à jour effectuée à un certain niveau de détail pourra provoquer des incohérences dans un jeu de données établi à une précision différente mais cette mise à jour n'est pas forcément erronée.

La principale difficulté réside donc dans **la gestion de la cohérence** et plus particulièrement dans **l'automatisation de la détection et de la résolution de conflits**. Cette étape est souvent laissée à la charge de l'utilisateur ([Berliner, 1990], [MySQL, 2003]) ou spécifiée au préalable par le biais de contraintes ([Kermarrec *et al.*, 2001], [Terry *et al.*, 1995]), ce qui ne convient pas pour notre étude. En effet, pour résoudre les conflits, les protocoles classiques définissent des règles d'ordonnancement, donnent des priorités ou encore utilisent la sémantique de l'application pour définir des contraintes d'intégration mais ne considèrent jamais le besoin/point de vue de l'utilisateur final. Cependant, les travaux en matière de qualité des données géographiques ([David et Fasquel, 1997], [Vasseur *et al.*, 2005], [Devillers et Jeansoulin, 2005], [Harding, 2005]) ont montré la **nécessité de prendre en considération le point de vue de l'utilisateur** afin que les données soient effectivement en **adéquation avec les besoins** de celui ci.

Nous pensons donc qu'une réflexion analogue doit être menée en ce qui concerne la mise à jour de jeux de données géographiques. En effet, nous avons vu qu'il est possible qu'un utilisateur ait à intégrer de nombreux ensembles d'évolutions provenant de sources multiples, ce qui implique que les mises à jour ne sont pas forcément toutes pertinentes pour son application. Une étude de la pertinence des mises à jour devient alors nécessaire pour exclure les évolutions qui ne sont pas adéquates au

besoin de l'utilisateur. A notre connaissance, aucun travail n'a été mené sur la qualité des évolutions, les recherches sur les mises à jour considérant toujours que les évolutions sont pertinentes pour l'utilisateur. Nous pensons donc qu'il est intéressant d'évaluer la pertinence des mises à jour multi-sources et de ne retenir que celles qui soient indispensables à l'utilisateur.

Notre étude se doit de répondre aux deux questions suivantes non abordées dans la littérature. **Comment lors de la réconciliation de deux mises à jour conflictuelles pouvons nous prendre en considération les différents points de vue des personnes qui ont saisi les évolutions afin de proposer le meilleur choix en fonction des besoins de l'utilisateur final ; et quelles règles de cohérence doivent être spécifiées pour que l'évolution d'un objet géographique ne provoque pas d'incohérences dans le jeu de données cible ?**

> Un de nos objectifs est donc de s'assurer qu'après l'intégration des évolutions, l'utilisateur possédera finalement les « bonnes » données dans le sens les moins erronées possibles et les plus adaptées à son besoin immédiat.

A partir de ces observations, nous pouvons maintenant **préciser les spécifications du système de réplication** qui doit être mis en place dans l'infrastructure militaire et **définir les objectifs** permettant de maintenir au mieux la cohérence des jeux de données spatiaux répartis :

- Le type d'objet répliqué est une donnée géographique.
- La topologie du réseau est ad-hoc sans serveur central.
- La réplication des données est partielle (ex : sur les systèmes opérationnels) ou totale (chez les producteurs).
- Les utilisateurs ont la possibilité de travailler en mode déconnecté.
- Les opérations de mises à jour sont spécifiques au domaine.
- La diffusion des mises à jour s'effectue en continu et en mode mixte (push et pull).
- Le transfert des mises à jour s'effectue par opérations exclusivement.
- La détection des mises à jour concurrentes est automatique.
- La résolution des conflits
  - est fonction des besoins des utilisateurs.
  - est effectuée le plus automatiquement possible.
  - est conforme au modèle de cohérence défini dans l'infrastructure.

Nous proposons dans la partie suivante un système permettant de répondre à ces besoins. Nous spécifions dans un premier temps une politique de gestion des mises à jour qui permet de mieux gérer l'échange des évolutions dans l'infrastructure. Puis nous définissons une stratégie d'intégration des mises à jour basée d'une part sur des sessions de mise à jour et d'autre part sur un contrôle de la cohérence.

Chapitre 3

Stratégies d'intégration des mises à jour multi-sources

Notre travail de recherche se situe dans le cadre d'une application collaborative asynchrone utilisant un protocole de réplication optimiste de données géographiques dans un contexte de prise de décision. Plus particulièrement, nous nous intéressons à la mise à jour d'un jeu de données géographique par des évolutions provenant de sources multiples. Ces évolutions ne sont pas nécessairement pertinentes pour l'utilisateur final, peuvent être en conflits et créer des incohérences si elles sont intégrées sans précautions.

Ce chapitre constitue notre proposition de solution à cette problématique. Nous définissons dans un premier temps les caractéristiques de l'infrastructure que nous avons mise en place et sur laquelle nous nous appuyons pour fournir des réponses aux divers problèmes résultant du contexte de cette étude (§3.1). Ensuite, nous définissons la politique d'échange des mises à jour qui est mise en place dans l'infrastructure militaire en spécifiant notamment les opérations de mises à jour permises et la structure des ensembles d'évolutions (§3.2). Puis nous détaillons le profil de métadonnées que nous avons spécifié et qui sera utilisé tout au long de la stratégie d'intégration des mises à jour, en particulier pour le filtrage de l'information non pertinente et lors de la réconciliation des données conflictuelles (§3.3). Enfin, nous rentrons dans le détail de la stratégie d'intégration des mises à jour multi-sources (§3.4). Nous exposons premièrement la stratégie globale d'intégration des mises à jour multi-sources (§3.4.1). Nous abordons ensuite une réflexion concernant l'adéquation des évolutions avec les besoins de l'utilisateur final (§3.4.2). Puis, nous développons la partie sur la vérification de la cohérence (§3.4.3). Nous détaillons en particulier l'algorithme de contrôle de concurrence qui est basé sur l'opération de mise à jour et sur la nature des données manipulées. Puis, nous spécifions le modèle de cohérence sur lequel nous nous appuyons pour effectuer la réconciliation des mises à jour conflictuelles. Ce modèle s'appuie sur le profil de métadonnées de la norme ISO 19115 que nous avons détaillé plus tôt dans ce chapitre(§3.3). Enfin, nous élaborons des mécanismes de réconciliation des évolutions conflictuelles basé sur la sémantique des objets géographique et adapté aux besoins des unités militaires. Finalement, nous justifions l'intérêt d'utiliser des sessions de mises à jour (§3.4.4).

3.1 Infrastructure de données spatiales militaire

Nous avons constaté lors de notre première analyse (§1.2) que dans le contexte particulier dans lequel nous nous trouvons, il est judicieux de définir une infrastructure de données spatiales. En effet, dans une infrastructure, nous pouvons spécifier explicitement un certain nombre d'informations telles que le nombre de sites de déploiement, les moyens matériels utilisés, le type de coopération... Cela permet de limiter les risques dûs aux accès concurrents et garantit une collaboration efficace entre les différents utilisateurs de données [Pierkot *et al.*, 2006], [Pierkot et Mustiere, 2007].

En particulier, un SDI permet de définir la nature de l'information échangée entre les différents acteurs. Dans l'infrastructure que nous mettons en place ici, nous considérons que l'information échangée est une données géographique de type vectorielle. La première étude réalisée pour le projet Envol VDC a permis de définir des identifiants uniques et pérennes pour les données vectorielles utilisées par l'armée française [Raynal, 2005]. Nous utilisons dans notre travail les résultats de cette étude et nous supposons que les données utilisées dans l'infrastructure possèdent toutes initialement ce type d'identifiant.

Par ailleurs, les observations que nous avons faites au cours de l'état de l'art ont montrées que des composants indispensables d'une infrastructure de données spatiales sont les métadonnées (§2.1.3). L'armée française préconise l'utilisation des métadonnées conformes à un profil de la norme ISO 19115 [METAFOR, 2005], [ISO19115, 2003] et spécifie que celles-ci doivent accompagner les ensembles de données lors des transferts [CARGENE, 2004]. Nous supposons donc dans notre infrastructure que les métadonnées sont correctement remplies et sont envoyées simultanément avec les ensembles d'évolutions. Nous reviendrons en détail dans le paragraphe 3.3 sur le profil de métadonnées que nous avons spécifié et que nous utilisons dans l'infrastructure.

Un autre constat résultant de l'état de l'art (Cf.§2.1.3) est qu'en fixant les rôles et responsabilités des acteurs, on est capable de définir les autorisations en matière de transformation, mise à jour, diffusion et intégration des données spatiales [Nebert, 2004]. L'analyse du scénario de démonstration du projet EnvolVDC [Leblanc et Villot, 2003] a permis de spécifier les acteurs (§1.2) mais également de définir les rôles élémentaires de chacun des acteurs de l'infrastructure. Chaque acteur peut avoir un ou plusieurs rôles parmi ceux décrits ci-dessous :

- ✓ L'acquéreur
  - commande ou demande de l'information à un fournisseur extérieur à l'infrastructure,
  - réceptionne et garantit le résultat de l'acquisition,
  - transmet l'information à un transformateur ou un diffuseur.

- ✓ Le collecteur
  - effectue des mesures in situ ou des levés sur le terrain,

- traite le résultat de ses mesures pour atteindre un niveau suffisant d'exploitabilité,
- garantit le résultat de la collecte,
- transmet l'information à un transformateur ou un diffuseur.

✓ Le diffuseur
- reçoit, valide et archive l'information qui lui est fournie par un acquéreur, un transformateur, ou un collecteur,
- met l'information à disposition d'autres acteurs.

✓ Le transformateur
- reçoit l'information d'un collecteur, d'un acquéreur ou d'un diffuseur,
- modifie l'information si nécessaire ou crée des informations nouvelles résultant de l'analyse de l'information fournie,
- garantit le résultat de la modification ou de la production,
- transmet l'information à un diffuseur

✓ L'utilisateur
- reçoit l'information d'un diffuseur,
- exploite l'information directement pour conduire sa mission

La figure 3.1 montre la correspondance entre les différents rôles et les acteurs de l'infrastructure militaire. Les acteurs et leurs rôles étant connus, il devient alors plus aisé de juger des capacités et des limites de chacun en matière de gestion des ressources (diffusion, transformation, utilisation ...), notamment en ce qui concerne la mise à jour des jeux de données en leur possession.

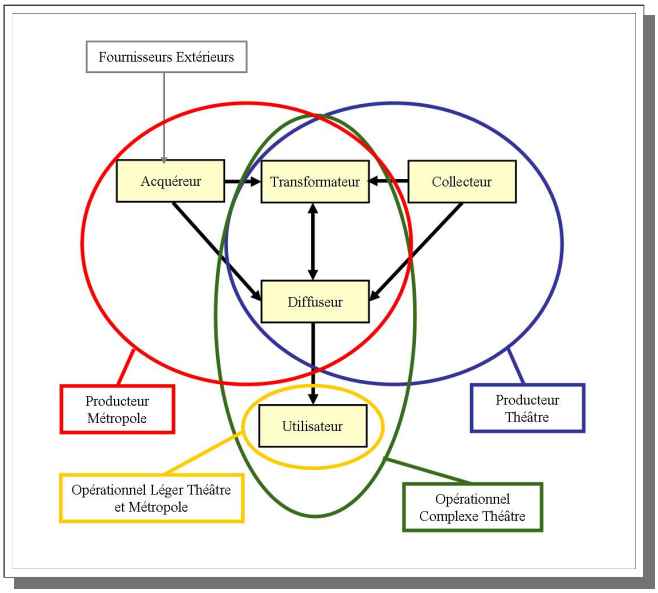

FIGURE 3.1 – Correspondance entre les rôles élémentaires et les acteurs de l'infrastructure

Nous avons ensuite modélisé les liens entre les acteurs, les données et les

évolutions, trois éléments fondamentaux de l'infrastructure [Pierkot *et al.*, 2005]. Ce modèle que nous avons appelé « modèle D-A-E » pour modèle « Données-Acteurs-Évolutions » est le socle de la stratégie d'intégration des évolutions que nous allons mettre en place au sein de l'infrastructure. En effet, en déterminant explicitement les relations entre ces trois entités, nous pouvons proposer un modèle de métadonnées (§3.3) basé sur cette structure, modèle que nous utiliserons tout au long de la stratégie pour d'une part filtrer les évolutions non pertinentes (§3.4.1) et d'autre part pour proposer des méthodes de réconciliation des mises à jour conflictuelles (§3.4.3). La figure 3.2 représente le modèle DAE (Données-Acteurs-Évolutions).

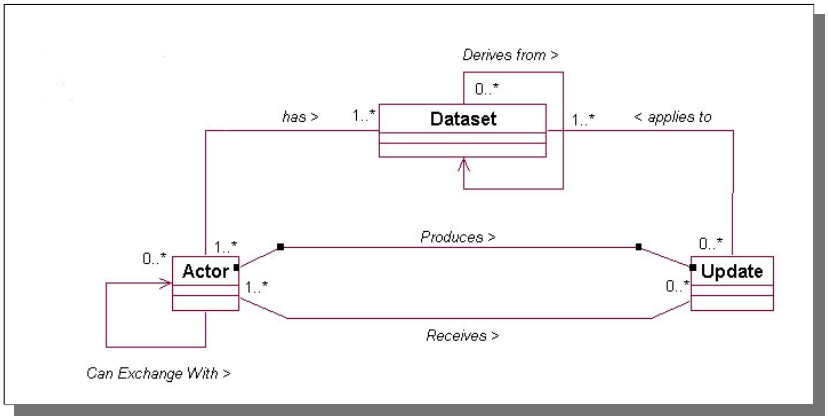

FIGURE 3.2 – Modèle Données-Acteurs-Évolutions

Ainsi, nous voyons qu'un jeu de données appartient au moins à un acteur et peut dériver d'un autre jeu de données. En effet, tous les jeux de données sont issus d'un jeu de données de référence initialement fourni en début de mission mais peuvent avoir été modifiés pour couvrir au mieux les besoins des utilisateurs (restriction de zone ou de couche d'information par exemple).
Un acteur possède quant à lui, au moins un jeu de données qu'il peut acquérir de différentes manières. Un acteur peut aussi produire ou recevoir des évolutions. On entend par "produire" le fait que c'est l'acteur lui même qui saisit les évolutions (par exemple grâce à un levé de terrain) et par "recevoir" le fait que les évolutions proviennent d'un autre acteur (collecte de nouvelles données par exemple). Les acteurs peuvent échanger de l'information entre eux, en fonction de leurs rôles dans l'infrastructure.
Les évolutions ont été collectées ou saisies pour un jeu de données particulier, elles s'appliquent donc au moins sur un jeu de données. Les évolutions appartiennent également au moins à un acteur, celui qui les a produit. Le modèle montre également qu'une évolution peut avoir été saisie par plusieurs acteurs et à partir de plusieurs jeux de données. Nous avons voulu ici souligner le fait que dans l'infrastructure, la mise à jour est asynchrone multimaîtres. Cela implique que les jeux de données peuvent être mis à jour à tout moment par des acteurs distincts qui possèdent de surcroît des jeux de données quelque peu différents car transformés en fonction de leurs besoins.

Nous avons ensuite voulu décrire plus précisément les interactions entre les

différents acteurs de l'infrastructure (Cf. figure 3.3). Dans ce modèle, nous considérons qu'une infrastructure globale est une organisation qui produit, transforme, utilise, met à jour et diffuse des données géographiques. L'infrastructure globale est divisée en infrastructures locales réparties sur différents sites.
Les acteurs se distinguent selon deux types : les acteurs internes et les acteurs externes. Les acteurs internes font partie d'une infrastructure locale et peuvent échanger leurs données et évolutions avec les autres acteurs, en fonction de leurs rôles dans l'infrastructure globale. Les acteurs externes quant à eux ne peuvent que fournir des nouvelles données ou mises à jour et ne peuvent aucunement recevoir des informations des autres acteurs.

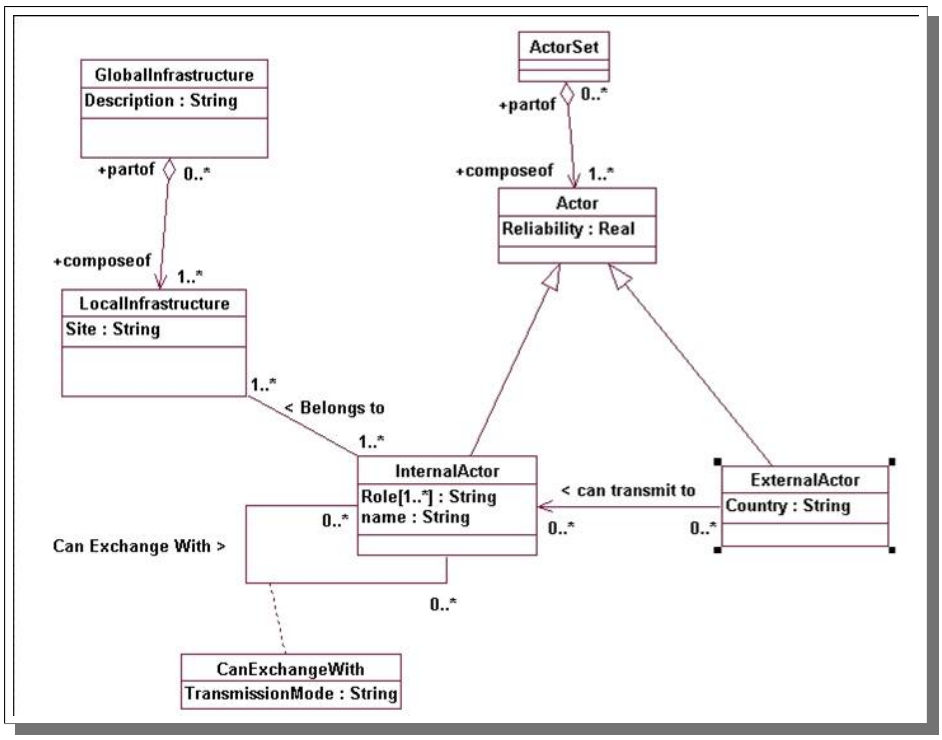

Figure 3.3 – Modélisation des acteurs dans l'infrastructure

Par exemple, dans le contexte militaire, l'infrastructure globale est l'organisation toute entière mise en place pour exécuter la mission. Les infrastructures locales sont les unités déployées sur le terrain ou au quartier général. Les acteurs internes sont les militaires en poste qui utilisent les données et les acteurs externes sont les alliés déjà présents sur la zone d'intervention, qui possèdent des informations qu'ils peuvent partager avec l'armée française.

3.2 Politique de gestion des évolutions

Un des avantages majeurs de l'utilisation d'une SDI est la possibilité de définir des politiques communes à tous les acteurs internes à l'infrastructure. Étant donné

que notre problématique concerne la mise à jour d'un jeu de données géographiques, nous avons défini une politique qui concerne la gestion des évolutions dans l'infrastructure. Celle-ci spécifie entre autres la nature des évolutions, la structure des ensembles d'évolutions et le format de livraison. Nous pouvons de ce fait gérer plus efficacement les échanges entre les différents acteurs de l'infrastructure.

3.2.1 Nature des évolutions

Nous définissons une **évolution** comme étant une action qui, lorsqu'elle est appliquée sur un objet géographique, conduit à la mise à jour du jeu de données dans lequel cet objet se trouve.

L'examen des travaux sur la classification des évolutions (§2.1.2) et les spécifications du système de réplication (§2.2) issues de l'analyse de l'état de l'art nous ont amenés à adopter une typologie basée sur les opérations elles-mêmes. Cependant, l'utilisation des deux seules opérations de création et de suppression comme le propose [Bedard *et al.*, 1997] ne nous semble pas suffisante car cette méthode ne distingue pas une opération de type modification à une autre de type création alors que nous pensons qu'il est préférable de gérer ces deux opérations individuellement, notamment lors de la vérification de la cohérence. La typologie des évolutions proposée par [Badard, 2000] est quant à elle riche et permet de traduire toutes les évolutions subies par les objets géographiques y compris les modifications attributaires. Cependant, nous n'utilisons pas cette classification dans le même but. L'objectif de [Badard, 2000] est d'extraire les évolutions d'une base de données spatiale, il veut donc rester au plus proche de la réalité. Notre but est de définir une politique de gestion des évolutions commune aux différents acteurs de l'infrastructure afin d'aider le processus de contrôle de la cohérence lors de l'intégration des évolutions dans un jeu de données utilisateur. Par conséquent, nous préférons limiter les types élémentaires des évolutions et ne considérer que les opérations qui nous paraissent nécessaires.

Nous avons formalisé cette classification en nous inspirant de la grammaire BNF étendue. [1] Les évolutions échangées dans l'infrastructure sont organisées de la manière suivante :

1. nous appliquons les règles suivantes à notre formalisation :
   - Les symboles terminaux sont notés en majuscule,
   - Les symboles non terminaux sont notés en minuscule,
   - la barre verticale (|) signifie une alternative,
   - L'étoile ($*$) indique une répétition de 0 ou plusieurs fois,
   - Le signe + indique une répétition d'au moins une fois.
   - Les accolades ({ }) sont utilisées pour décrire un ensemble de valeur

<evolution_elementaire> : := <nouvelle_donnee> | <mise_a_jour>
<nouvelle_donnee> : := <creation>
<mise_a_jour> : := <modification> | <suppression>
<modification> : := <modification_attributaire> | <modification_geometrique> | <modification_mixte>
<creation> : := <geometrie> <attribut*>
<suppression : := <id_objet>
<modification_attributaire> : := <id_objet> <attribut+>
<modification_geometrique> : := <id_objet> <geometrie>
<modification_mixte> : := <id_objet> <geometrie> <attribut+>
<geometrie> : := <type_geometrie> <coordonnees>
<id_objet> : := NOMBRE_HEXADECIMAL
<attribut> : := {NOM, VALEUR}
<type_geometrie> : := POINT | LIGNE | SURFACE
<coordonnees> : := {ABSCISSE,ORDONNEE}+

Dans notre classification, une **évolution élémentaire** est donc soit une **nouvelle donnée**, soit une **mise à jour** d'une donnée existante.

Une nouvelle donnée correspond à une opération de type **création** et une mise à jour est une opération de type **suppression** ou **modification**.

La modification peut s'appliquer sur la **géométrie** de l'objet, sur les **attributs** ou sur les deux caractéristiques simultanément, on dit alors que la modification est **mixte**.

Seul l'**identifiant** de l'objet est fourni avec une opération de type **suppression**. La raison étant que cette information est suffisante pour retrouver la donnée et la supprimer de la base de données.

Pour une évolution de type **création**, la géométrie de l'objet et la liste des attributs sont donnés. L'identifiant de l'objet concerné n'existe pas encore à ce stade et sera déterminé lors de l'intégration de l'évolution dans la base de données.

Concernant les **modifications**, les éléments dépendent du type de mise à jour qu'a subit l'objet :
- Dans le cas d'une **modification géométrique**, l'identifiant et la nouvelle géométrie de l'objet sont fournis.
- Dans le cas d'une **modification attributaire**, l'identifiant de l'objet et la liste des attributs sont donnés. La liste contient au moins un attribut qui est concerné par la mise à jour.
- Dans le cas d'une **modification mixte**, l'identifiant, la nouvelle géométrie et la liste des attributs sont transmis. La liste contient au moins un attribut qui est concerné par la mise à jour.

La **géométrie** est communiquée par le couple (type,coordonnées) qui précise le type de géométrie utilisé pour saisir l'évolution (point, ligne, polygone) et les

coordonnées associées. Les coordonnées prennent la forme d'une liste de valeurs (abscisse, ordonnée) permettant de localiser spatialement l'objet.

L'**identifiant** détermine l'objet qui a subi la mise à jour. C'est un nombre hexadécimal résultant de la production d'une empreinte numérique par à un algorithme de type MD5 [Rivest, 1992]. Cet identifiant est créé dans un premier temps lors de la constitution du jeu de données de référence et ensuite lors de l'intégration des nouvelles données dans la base de données. Il est par ailleurs unique, pérenne et ne peut être modifié ([Raynal, 2005]).

Les **attributs** transmettent les propriétés non spatiales de l'objet. Ils sont livrés sous forme d'une liste de couples (nom,valeur)

La figure 3.4 montre l'arbre simplifié correspondant à la structure des évolutions telles que nous les avons définies dans l'infrastructure militaire.

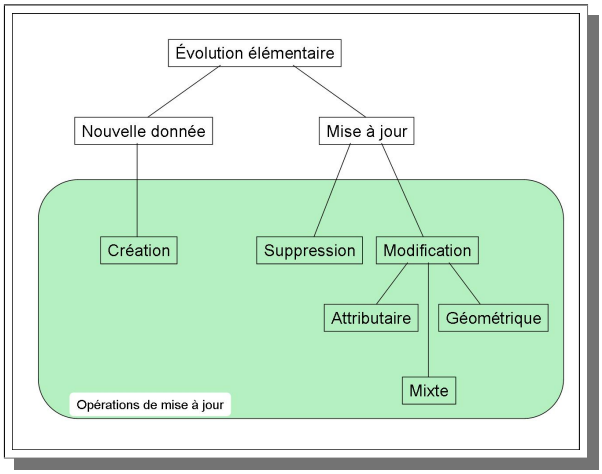

Figure 3.4 – Structure des évolutions de l'infrastructure

3.2.2 Structure des ensembles d'évolutions pour la livraison

La conclusion du chapitre précédent nous a permis de spécifier le système de réplication qui est mis en place dans l'infrastructure. Parmi les contraintes que nous avons imposées, le transfert des évolutions doit s'effectuer exclusivement par opérations. Cela signifie que seules les opérations de mises à jour sont livrées aux utilisateurs. Les évolutions que nous avons définies dans l'infrastructure sont donc classées en fonction du type d'opérations qui s'applique sur l'objet, et nous pouvons dès à présent, définir l'organisation des ensembles d'évolutions.

Nous décidons donc de structurer les ensembles d'évolution selon un modèle commun basé sur le type d'opérations qui conduit à l'évolution de l'objet. La livraison des évolutions s'effectue de ce fait grâce à un journal qui recense toutes les opérations que le jeu de données a subi depuis sa dernière modification. Nous pouvons utiliser ce type de format de livraison car tous les jeux de données utilisés

dans l'infrastructure, même s'ils ont été dérivés en fonction du besoin de l'utilisateur final, dépendent d'un même jeu de référence. Nous avons donc la possibilité de diffuser uniquement les opérations sur les objets concernés par l'évolution.

La structure utilisée dans l'infrastructure pour définir les ensembles d'évolutions est la suivante :

```
Produit Nom_produit
  Pour chaque EnsEvol Nom_EnsEvol
  Pour chaque Evol Nom_Evol
    Objet  = ID
    Géométrie = (type, coordonnees)
    Attributs* = (nom, valeur)
    Lien_Evol* = Nom_Evol_Associée
    Lien_Data* = ID_Data_Associée
  Fin Evol
 Fin Theme
Fin Produit
```

Dans cette organisation, un **produit** regroupe plusieurs ensembles d'évolutions provenant d'une infrastructure locale et saisi à partir d'un jeu de données unique.

Un **ensemble d'évolutions** contient une suite d'évolutions ayant été saisie pour une couche thématique particulière. Par exemple, le thème transport contenant les routes et voies ferrées.

Une **évolution** est telle qu'elle a été définie ci-dessus et correspond à une opération de mise à jour (création, suppression, modification géométrique, modification attributaire, modification mixte). Elle fournit les informations sur l'identifiant de l'objet, la géométrie et la liste des attributs associés à l'évolution. D'autre part, une évolution peut être liée à d'autres évolutions ou à des données, le lien correspondant soit à une relation topologique entre les objets (par exemple, le pont chevauche cette rivière, cette route est connectée à cette autre route), soit à une relation de dépendance entre les évolutions (par exemple, une suppression suivie d'une création).

Finalement, tous les ensembles d'évolutions échangés à l'intérieur de l'infrastructure sont structurés de la même façon et contiennent uniquement les types d'évolutions élémentaires définies par la classification (Cf.figure 3.5). Ce format présente plusieurs avantages : premièrement, il permet de respecter les spécifications du système en permettant le transfert des évolutions par opération, deuxièmement il permet l'utilisation et la sauvegarde des journaux de modification qui peuvent s'avérer utiles en cas de problème (retour à une version antérieure par exemple) et enfin il permet la comparaison entre les évolutions effectuées sur différents sites de l'infrastructure afin de détecter celles qui sont en conflit.

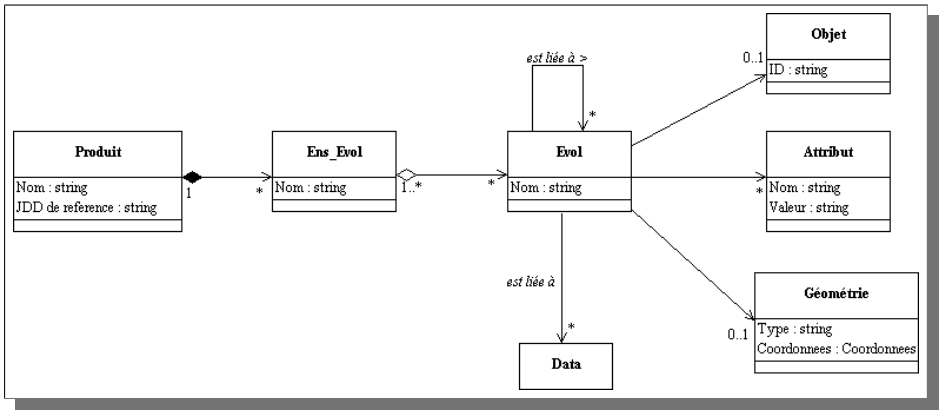

Figure 3.5 – Modélisation des ensembles d'évolutions de l'infrastructure

Cependant, cette politique de gestion des évolutions ne suffit pas à régler tous les problèmes liés à la mise à jour par des évolutions multiples et provenant de sources distinctes. Elle apporte certes une brique à la solution finale, notamment en ce qui concerne la détection des conflits (nous verrons comment nous utilisons ces informations pour aider le contrôle de concurrence au § 3.4.3), mais les informations recueillies dans ce schéma ne sont pas suffisantes dès lors que l'on veut réconcilier les évolutions concurrentes (nous verrons la solution que nous préconisons au §3.4.3) ou encore lorsqu'on veut filtrer les évolutions non pertinentes pour l'utilisateur final (nous discuterons de cette partie au §3.4.2). Nous proposons pour cela d'utiliser des métadonnées afin d'obtenir des informations supplémentaires sur les évolutions telles que leur qualité ou encore leur provenance.

3.3 Modèle de métadonnées

Les **métadonnées** sont des éléments importants pour l'accès, la diffusion et la bonne utilisation des données géographiques. [Servigne *et al.*, 2005] soulignent qu'elles permettent de documenter le plus précisément possible les données afin de faciliter leur partage et leur diffusion et ceci en vue de simplifier leur intégration et leur réutilisation. De plus, comme nous l'avons vu dans l'état de l'art, les métadonnées constituent un des composants indispensable d'une infrastructure de données spatiales (Cf.§2.1.3).

Les métadonnées sont également utiles pour inventorier les données chez le producteur, permettre aux utilisateurs d'identifier les produits (localisation, moyens d'accès) ou encore pour fournir une description complète des données (qualité, système de projection...). Nous pensons donc que leur utilisation peut aider la gestion des évolutions et faciliter les échanges dans l'infrastructure militaire. En particulier, dans notre étude, elles vont permettre d'une part de filtrer les évolutions non pertinentes pour l'utilisateur et d'autre part d'aider la résolution de conflits, par comparaison des informations qu'elles véhiculent (identification de la source, qualité des évolutions, actualité du produit...). Nous allons montrer dans la quatrième partie de ce chapitre comment utiliser ces métadonnées pour aider l'intégration des évolutions multi-sources dans un jeu de données utilisateur.

Parmi les nombreuses informations que l'on peut trouver dans les métadonnées (identification, représentation spatiale, description des contenus, ...), il y a celles concernant la **qualité des ressources**. Nous avons vu dans l'état de l'art que la qualité se distingue en deux types : la **qualité interne** qui décrit le degré de conformité des données avec les spécifications des produits et la **qualité externe** qui décrit le degré d'adéquation des données avec le besoin de l'utilisateur (§2.1.4). Les métadonnées de qualité fournissent généralement des indications sur la qualité interne des produits car elles sont souvent remplies par le producteur de données qui ne connaît pas a priori l'utilisation finale qui en sera faite. La qualité externe est plus difficile à apprécier car elle dépend des besoins de chaque utilisateur et diffère donc d'un utilisateur à l'autre.

Dans notre étude, nous allons utiliser la qualité interne des données essentiellement lors de la réconciliation. En effet, lorsqu'un choix s'impose entre deux évolutions ayant des caractéristiques similaires, il est utile de connaître certaines informations qui permettent d'apprécier la qualité de l'évolution telles que la précision ou la cohérence logique.
Par ailleurs, nous utiliserons la qualité externe à deux moments de la stratégie d'intégration : lors de la vérification de la pertinence et lors de la réconciliation des données conflictuelles. En effet, en comparant ces informations avec les besoins réels de l'utilisateur, le processus pourra décider de l'action à effectuer. Ce choix conduira finalement à l'intégration ou à l'exclusion de l'évolution.
Certains critères de qualité décrits dans l'état de l'art (§2.1.4) peuvent servir à décrire conjointement la qualité interne et la qualité externe d'un produit (actualité, généalogie, ...) mais ces éléments de métadonnées ne suffisent cependant pas à évaluer correctement la qualité en fonction du besoin de l'utilisateur final. En effet, comment savoir par exemple si une évolution est pertinente ou possède une valeur ajoutée pour l'utilisateur qui en fera l'usage ? Nous pensons donc qu'il faut ajouter des critères aux métadonnées afin de définir l'adéquation des évolutions avec les exigences de l'utilisateur.

Par ailleurs, il nous semble intéressant de définir la qualité à des **granularités** différentes c'est à dire non seulement sur l'ensemble des évolutions mais également sur les évolutions elles-mêmes. En effet, l'évaluation de la qualité au niveau de l'ensemble tout entier permet de filtrer rapidement les collections qui ne sont pas pertinentes pour l'utilisateur, alors que la mesure de la qualité au niveau des évolutions aide le processus de réconciliation à effectuer le meilleur choix lorsque deux évolutions sont en conflit.

La figure 3.6 montre les métadonnées de qualité associées aux évolutions. Nous verrons dans le paragraphe 3.3.3 le détails des métadonnées de qualité que nous avons ajouté a notre profil.

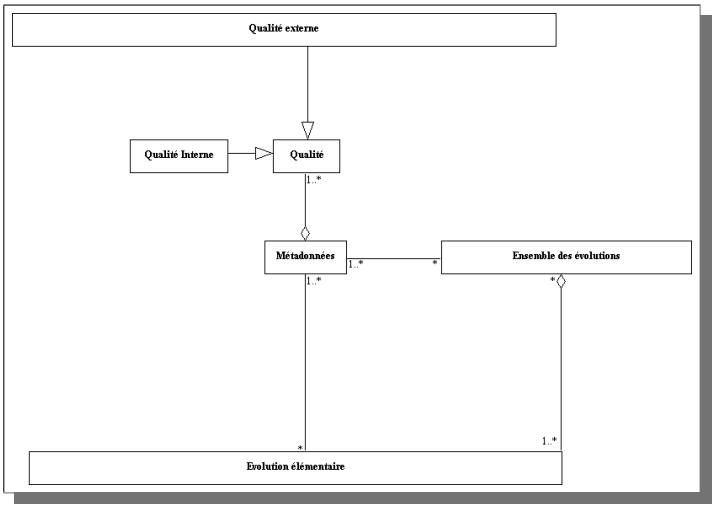

Figure 3.6 – Métadonnées de qualité pour les évolutions

Pour favoriser l'interopérabilité entre les différents utilisateurs de métadonnées, il faut utiliser des normes créées à cet effet (cf. §2.1.3). La directive européenne [INSPIRE, 2007] insiste d'ailleurs sur l'importance d'utiliser des métadonnées harmonisées afin de faciliter l'usage des données géographiques. La norme qui aujourd'hui intéresse le plus la communauté est l'ISO 19115 (Information géographique - Métadonnées). L'état français a, du reste, demandé une étude sur la mise en oeuvre de ce standard à l'échelon national [ADAE, 2006]. Par ailleurs, l'armée française préconise également l'utilisation des métadonnées conformes à la norme ISO 19115 [CARGENE, 2004] et un profil de cette norme pour les données militaires a été créé [METAFOR, 2005] (§ 3.3.2). C'est donc tout naturellement que nous supposons dans ce travail que les métadonnées doivent respecter le format ISO 19115.

3.3.1 La norme ISO 19115

Depuis 2003, ISO 19115 [ISO19115, 2003] est la norme des métadonnées spécifique à l'information géographique. Elle a été établie par le comité technique 211 de l'Organisation Internationale de Normalisation (International Organization for Standardization, en anglais). L'ISO 19115 et ses normes associées font partie d'une famille de normes extensibles et modulaires. D'autres standards ont également été utilisés pour élaborer le standard de métadonnées. Nous listons l'ensemble des normes associées et complémentaires de la norme ISO 19115 en annexe A.

Le standard ISO 19115 définit les métadonnées permettant de décrire l'information géographique. Ces métadonnées peuvent s'appliquer à différentes granularités :

- Au niveau d'une collection de données qui partagent les mêmes caractéristiques (`DataSeries`). Par exemple l'ensemble des données vectorielles de type VMAP.
- Au niveau d'un ensemble de données cohérentes produit ou diffusé par un même fournisseur (`DataSet`). Par exemple, l'ensemble des données vectorielles de type VMAP, à l'échelle 1/250000, produit par le producteur métropole.
- Au niveau d'un ensemble d'objets géographiques (`FeatureType`). Par exemple, l'ensemble des objets de la classe « Tronçon de route ».

– Au niveau d'un objet géographique du monde réel (`FeatureInstance`). Par exemple, la route RN2.
– Au niveau d'un attribut d'un objet géographique (`AttributeInstance`). Par exemple, l'attribut « nombre de voie » de la route RN2.

La norme se décompose en plusieurs **sections** de métadonnées qui contiennent une ou plusieurs **entités** de métadonnées, elles mêmes constituées d'**éléments** de métadonnées.
Le standard est présenté grâce au langage de modélisation UML. Les sections sont représentées avec des packages, les entités par des classes et les éléments par des attributs.
Les sections, entités et éléments de métadonnées peuvent être obligatoire, conditionnelles (obligatoires sous condition) ou encore optionnelles. Le détail des sections et entités de métadonnées est donné en annexe A.

Par ailleurs, le standard ISO 19115 définit volontairement un ensemble volumineux de métadonnées de telle sorte qu'elles puissent être exploitées par de nombreux d'utilisateurs. Généralement, un sous ensemble de la norme suffit largement à une communauté spécifique. Cependant, la norme recommande qu'une quantité minimale de métadonnées soit maintenue. Les éléments constituant ce minimum est appelé le noyau (cf. Figure 3.7) et permet de répondre aux questions les plus fréquemment posées par des utilisateurs de données géographiques (quoi ? où ? quand ? qui ?).

Dataset title (M) (MD_Metadata > MD_DataIdentification.citation > CI_Citation.title)	**Spatial representation type** (O) (MD_Metadata > MD_DataIdentification.spatialRepresentationType)
Dataset reference date (M) (MD_Metadata > MD_DataIdentification.citation > CI_Citation.date)	**Reference system** (O) (MD_Metadata > MD_ReferenceSystem)
Dataset responsible party (O) (MD_Metadata > MD_DataIdentification.pointOfContact > CI_ResponsibleParty)	**Lineage** (O) (MD_Metadata > DQ_DataQuality.lineage > LI_Lineage)
Geographic location of the dataset (by four coordinates or by geographic identifier) (C) (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicBoundingBox or EX_GeographicDescription)	**On-line resource** (O) (MD_Metadata > MD_Distribution > MD_DigitalTransferOption.onLine > CI_OnlineResource)
Dataset language (M) (MD_Metadata > MD_DataIdentification.language)	**Metadata file identifier** (O) (MD_Metadata.fileIdentifier)
Dataset character set (C) (MD_Metadata > MD_DataIdentification.characterSet)	**Metadata standard name** (O) (MD_Metadata.metadataStandardName)
Dataset topic category (M) (MD_Metadata > MD_DataIdentification.topicCategory)	**Metadata standard version** (O) (MD_Metadata.metadataStandardVersion)
Spatial resolution of the dataset (O) (MD_Metadata > MD_DataIdentification.spatialResolution > MD_Resolution.equivalentScale or MD_Resolution.distance)	**Metadata language** (C) (MD_Metadata.language)
Abstract describing the dataset (M) (MD_Metadata > MD_DataIdentification.abstract)	**Metadata character set** (C) (MD_Metadata.characterSet)
Distribution format (O) (MD_Metadata > MD_Distribution > MD_Format.name and MD_Format.version)	**Metadata point of contact** (M) (MD_Metadata.contact > CI_ResponsibleParty)
Additional extent information for the dataset (vertical and temporal) (O) (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_TemporalExtent or EX_VerticalExtent)	**Metadata date stamp** (M) (MD_Metadata.dateStamp)

Figure 3.7 – Noyau de l'ISO 19115

Le noyau défini dans l'ISO 19115 est constitué d'éléments obligatoires, option-

nels ou conditionnels. La norme recommande d'utiliser les éléments optionnels afin d'augmenter l'interopérabilité et de permettre aux utilisateurs de comprendre sans ambiguïté les données fournies par le producteur et/ou le distributeur. Le détail des éléments du noyau est fourni en annexe A.

Une communauté spécifique d'utilisateurs de données géographiques n'utilise donc généralement qu'une partie des métadonnées définies dans la norme et a contrario a souvent besoin d'ajouter des métadonnées qui ne sont pas spécifiées dans le standard. ISO 19115 permet cela grâce à la définition de profils communautaires (cf. Figure 3.8). Un profil permet de restreindre la norme à un sous ensemble d'éléments qui doivent être obligatoires et de l'étendre en ajoutant des sections, entités et éléments manquants.

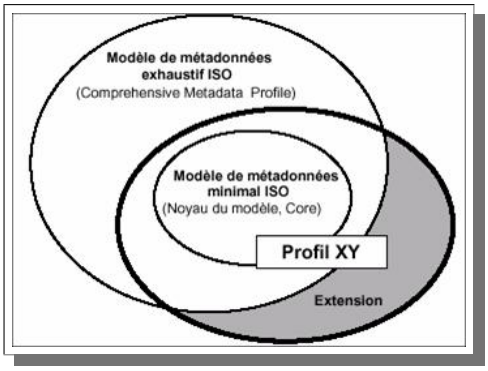

Figure 3.8 – Profil communautaire de l'ISO 19115

Le standard stipule qu'un profil communautaire doit contenir le noyau ISO 19115 c'est à dire au moins toutes les informations obligatoires et toutes les métadonnées conditionnelles lorsque les conditions requises sont satisfaites. Par ailleurs, un profil doit adhérer aux règles de définition des extensions et ne doit pas changer ni le nom, ni la définition, ni le type d'un élément de métadonnées.
La norme définit avec précision les mécanismes de restriction et d'extension. Une extension ne peut en aucun cas permettre ce qui n'est pas autorisé dans le standard.

Les types d'extensions accordés par l'ISO 19115 sont les suivants :
- Ajouter un nouveau package, une nouvelle classe ou un nouvel attribut. Les nouvelles classes doivent hériter de classes existantes.
- Créer une nouvelle liste de codes pour contraindre le domaine d'un attribut qui était initialement défini comme un texte libre.
- Créer de nouveaux éléments dans une liste de code existante.

Les types de restrictions accordés par l'ISO 19115 sont les suivants :
- Imposer une contrainte plus exigeante à une classe ou un attribut (rendre obligatoire ce qui est optionnel).
- Restreindre le domaine de valeurs d'un attribut.

3.3.2 METAFOR : le profil français de l'ISO 19115 pour la gestion des données militaires

DNG3D est un programme interarmées destiné à doter la défense nationale des moyens de disposer des données numériques géographiques et des modèles 3D nécessaires à l'emploi des systèmes d'armes et des systèmes d'information sur les zones d'intérêt de la défense [CARGENE, 2004].
Ce programme préconise de s'appuyer sur la norme ISO 19115 afin de définir les métadonnées nécessaires aux besoins de la défense.
METAFOR est le format de fichiers résultant de ce travail et contient les métadonnées relatives aux produits géographiques utilisés par l'armée française [METAFOR, 2005]. Ce format est basé sur un profil communautaire de la norme ISO 19115 et est construit sur une implémentation en XML de l'ISO 19115 et des normes associées.

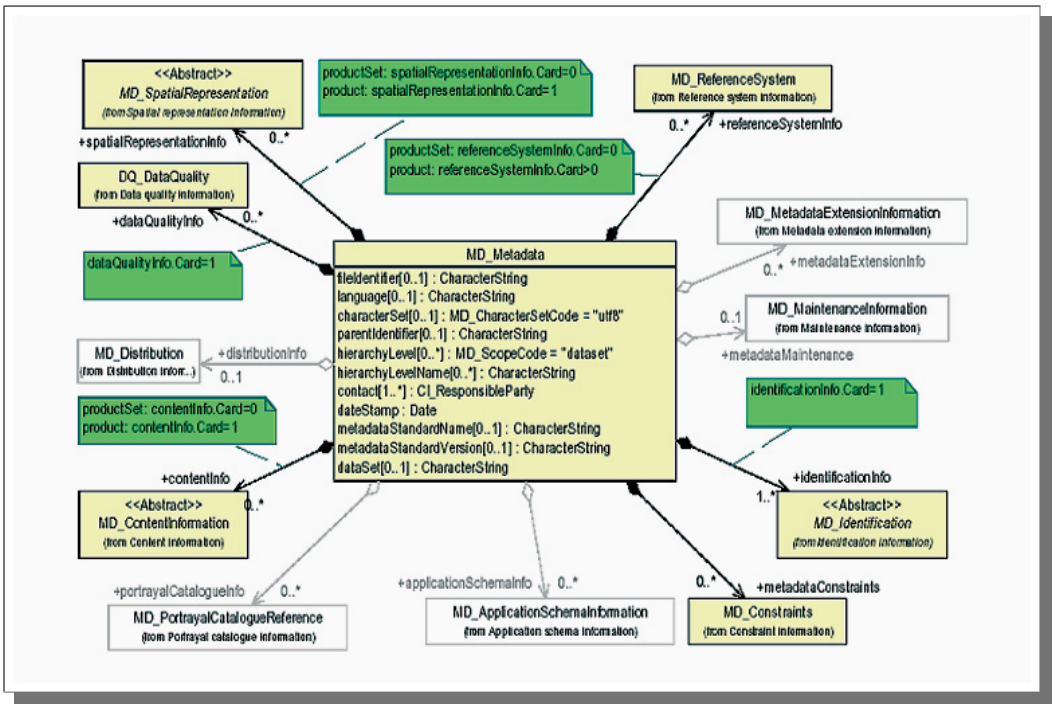

Figure 3.9 – Principales sections de METAFOR

Les sections de métadonnées de la norme ISO 19115 qui ont été gardées dans METAFOR sont :

- L'identification des ressources (`MD_DataIdentification`) et en particulier :
  - la description des aperçus et vignettes (`MD_BrowseGraphic`),
  - les mots clés permettant de caractériser les produits (`MD_Keywords`),
  - le format dans lequel sont encodés les produits (`MD_Format`),
  - les contraintes légales (`MD_LegalConstraints`) et de sécurité (`MD_SecurityConstraints`).
- La qualité des ressources à partir de la classe `MD_DataQuality` et de ses classes dérivées :

  - `LI_Lineage` qui fournit les informations de généalogie. Elle est obligatoire dans METAFOR et doit décrire au moins une source. Elle utilise :
    - `DG_Source` pour décrire brièvement les sources,
    - `DG_ProcessStep` pour décrire les phases du processus de production.
  - `DQ_Element` qui permet de qualifier certains éléments de la qualité tels que :
    - l'exhaustivité (`DQ_Completeness`),
    - la précision des attributs (`DQ_ThematicAccuracy`),
    - la cohérence logique (`DQ_LogicalConsistency`),
    - la précision temporelle (`DQ_TemporalAccuracy`),
    - la précision géométrique (`DQ_PositionnalAccuracy`).

    Elle permet également d'exprimer différents types de résultats provenant des mesures qualités effectuées :
    - un résultat quantitatif (`DQ_QuantitativeResult`),
    - un résultat pour vérifier la conformité des données au regard de la mesure(`DQ_ConformanceResult`).
- Des informations sur la représentation spatiale utilisée pour géoréférencer l'image (`MD_Georectified`).
- Les informations sur le contenu des produits sont données par le biais de la classe `MD_CoverageDescription`. Elles se distinguent en fonction des données manipulées :
  - `MD_CoverageDescription` pour les MNT,
  - `MD_ImageDescription` pour les images.
- Le système de référence (`MD_ReferenceSystem`) qui identifie un système de référence de coordonnées. La définition des systèmes de coordonnées et des unités de mesures est fournie dans les registres `DG_UomRegister` et `DG_CrsRegister`.

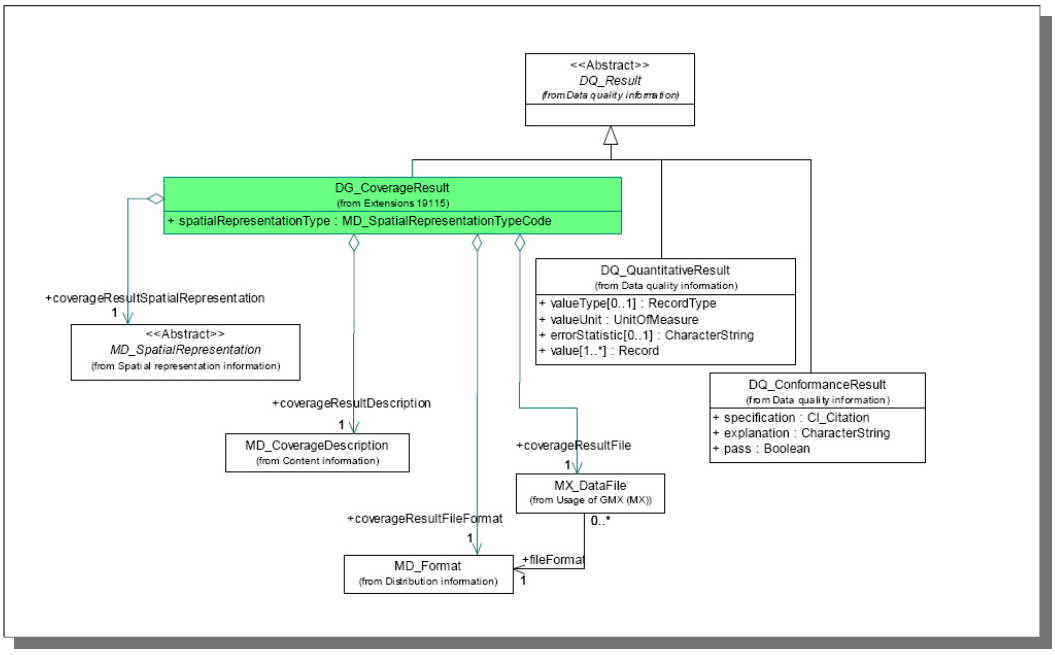

Figure 3.10 – Résultat des mesures de qualité dans METAFOR

Parmi les informations non présentes dans la norme ISO 19115 et ajoutées dans METAFOR, on trouve :

- Des couches de qualification (`DG_CoverageResult`) qui permettent l'échange de couches de données géoréférencées comme résultat d'une mesure de qualité (cf. figure 3.10). Cependant, ces couches de qualification ne contiennent pas pour la plupart des résultats de mesures de la qualité au sens strict du terme, mais plutôt des résultats de mesure permettant d'appréhender la capacité du jeu de données à répondre à l'emploi que l'utilisateur envisage pour lui.
- Une classe `DG_Usability` qui permet d'intégrer l'ensemble des couches de qualification dans la classification des mesures de la norme (cf. figure 3.11)

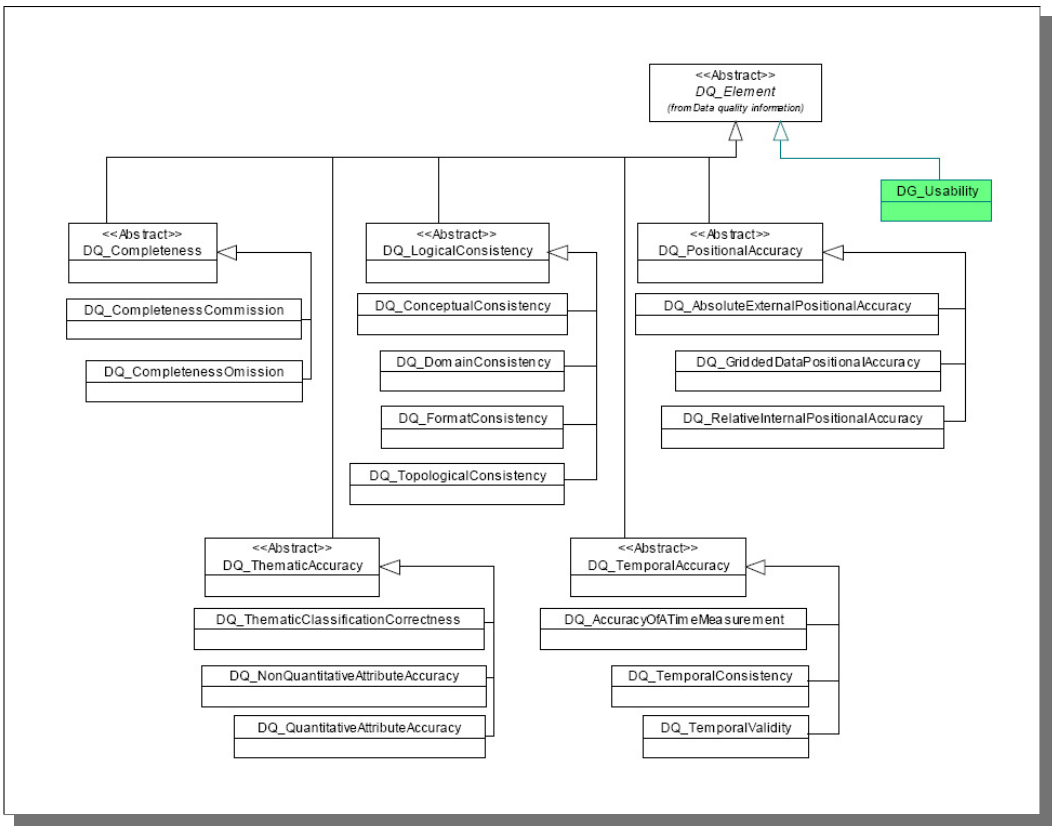

Figure 3.11 – Classification des mesures de qualité dans METAFOR

METAFOR est donc le format de métadonnées utilisé par l'armée française pour la gestion des données militaires. Ce format prend en considération les besoins des utilisateurs en matière d'échange d'informations au cours des missions militaires. Cependant, les spécifications DNG3D ne considèrent à ce jour que les produits MNT et images et le format METAFOR n'a été établi à l'origine que pour satisfaire l'échange de telles données. Les données vectorielles ne sont donc pas encore prises en compte dans CARGENE et dans METAFOR. Le standard ISO 19115 possède des classes spécifiques aux données vectorielles (`MD_VectorSpatialRepresentation`) mais celles ci n'ont pas été activées dans METAFOR.

3.3.3 MUMSDI, un profil de métadonnées pour les évolutions de données militaires

ISO 19115 et METAFOR sont donc conçus pour fournir l'information sur l'utilisation et l'échange de jeux de données. Les ensembles d'évolutions ne sont pas pris en compte dans la norme ou le profil de l'armée française. En effet, les métadonnées définies dans ISO 19115 ou METAFOR se rapportent à un jeu de données, un ensemble de jeux de données, un produit ou encore un objet ou un attribut mais aucunement ne référence un ensemble d'évolutions ou une évolution élémentaire (telle qu'une création, une suppression ou une modification). Il n'est donc pas possible à l'heure actuelle de fournir des métadonnées se rapportant à un ensemble d'évolutions que l'on voudrait fournir à un utilisateur qui possède déjà le jeu de données de référence.

Notre analyse des besoins sur les informations de mise à jour a montré que pour gérer au mieux les évolutions nous devons être capable :
- D'identifier la source des informations disponibles.
- De résoudre les conflits dus à la mise à jour.
- De valider une unique version de référence.

Il faut par conséquent ajouter des éléments de métadonnées qui prennent en compte ces nouveaux besoins. Nous proposons donc d'étendre ISO 19115 aux évolutions. Nous avons pour cela créé un profil de métadonnées que nous avons appelé MUMSDI (Metadata for Updating a Military Spatial Data Infrastructure), qui est spécifiquement dédié à la gestion des évolutions de données militaires. Ce profil s'appuie sur le format METAFOR pour tout ce qui concerne la description des données militaires, et sur le mécanisme de restriction et d'extension fourni dans le standard ISO 19115 pour tout ce concerne les éléments caractérisant les évolutions de données vectorielles, notamment des informations de qualité et en particulier celles qui permettent de déterminer l'adéquation aux besoins des utilisateurs.

Une des exigences de notre approche est que les métadonnées fournies avec les évolutions doivent être correctement renseignées. Cependant, comme le souligne le rapport final du projet GINIE, les données spatiales sont souvent mal, voire pas du tout, documentées du fait d'un trop grand nombre d'éléments à renseigner [GINIE, 2004]. Pour éviter ce risque majeur, nous devons choisir parmi les éléments de métadonnées, ceux qui sont indispensables et ceux qui peuvent être optionnels tout en certifiant que ce nombre minimum est suffisant pour assurer la traçabilité des évolutions.
Une autre contrainte imposée par la norme elle-même spécifie que chaque profil doit contenir au minimum le noyau défini dans l'ISO 19115. Ce noyau contient plusieurs éléments obligatoires, conditionnels et optionnels et le standard recommande de les exploiter tous dans leur intégralité.
Pourtant dans notre contexte, nous pensons qu'il n'est pas nécessaire d'utiliser tous les éléments optionnels du noyau car, dans ce travail, les métadonnées qui accompagnent les évolutions sont des métadonnées d'exploitation utilisées par un processus d'aide à l'intégration des mises à jour plutôt que des métadonnées d'ex-

ploration utilisées à des fins de recherche des évolutions. Des métadonnées telles que celles permettant de décrire les moyens pour obtenir les ressources en ligne (`CI_OnlineResource`) deviennent alors obsolètes et sont inutiles dans cette étude.
Par ailleurs, certains éléments obligatoires sous conditions n'ont plus lieu d'existence dans notre profil. Par exemple, le noyau ISO oblige l'utilisation de `MD_Metadata` > `MD_DataIdentification.characterSet` et de `MD_Metadata.characterSet` si le jeu de caractère utilisé n'est pas défini par le standard ISO 10646-1. Dans notre infrastructure, nous imposons l'utilisation systématique du standard ISO 10646-1 pour définir les jeux de caractères utilisés dans les ensembles d'évolution. Il en est de même pour `MD_Metadata.language`.
Finalement, nous n'avons conservé qu'un ensemble de métadonnées minimal pour décrire les évolutions et faciliter l'échange de mises à jour dans une infrastructure militaire.

Le profil est présenté par le biais de schémas UML dont la notation est résumée dans les figures 3.12 et 3.13. Le dictionnaire de données contenant l'intégralité des classes utilisées dans le profil MUMSDI est fourni en annexe B. Dans les schémas conceptuels, les extensions d'entités sont assurées par la création de nouvelles classes dont le nom est préfixé par MU_ et qui héritent d'une classe définie dans ISO 19115. Ces nouvelles classes sont représentées sur fond orange. Les classes du modèle UML définies dans ISO 19115 mais non utilisées dans le profil sont représentées sur fond blanc. Les classes définies dans ISO 19115 et utilisées dans le profil sont représentées sur fond jaune.

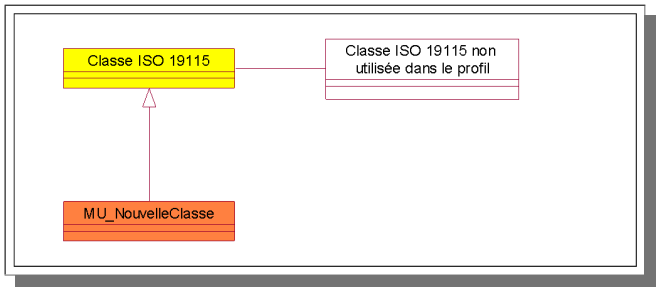

Figure 3.12 – Extension et désactivation des classes ISO 19115 dans MUMSDI

Les restrictions sont spécifiées avec les cardinalités et représentées dans des étiquettes attachées aux classes ou aux relations concernées, sur fond orange.

Nous présentons dans la suite de cette partie, l'ensemble de métadonnées du profil MUMSDI. Nous voyons dans un premier temps les métadonnées de qualité que nous décrivons en détail car elles constituent le coeur du profil et sont de ce fait indispensables. Puis nous présentons les extensions et restrictions principales que nous avons apportées à la norme.

Métadonnées de qualité dans le profil MUMSDI

Les métadonnées de qualité définies par l'ISO 19115 sont utilisées pour décrire la qualité des données du point de vue du producteur. Parfois, l'information de qualité

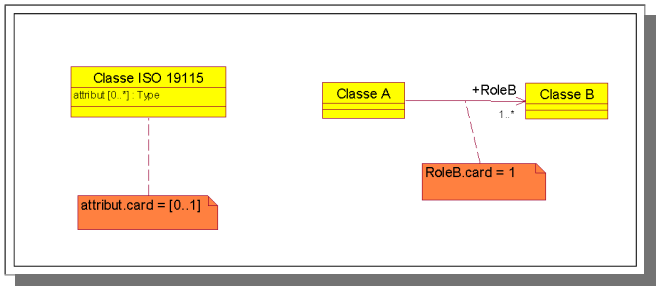

Figure 3.13 – Contraintes sur les cardinalités des attributs et des rôles dans MUMSDI

fournie par le producteur n'est pas satisfaisante pour l'utilisateur final car celui ci a besoin de données qu'il va exploiter dans un contexte particulier et non global. C'est le cas lors de missions militaires où les unités localisées sur les sites différents n'ont pas les mêmes besoins et peuvent utiliser des données à différents niveaux de détail ou avec des qualités distinctes. Les métadonnées de qualité produites par le producteur ne sont donc pas systématiquement adéquates pour toutes les unités et le point de vue de l'utilisateur final doit être pris en compte dans les éléments de métadonnées. Nous proposons dans notre profil de métadonnées, d'ajouter des éléments de qualité afin de prendre en considération le point de vue de l'utilisateur et de restreindre certains éléments de la norme qui ne sont pas utiles dans un contexte de mise à jour de données géographiques vectorielles militaires.

La figure 3.14 montre une vue globale des informations de qualité dans notre profil MUMSDI.

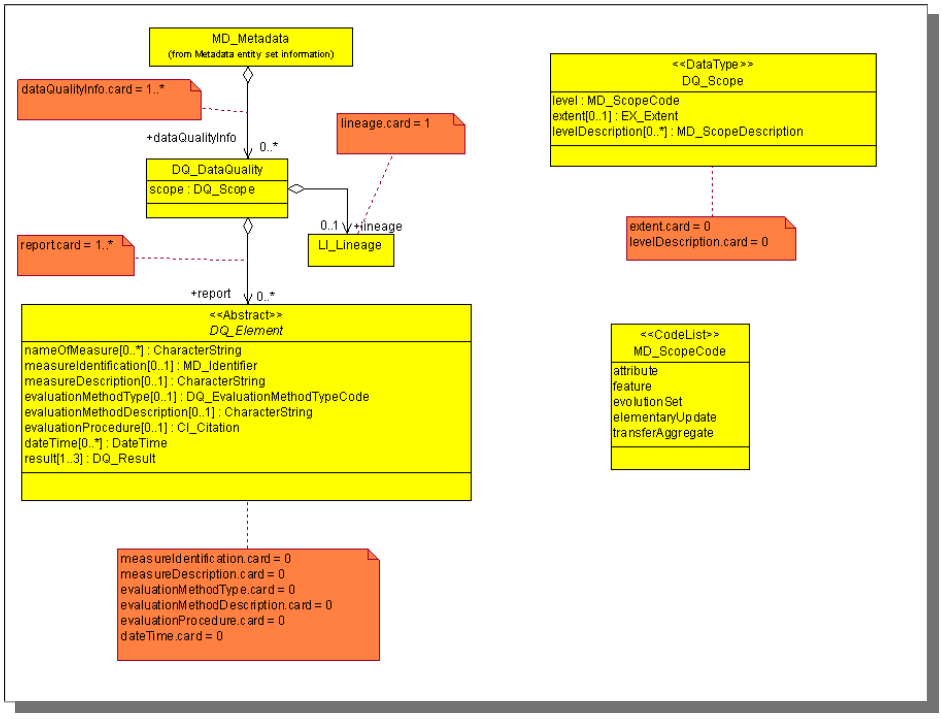

Figure 3.14 – Information de qualité dans le profil MUMSDI

La première modification que nous avons apportée concerne la cardinalité du rôle `dataQualityInfo` que nous avons contraint car nous pensons que pour évaluer correctement les évolutions, **il faut rendre obligatoire l'information de qualité**. De même, nous avons modifié les cardinalités des rôles `lineage` et `report` de la section `dataQualityInfo` afin de les rendre également obligatoires dans le profil. Ces contraintes révéleront tout leur intérêt lors du traitement des évolutions conflictuelles. En effet, lorsqu'aucune information de qualité n'est disponible, le processus de réconciliation ne peut se dérouler automatiquement et une interaction avec l'utilisateur est alors requise. Et même dans ce cas, le résultat de cette analyse n'est pas non plus optimal car soumis à la seule interprétation de l'utilisateur qui doit effectuer un choix sans connaître l'information de qualité, ce qui peut par conséquent entraîner des erreurs dues à une mauvaise appréciation.

Nous avons aussi modifié la portée des éléments de qualité (attribut `level` de la classe `DQ_Scope`). Cette information est donnée dans la norme par une liste (`MD_ScopeCode`) qui contient un certain nombre d'éléments permettant de décrire le niveau sur lequel l'information va s'appliquer. Nous avons modifié cette liste dans le profil MUMSDI afin de **prendre en compte uniquement les éléments concernant les informations de qualité s'appliquant sur des évolutions**. L'ensemble des valeurs de `MD_ScopeCode` définies dans le standard ISO 19115 a donc été supprimé à l'exception de :

- `attribute` qui permet d'appliquer les informations sur une valeur d'attribut
- `feature` qui permet d'appliquer les informations sur une instance d'un objet géographique

Par ailleurs, d'autres valeurs ont été ajoutées afin de prendre en considération les évolutions :

- `evolutionSet` permet d'appliquer les informations sur un ensemble d'évolutions saisies sur une même couche thématique
- `elementaryUpdate` permet d'appliquer les informations sur une évolution élémentaire
- `tranferAggregate` issu du format METAFOR et qui permet d'appliquer les informations sur un produit dans son intégralité

Nous avons également supprimé un certain nombre d'attributs que nous avons considérés comme non utiles dans un contexte de mise à jour. La principale raison de cette limitation vient du fait que plus il y a d'éléments de métadonnées correctement remplis, moindre est la difficulté à intégrer les évolutions. Or, nous avons vu précédemment qu'une des difficultés liée à l'utilisation de métadonnées réside dans le nombre important d'éléments de métadonnées à renseigner, conduisant souvent le producteur à ne remplir que les champs nécessaires voire à ne rien documenter. Nous pensons donc que **moins il y aura d'éléments superflus** dans le profil MUMSDI, **plus facile sera leur saisie et meilleure sera leur interprétation**. Nous avons par exemple, supprimé l'attribut `levelDescription` de la classe `DQ_Scope` et les attributs descriptifs de la classe `DQ_Element` qui n'apportent aucune valeur ajoutée permettant de mieux apprécier la qualité des évolutions. Nous avons également retiré l'attribut `extent` de la classe `DQ_Scope` qui n'a aucune utilité dans notre contexte car les informations de qualité ont une étendue qui coïncide

exactement avec l'étendue du produit ou de l'ensemble d'évolutions concernés.

Les informations de généalogie sont, quant à elles, accessibles depuis la classe `LI_Lineage` qui est agrégée à `DQ_DataQuality` et optionnelle dans la norme ISO 19115. Les informations disponibles depuis cette classe concernent la description des sources (`LI_Source`) et la description des phases du processus de production (`LI_ProcessStep`). Dans MUMSDI, **les informations de généalogie sont rendues obligatoires et au moins une source doit être décrite** (Cf. figure 3.15). Cette contrainte permet de connaître la provenance des évolutions notamment via l'attribut `sourceCitation` que nous rendons également obligatoire. Cet élément spécifie d'une part les données de référence ayant été utilisées comme support à la mise à jour et renseigne d'autre part sur l'auteur des mises à jour permettant ainsi d'estimer la fiabilité des évolutions (nous rappelons que les acteurs et leurs rôles sont connus dans l'infrastructure).
L'auteur des mises à jour a de surcroît la possibilité de fournir des informations complémentaires sur les traitements qui ont permis d'obtenir les évolutions via la classe `LI_ProcessStep`. Par exemple, la description des traitements effectués sur des évolutions issues de la généralisation des données d'un jeu de données ayant une échelle plus grande afin de combler le manque d'informations à l'échelle utilisée.

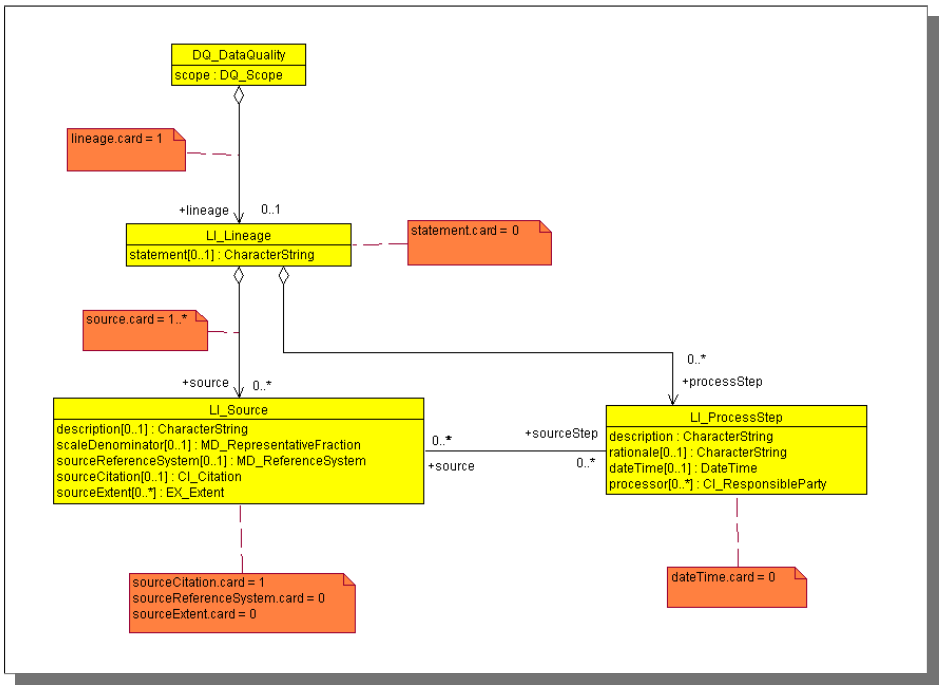

Figure 3.15 – Expression des informations de généalogie dans le profil MUMSDI

Par ailleurs, l'ensemble des informations de qualité peut être composé d'un ensemble de mesures qualité. Cet ensemble est accessible depuis la classe `DQ_Element` dont seuls le nom (attribut `nameOfMesure`) et le résultat (attribut `result`) des mesures qualités sont spécifiés dans le profil MUMSDI (Cf. figure 3.16). `DQ_Element` est une classe abstraite, donc non instanciable et sous typée en plusieurs sous-classes représentant des mesures de qualité élémentaires. Certaines de ces sous-classes n'ont

pas d'intérêt dans notre contexte et ont par conséquent été désactivées.
C'est le cas notamment de `DQ_CompletenessCommission` qui a été initialement créée pour définir les données en excès dans un jeu de données (par exemple, un bâtiment a été reporté deux fois), ce qui a priori n'arrive jamais dans un ensemble d'évolutions ayant été saisi par le même auteur.
Nous avons ensuite désactivé la classe abstraite `DQ_LogicalConsistency` et ses classes dérivées et la classe `DQ_ThematicClassificationCorrectness` car nous nous interessons dans ce travail de recherche, uniquement à la cohérence des données et non à la cohérence des modèles ou à la classification.
Nous avons également exclus la classe abstraite `DQ_TemporalAccuracy` car nous supposons que la précision temporelle exacte dépend de la date de saisie des évolutions.
Enfin, nous avons désactivé la classe `DQ_GriddedDataPositionalAccuracy` car les données et évolutions sont exclusivement de type vectoriel dans notre étude.

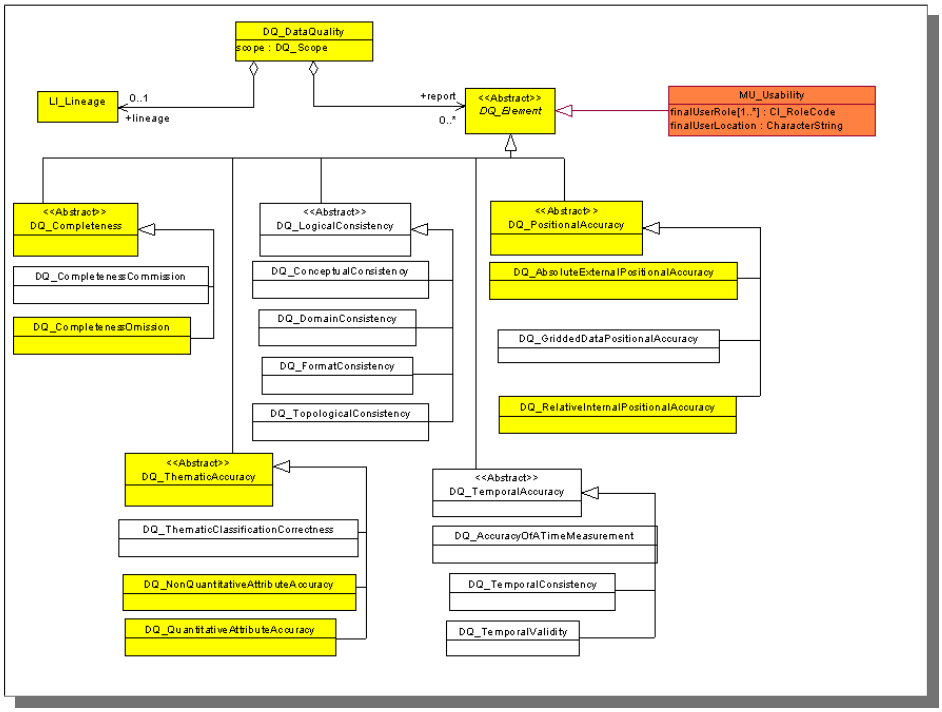

Figure 3.16 – Eléments de qualité dans le profil MUMSDI

Par ailleurs, nous avons ajouté une classe `MU_Usability` afin de percevoir la capacité des évolutions à répondre à l'emploi que l'utilisateur pourrait en faire. En effet, nous avons vu que le besoin en matière de données et d'évolutions n'est pas le même et dépend de la localisation (quartier général ou terrain d'action) et du rôle (producteurs, opérationnels ou utilisateurs) des acteurs. Nous avons par conséquent ajouté cet élément afin qu'il renseigne sur **le degré de conformité des évolutions avec l'usage qui peut en être fait pour un type donné d'utilisateur**. Cette classe possède deux attributs permettant de renseigner le type d'utilisateur concerné par la mesure qualité (attribut `finalUserRole`) et le site sur lequel cet utilisateur se trouve (attribut `finalUserLocation`). Le résultat de cette mesure est de type

`ConformanceResult` qui signale si la mesure est conforme ou non (attribut `pass`).

Enfin, l'expression du résultat de la qualité dans MUMSDI est montrée dans la figure 3.17. Les résultats disponibles dans le standard (`DQ_QuantitativeResult` et `DQ_ConformanceResult`) peuvent être facilement fournis par un producteur mais difficilement par les autres utilisateurs (par exemple les opérationnels déployés sur le terrain d'action). En effet, ces utilisateurs particuliers n'ont pas les moyens techniques d'évaluer précisément les évolutions et doivent apprécier par eux mêmes la qualité des mises à jour qu'ils ont effectuées. Nous pensons donc que les résultats quantitatifs ne sont pas suffisants pour décrire les informations de qualité des évolutions et nous avons donc ajouté des éléments qualitatifs afin de juger la qualité des évolutions d'une manière plus souple. Une classe `MU_QualitativeResult` a été créée à cet effet. Elle permet d'une part de dire rapidement si les informations non spatiales attachées aux évolutions élémentaires ont été correctement documentées (attribut `documentation` qui peut prendre les valeurs non documenté, mal documenté, à moitié documenté, bien documenté et totalement documenté) et d'autre part quels sont les types d'erreurs que peuvent contenir les mises à jour (attribut `errorType` de type MU_Error qui possède trois attributs permettant de définir l'erreur probable comme étant une erreur de type géométrique (précision, exactitude), une erreur sur les attributs ou une erreur de type topologique (chevauchement, mauvaises connexions)).

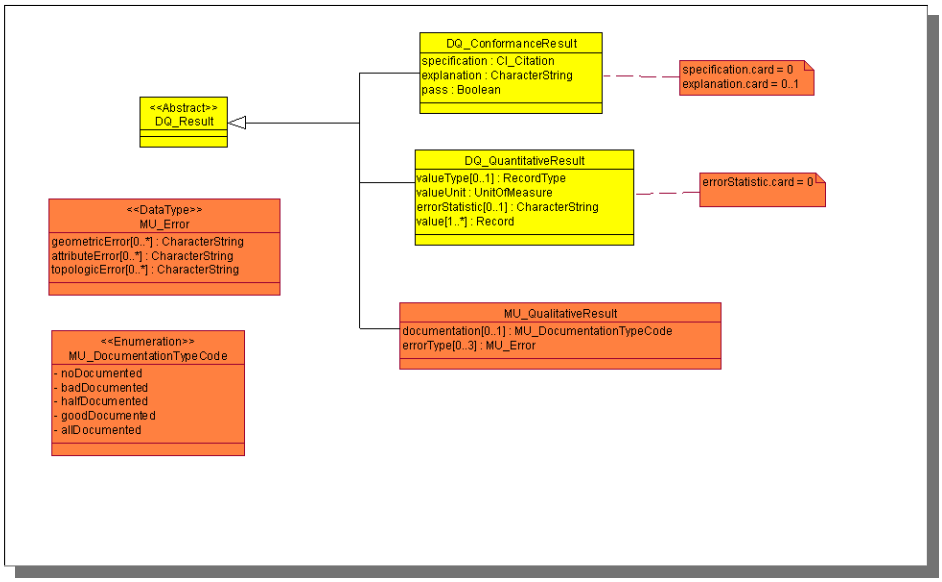

Figure 3.17 – Expression du résultat de la qualité dans le profil MUMSDI

Autres extensions et restrictions apportées au profil MUMSDI

L'extension première que nous avons apportée concerne la création d'un nouveau package qui contiendra toutes les métadonnées relatives au profil MUMSDI. Toutes les métadonnées définies dans ce package porteront l'extension `MU_`.

Nous avons ensuite ajouté deux nouvelles classes de métadonnées comme nous l'avons vu dans le paragraphe précédent consacré aux métadonnées de qualité. La première classe (`MU_Usability`), permet de fournir les renseignements sur l'utilisabilité des évolutions. La seconde (`MU_QualitativeResult`) permet de fournir des informations sur la qualité relative des mises à jour.

Les restrictions que nous avons apportées dans le profil MUMSDI par rapport au standard concernent surtout le champ d'application des entités et éléments de métadonnées dont nous avons réduit la cardinalité et le nombre d'éléments dans les listes de code. La figure 3.18 montre les listes de code (`CI_RoleCode`) et (`CI_DateTypeCode`) que nous avons modifiées.
Dans la liste `CI_RoleCode`, nous avons ajouté les acteurs spécifiques à une mission militaire et avons supprimé tous les rôles qui ne sont pas utilisés dans l'infrastructure militaire.
Dans `CI_DateTypeCode`, nous avons ajouté une information permettant d'identifier un événement de type mise à jour. En effet, dans le noyau ISO, le champ date est obligatoire mais les types prédéfinis dans la norme (`création, publication et révision`) ne permettent pas de spécifier qu'il s'agit d'une mise à jour. Nous avons donc créé un élément `update` que nous avons ajouté à cette liste de code.

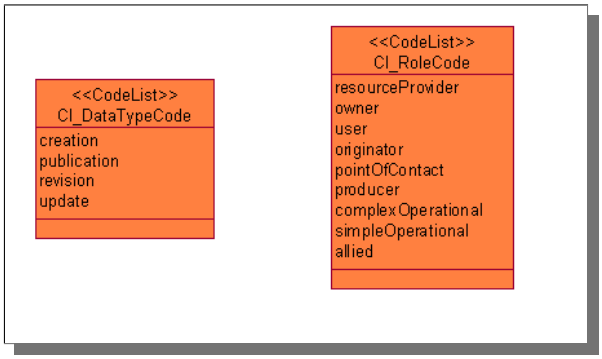

Figure 3.18 – Listes de code étendues dans le profil MUMSDI

Enfin, nous avons limité la portée du type de données `EX_Extent` à la seule classe `EX_GeographicExtent` et à ces deux classe dérivées `EX_GeographicPolygon` et `EX_GeographicBoundingBox` afin de pouvoir stocker les informations sur la zone concernée par les mises à jour. Les autres classes agrégées à `EX_Extent` n'ayant pas d'intérêts pour notre étude.

Finalement, la liste détaillée des métadonnées du standard ISO 19115 que nous avons maintenues dans notre profil MUMSDI est donnée en annexe B.

3.3.4 Prise en compte des besoins des utilisateurs : Métadonnées sur les acteurs

Dans cette partie, nous identifions les métadonnées des acteurs afin de définir formellement les besoins et les contraintes des utilisateurs de l'infrastructure.

Les utilisateurs de l'infrastructure militaire n'ont pas tous les mêmes besoins en matière d'évolutions. Ces besoins dépendent de plusieurs facteurs tels que le rôle de l'utilisateur, le site sur lequel ce dernier se situe... Par exemple, un producteur situé au quartier général a plutôt besoin de données (et donc d'évolutions) fiables car celles-ci peuvent servir à la planification des plans de vols. Des erreurs dans les données pourraient avoir des répercussions catastrophiques. Le choix des évolutions à intégrer dans leur jeu de données doit donc être dirigé par cette contrainte de précision.
En revanche, les opérationnels sur le théâtre des opérations souhaitent plutôt obtenir un maximum de données rapidement quel que soit leur qualité, car leur préoccupation première est de déployer les troupes sur le terrain. Ces utilisateurs préfèrent intégrer des évolutions de qualité moindre (topologie et géométrie incertaine, valeurs d'attributs non renseignées ...) plutôt que ne pas avoir d'informations sur la zone concernée. Le choix des évolutions à intégrer devra donc prendre en considération ces critères.
Par ailleurs, un certain nombre de contraintes de cohérence liées au jeu de données utilisé et aux moyens techniques disponibles doivent également être définies pour chaque acteur de la mission. En effet, un acteur de l'infrastructure dispose d'un jeu de données certes dérivé d'un unique jeu de référence, mais qui a peut être été transformé pour pouvoir être exploité par le système de l'utilisateur.

Les besoins et contraintes des utilisateurs dépendent donc du rôle et du site sur lequel les acteurs sont déployés mais également des moyens techniques mis en place pour la mission. Au début de la mission, les contraintes de cohérence à respecter et les besoins de chaque acteur sont spécifiés formellement. L'ensemble de ces informations constitue les métadonnées des acteurs. Les besoins ne sont pas figés et peuvent évoluer au cours de la mission. En revanche, les contraintes sont imposées dès le début et ne peuvent plus être modifiées.

Nous avons formalisé l'ensemble des métadonnées attachées aux acteurs en nous inspirant de la même grammaire BNF étendue que celle que nous avons utilisée pour décrire les évolutions. Les métadonnées des acteurs de l'infrastructure sont organisées de la manière suivante :

<metadonnees_acteurs> : := <ens_besoins_utilisateur> <ens_contraintes_coherence>
<ens_besoins_utilisateur> : := <zone> <date> <couche_thematique*> <type_evol*>
<precision_geometrique*> <precision_attributaire*> <exhaustivite*> <fiabilité*>
<ens_contraintes_coherence> : := <contrainte_spatiale*> <contrainte_attributaire*>
<contrainte_contextuelle*>
<contrainte_spatiale> : := <precision_geometrique> | <resolution>
<contrainte_attributaire> : := <precision_attributaire_qualitative> |
<precision_attributaire_quantitative>
<contrainte_contextuelle> : := <source*> | <processus*> | <zone> | <date> |
<exhaustivite*>
<zone> : := <coordonnees_no> <coordonnees_so> <coordonnees_ne> <coordonnees_se>
<coordonnees_no> : := <coordonnees>
<coordonnees_so> : := <coordonnees>
<coordonnees_ne> : := <coordonnees>
<coordonnees_se> : := <coordonnees>
<date> : := DATE
<coordonnees> : := {ABSCISSE, ORDONNEE}+
<couche_thematique : := NOM_THEME
<type_evol> : := CREATION | SUPPRESSION | MODIFICATION_GEOMETRIQUE |
MODIFICATION_ATTRIBUTAIRE | MODIFICATION_MIXTE
<precision_geometrique> : := NOMBRE
<resolution> : := NOMBRE
<fiabilité> : := POURCENTAGE
<precision_attributaire_qualitative> : := POURCENTAGE
<precision_attributaire_quantitative> : := NOMBRE
<exhaustivite> : := POURCENTAGE
<source> : := TYPE_SOURCE
<processus> : := NATURE_PROCESSUS

Les métadonnées d'un acteur sont donc constituées d'un ensemble recensant les besoins de l'utilisateur et d'un ensemble de contraintes de cohérence qui dépendent du jeu de données utilisé et des moyens techniques disponibles pour cet acteur.

L'ensemble des besoins des utilisateurs est défini par :
- Une zone définissant l'étendue spatiale maximale (coordonnées du rectangle englobant la zone),
- Une date d'actualité minimale (la date la plus ancienne qui puisse être acceptée)
- Une ou plusieurs couches thématiques (hydrographie, transport, ...)
- Un ou plusieurs types d'évolutions (créations, suppressions, modifications)
- Une précision géométrique minimale (ce critère est défini en fonction des thèmes et types d'évolutions souhaités par l'utilisateur)
- Une précision attributaire minimale (ce critère est défini en fonction des thèmes et types d'évolutions souhaités par l'utilisateur)
- Une exhaustivité minimale (ce critère est défini en fonction des thèmes et types d'évolutions souhaités par l'utilisateur)
- Une fiabilité qui correspond au pourcentage d'erreurs acceptées pour chaque

type d'erreur (géométrique, attributaire et/ou topologique)

La liste des contraintes de cohérence est définie par :
- Des contraintes spatiales
  - Une précision géométrique qui correspond à la précision minimale autorisée dans le jeu de données. Elle est exprimée en mètre.
  - Une résolution minimale qui correspond à la taille minimale autorisée pour le plus petit objet représentable. Elle dépend de l'échelle du jeu de données.
- Des contraintes attributaires
  - Une précision attributaire qualitative qui correspond au nombre maximal d'objets autorisé dont la valeur des attributs non spatiaux a été mal renseignée dans le jeu de données. Elle est exprimée en pourcentage.
  - Une précision attributaire quantitative qui correspond à la précision minimale autorisée pour les valeurs d'attributs. Elle est exprimée en pourcentage.
- De contraintes de contexte
  - Une ou plusieurs sources de données qui correspondent aux types de sources autorisés pour faire évoluer le jeu de données (jeu de données de référence, source extérieure à l'infrastructure ...)
  - Un ou plusieurs acteurs qui correspondent aux rôles des acteurs dont on peut accepter les évolutions.
  - Un ou plusieurs processus qui correspondent à la nature des processus autorisés pour faire évoluer le jeu de données (généralisation, levé terrain, analyse de photo aérienne ou d'images satellites, scannage de cartes papiers ...)
  - Une exhaustivité qui correspond au nombre minimal d'objets appartenant à un thème souhaité. Elle est exprimée en pourcentage.
  - La date d'actualité du jeu de données. Toutes les évolutions effectuées avant cette date ne devront pas être considérées.
  - L'étendue maximale du jeu de données. Toutes les évolutions située hors de cette zone ne devront pas être considérées.

Les contraintes de cohérence sont organisées dans un fichier dont la structure est la suivante :

```
Contraintes Nom_JDD
  Pour chaque Objet Nom_Objet
    Contraintes_Geometriques
    Contraintes_Attributaires
   Contraintes_Contextuelles
 Fin Objet
Fin Contraintes
```

Les besoins des acteurs sont organisés dans un fichier dont la structure est la suivante :

```
Besoins Nom_utilisateur
  Zone_Spatiale_Max
  Date_Actualite_Min
  Exhaustivite_Min
  Précision_Geometrique_Min
  Précision_Semantique_Min
  Fiabilité
  Pour chaque CoucheThematique Nom_Couche_Thematique
    Exhaustivité
    Précision_Geometrique
    Précision_Semantique
    Fiabilité
  Pour chaque TypeEvol Nom_TypeEvol
    Exhaustivité
    Précision_Geometrique
    Précision_Semantique
  Fin TypeEvol
 Fin CoucheThematique
Fin Besoins
```

3.3.5 Relations entre les métadonnées et le modèle DAE

Dans cette section, nous présentons les relations qui peuvent être établies entre les métadonnées et le modèle « Données, Acteurs, Évolutions » que nous avons défini au paragraphe 3.1. La figure 3.19 montre les relations entre le modèle DAE et les métadonnées, en particulier le type des métadonnées attachées à chaque entité du modèle. Nous examinons dans un premier temps, comment les métadonnées sont liées aux données dans la norme ISO 19115. Ensuite, nous voyons de quelle manière les données sont prises en compte dans le format METAFOR. Enfin, nous exposons notre point de vue sur la prise en charge des évolutions dans le profil MUMSDI.

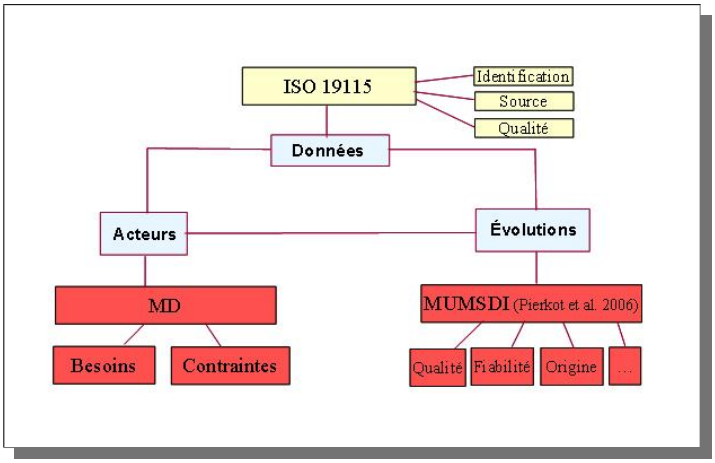

Figure 3.19 – Relations entre les métadonnées et le modèle DAE

Relations entre données et métadonnées dans la norme ISO 19115

Le lien entre données et métadonnées défini dans l'ISO 19115 est présenté dans la figure 3.20. Dans la norme ISO 19115, un jeu de données est représenté par la classe `DS_Dataset` et contient des métadonnées. Les jeux de données peuvent être agrégés en un ensemble de jeux de données via la classe `DS_Aggregate`. Chaque ensemble de jeux de données possède également des métadonnées.
Un jeu de données ou un ensemble de jeux de données est obligatoirement associé à un ensemble de métadonnées, mais il arrive parfois qu'il soit attaché à des ensembles de métadonnées reliés à d'autres ressources composant l'ensemble des jeux de données ou le jeu de données (un objet géographique par exemple). Le niveau hiérarchique des métadonnées défini par la valeur de l'attribut `hierarchyLevel` de la classe `MD_Metadata` permet de préciser si les métadonnées se rapportent au jeu de données (valeur `dataset` de la liste de code `MD_ScopeCode`) ou à une autre ressource composant le jeu de données.

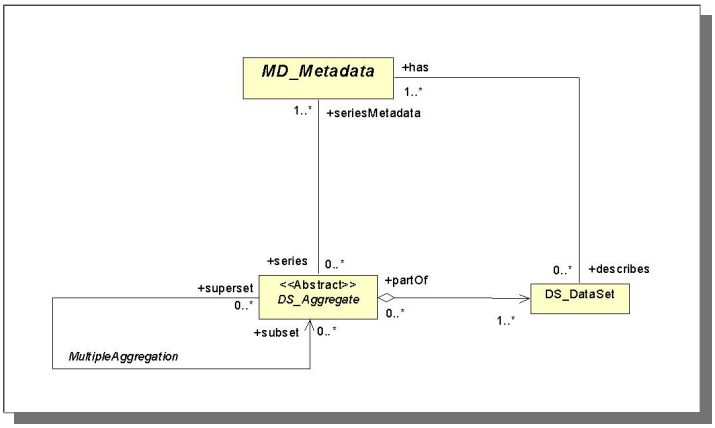

Figure 3.20 – Principe de l'échange des données dans ISO 19115

Nous constatons qu'aucune disposition n'a été prise dans la norme ISO 19115 pour prendre en compte l'échange des évolutions.

Prise en compte des données dans le profil METAFOR

Dans METAFOR, l'élément de base permettant tout échange ou stockage de données géographiques est un produit. Les produits se distinguent sous deux formes : les produits unitaires et les ensembles de produits. Par convention un produit unitaire n'appartient qu'à un seul ensemble de produits.

Les produits unitaires sont constitués :
- D'un ensemble de métadonnées permettant d'accéder aux données et décrivant le produit. L'implémentation se fait au moyen d'un ou plusieurs fichiers au format XML utilisant le profil d'implémentation ISO 19115 défini pour la gamme DNG3D.
- D'un ensemble de données constituant l'objet lui même.
- Des informations complémentaires facultatives.

Les ensembles de produits sont constitués :
- D'un ou plusieurs produits unitaires regroupés pour faciliter leur échange ou leur stockage.
- D'un ensemble d'informations pour accéder aux produits unitaires
- D'un ensemble de métadonnées décrivant l'ensemble de produits. L'implémentation se fait au biais d'un ou plusieurs fichiers au format XML utilisant le profil d'implémentation ISO 19115 défini pour la gamme DNG3D.
- Des informations complémentaires facultatives.

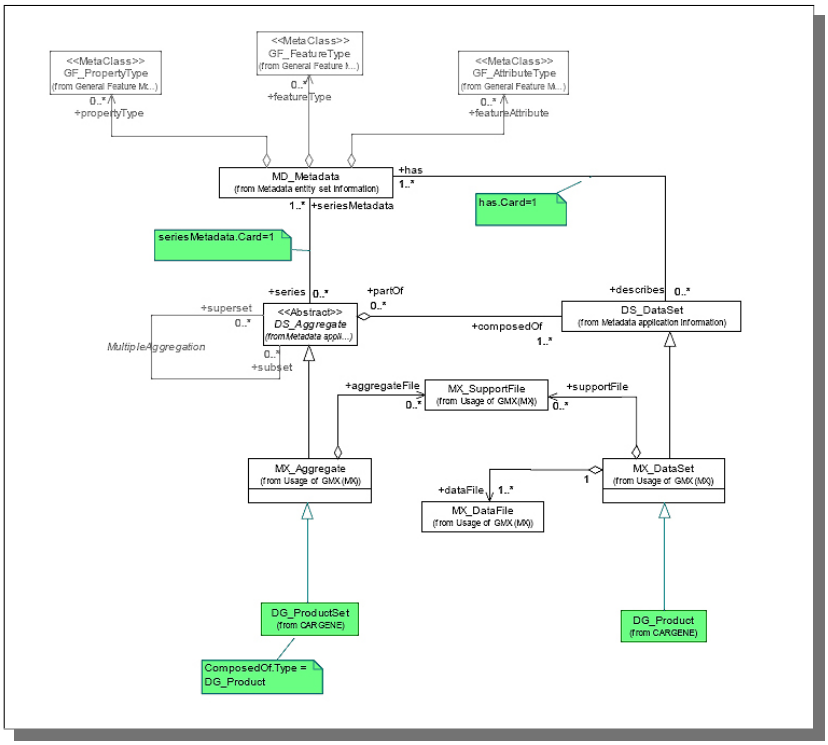

Figure 3.21 – Relation entre les produits et leurs métadonnées dans METAFOR

La figure 3.21 montre comment un produit DNG3D est assimilé à un jeu de données de la norme ISO 19115. La classe `DG_Product` correspondante, dérivée de la classe `MX_Dataset` de ISO 19115 est l'élément racine des fichiers de métadonnées d'un produit. Un ensemble de produits DNG3D est assimilé à un agrégat de jeux de données de la norme ISO 19115. La classe `DG_ProductSet` correspondante, dérivée de la classe `MX_Aggregate` de ISO 19115, est l'élément racine des fichiers de métadonnées d'un ensemble de produits.

Nous constatons qu'aucune disposition n'a été prise dans METAFOR pour prendre en compte l'échange des évolutions.

Prise en compte des évolutions dans le profil MUMSDI

Le contexte d'emploi des métadonnées dans le cadre de l'échange d'évolutions n'est donc pas pris en charge dans la norme ISO 19115, ni dans le format militaire

METAFOR. Nous proposons de définir celui ci dans le profil MUMSDI.

Deux orientations sont possibles pour rattacher les évolutions avec le modèle de métadonnées. Soit nous procédons par extension du format de fichier METAFOR ou soit nous étendons directement la norme ISO 19115. Le choix dépend de la manière dont les évolutions sont perçues.
Si l'on considère que les changements sont des données et que les indications de changement sont les métadonnées alors les ensembles d'évolutions sont des produits au sens CARGENE. Dans ce cas nous devons étendre la classe `DG_Product` définie dans METAFOR avec une classe `MU_EvolProduct` définissant particulièrement les ensembles d'évolutions (Cf. figure 3.22).

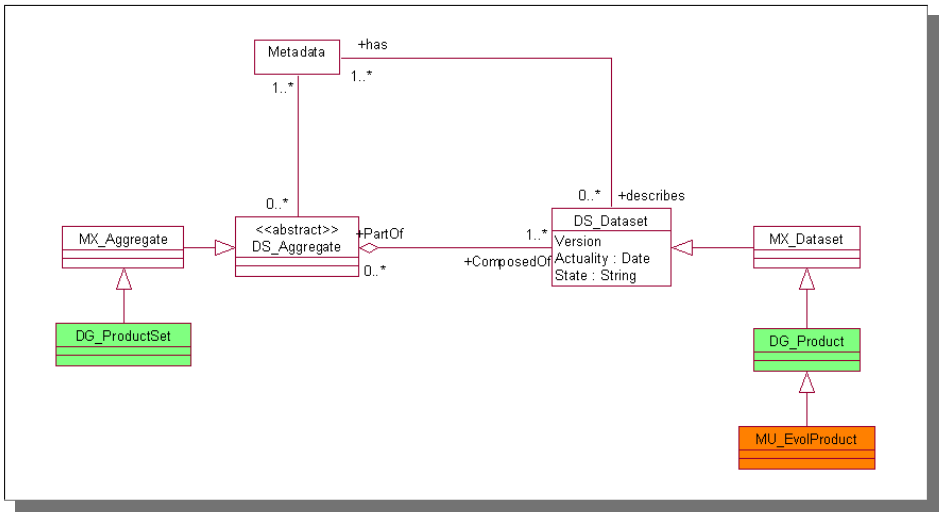

Figure 3.22 – Relation entre les évolutions et leurs métadonnées : Extension de Metafor

En revanche, si l'on considère que les évolutions ne véhiculent aucune donnée proprement dite, mais uniquement les changements intervenus, alors ils ne peuvent être considérés comme des produits au sens DNG3D. Il faut dans ce cas créer une nouvelle classe `MU_EvolutionSet` dérivée de `DS_DataSet` afin de prendre en compte un ensemble d'évolutions concernant une même couche thématique, et une classe `MU_ProductSet` dérivée de `DS_Aggregate` afin de considérer les produits regroupant l'ensemble de toutes les évolutions (Cf. figure 3.23).

Nous avons défini formellement le format des fichiers d'évolutions qui peuvent être transmis dans une infrastructure militaire au paragraphe 3.2.2. Ces fichiers contiennent uniquement les changements intervenus entre deux versions d'un jeu de données car dans l'infrastructure, un jeu de données de référence est à l'origine de tous les jeux de données utilisés par les acteurs sur les différents sites (fourni par le producteur situé au quartier général en début de mission). Il n'est donc pas nécessaire de fournir les données avec les évolutions lors du transfert des mises à jour car nous pouvons retrouver celles-ci à l'aide de traitements appropriés (recherche d'identifiant ou appariement géométrique). Nous choisissons donc d'utiliser la deuxième solution pour relier les métadonnées aux évolutions.

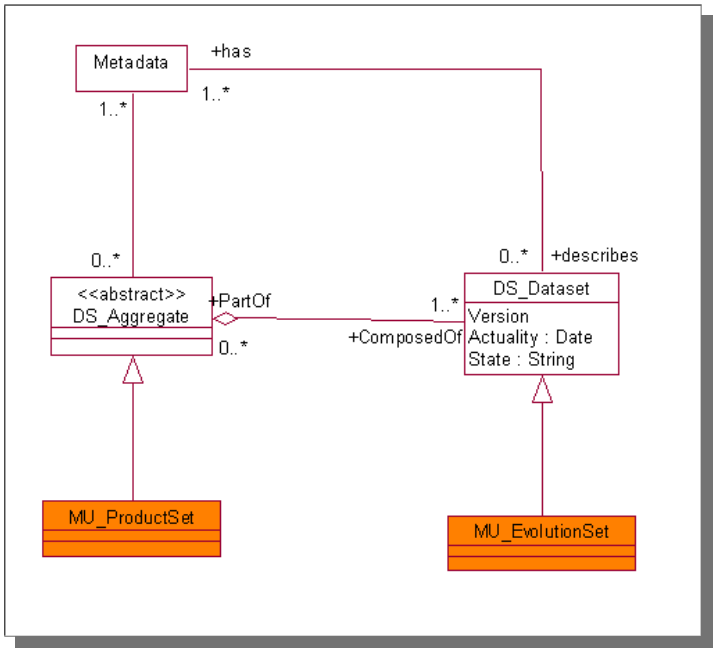

Figure 3.23 – Relation entre les évolutions et leurs métadonnées : Extension directe de ISO 19115

Comme dans le standard, le niveau hiérarchique des métadonnées définit par la valeur de l'attribut `hierarchyLevel` de la classe `MD_Metadata` permet de préciser si les métadonnées se rapportent à un ensemble d'évolutions (valeur `evolutionSet` de la liste de code `MD_ScopeCode`) ou à une autre ressource composant le jeu de données.

3.4 Stratégie d'intégration des mises à jour multi-sources

La première partie de ce chapitre a été consacrée à la mise en place d'une base pour aider la gestion des évolutions au sein d'une infrastructure militaire. En particulier, nous avons dans un premier temps créé une infrastructure militaire dans laquelle nous avons spécifié les rôles des différents acteurs ainsi que les relations entre les données, les acteurs et les évolutions à travers un modèle (le modèle Données Acteurs Evolutions). Puis nous avons défini une politique de gestion des mises à jour en spécifiant le format des évolutions et la structure des ensembles d'évolutions qui peuvent être échangés dans l'infrastructure. Ensuite, nous avons défini un profil de métadonnées permettant de documenter le transfert des évolutions et nous avons déterminé les besoins des acteurs d'une mission militaire. Enfin nous avons établi les liens entre les métadonnées et les entités du modèle DAE.

Avec les hypothèses prises dans l'infrastructure militaire et grâce à la définition des métadonnées utiles, nous sommes maintenant en mesure de proposer une stratégie d'intégration des évolutions qui va permettre de résoudre les différents

problèmes dus à la mise à jour multi-sources d'un jeu de données géographiques vectoriel. Cette stratégie est basée sur trois points stratégiques : en premier lieu, nous effectuons un filtrage des évolutions afin d'exclure les évolutions qui ne sont pas pertinentes pour l'utilisateur final. Ensuite, nous faisons un contrôle de la cohérence afin de détecter et de traiter les éventuels conflits entre les nombreuses évolutions provenant de sources distinctes et entre les évolutions et les données de l'utilisateur. Enfin nous réalisons des sessions de mises à jour afin d'intégrer les évolutions pertinentes et non conflictuelles dans le jeu de données de l'utilisateur.

Cette partie se décompose de la façon suivante : premièrement, nous présentons l'approche générale de la stratégie d'intégration que nous avons établie pour résoudre les problèmes liés à la mise à jour de données géographiques répliquées sur des sites distincts. Puis, nous donnons une piste pour la vérification de la pertinence des évolutions. Ensuite, nous détaillons le processus de vérification de la cohérence et en particulier nous nous attardons sur les mécanismes de détection de la concurrence et sur les protocoles de réconciliation mis en place pour résoudre les conflits. Enfin, nous discutons de l'intérêt d'utiliser des sessions de mise à jour.

3.4.1 Stratégie globale d'intégration des évolutions

Nous voyons dans ce paragraphe la stratégie globale d'intégration des évolutions qui va permettre de mettre à jour le jeu de données d'un utilisateur par des évolutions provenant de sources multiples. Nous détaillons dans un premier temps l'approche générale de la stratégie en décrivant chacune des différentes étapes mises en place, puis nous montrons l'agencement de ces différentes étapes dans la stratégie globale, enfin nous formalisons ces enchaînements en les spécifiant selon le modèle « Événements-Conditions-Actions » .

Approche générale

La figure 3.24 présente l'approche générale de la stratégie d'intégration des évolutions. La stratégie d'intégration s'effectue en trois étapes et s'applique sur le jeu de données d'un utilisateur de l'infrastructure pour lequel des ensembles d'évolutions sont proposés. L'exécution des trois phases de la stratégie d'intégration conduit à la mise à jour du jeu de données de l'utilisateur.

La première étape constitue **l'évaluation de la pertinence** des évolutions. Nous effectuons ici un filtrage des évolutions qui ne sont pas utiles pour l'utilisateur. Nous utilisons pour cela les métadonnées associées aux évolutions et les métadonnées associées aux besoins des utilisateurs. A la fin de cette étape, l'ensemble des évolutions proposé à l'intégration ne contient plus que les évolutions pertinentes pour l'utilisation que l'utilisateur veut en faire. Cette étape permet finalement de limiter les ensembles d'évolutions à un nombre restreint et pertinent d'évolutions à intégrer. Nous revenons sur cette partie dans le paragraphe 3.4.2.

La seconde étape concerne **la vérification de la cohérence** dont le but est de détecter et de traiter les éventuels conflits qui peuvent se produire du fait de la prove-

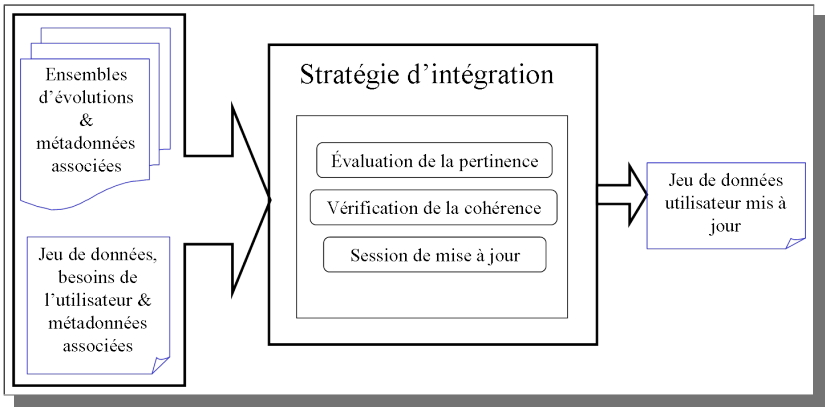

Figure 3.24 – Stratégie d'intégration des évolutions multi-sources dans un jeu de données utilisateur

nance multiple des ensembles d'évolutions. Cette étape qui s'effectue en deux temps est constituée tout d'abord d'un **contrôle de la concurrence** puis d'une phase de **réconciliation des données conflictuelles**. Nous utilisons ici la structure des évolutions et des traitements géométriques pour déterminer les éventuels conflits et nous utilisons ensuite les métadonnées associées aux évolutions, aux données et aux besoins de l'utilisateur pour définir les méthodes de réconciliation. A la fin de cette étape, nous obtenons un ensemble contenant des évolutions pertinentes et non conflictuelles que nous pouvons intégrer dans le jeu de données de l'utilisateur. Nous détaillons cette partie dans le paragraphe 3.4.3.

La troisième et dernière étape est directement liée à la seconde et permet d'intégrer, dans le jeu de données de l'utilisateur, les évolutions préalablement traitées. Nous verrons dans le paragraphe consacré à cette étape (§3.4.4.), les raisons pour lesquelles nous jugeons qu'il est nécessaire d'effectuer **des sessions de mises à jour**.

Enchaînement des étapes de la stratégie d'intégration

La première phase (vérification de la pertinence) s'effectue intégralement dès réception des produits d'évolution. La deuxième et la troisième phase s'effectuent, quant à elles, en exclusions mutuelles et peuvent survenir plusieurs fois pendant l'exécution du processus global de la stratégie d'intégration (Cf. Figure 3.25). En effet, une session de mise à jour peut se produire à tout moment, par exemple, lorsque toutes les évolutions proposées ont été traitées, ou à des dates périodiques qui ont été programmées au début de la mission, ou encore à la demande de l'utilisateur. Par ailleurs, l'ensemble des évolutions non conflictuelles issu de la phase de vérification de la cohérence sert de point d'entrée à la session de mise à jour car il constitue l'ensemble des évolutions à intégrer dans le jeu de données de l'utilisateur. L'étape de vérification de la cohérence doit par conséquent être stoppée dès qu'une session de mise à jour est déclenchée. Dès que la session de mise à jour est terminée, la vérification de la cohérence redémarre alors s'il y a encore des évolutions à traiter ou si de nouvelles évolutions ont été proposées.

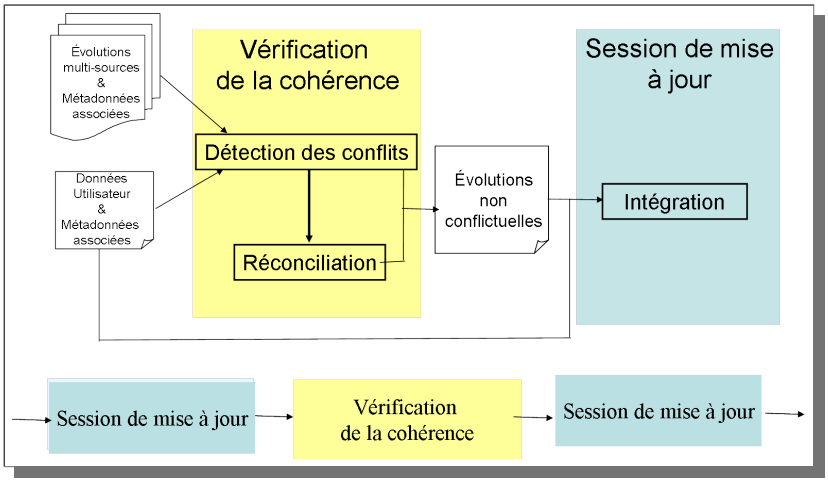

Figure 3.25 – Enchaînement des phases de vérification de la cohérence et de session de mise à jour

Nous utilisons le formalisme Événements-Conditions-Actions (ECA) afin de modéliser les enchaînements de la stratégie globale d'intégration des mises à jour multi-source. Ce formalisme, souvent utilisé dans les bases de données actives permet la définition d'opérations qui sont exécutées automatiquement lorsque certains événements se produisent ou lorsque certaines conditions sont satisfaites. Nous détaillons le principe du formalisme ECA en annexe C.

Dans ce travail, nous avons utilisé la sémantique suivante pour formaliser les enchaînements de la stratégie globale d'intégration des mises à jour multi-source :

- ✓ Les règles sont fonction de l'application que nous voulons mettre en place à savoir la gestion des évolutions multi-sources dans un contexte de réplication de données spatiales lors d'une mission militaire. Les événements, conditions et actions sont donc spécifiques à cette application.
- ✓ Nous avons distingué trois types d'événements primitifs :
  - les **événements de mises à jour** : insertion, suppression ou modification d'un élément,
  - les **événements temporels** : début et fin de règle, date,
  - les **événements externes** : demande de l'utilisateur, appel de fonction ou liste vide.

  Nous rendons possible la combinaison de ces événements afin d'obtenir des événements complexes grâce aux opérateurs logiques ET et OU.
- ✓ Les objets concernés par ces événements sont les objets de l'application c'est-à-dire la liste des évolutions à traiter, la base de données courante de l'utilisateur, une copie de la base de données courante et le journal des évolutions traitées.
- ✓ Les règles sont modélisées indépendamment des données. Deux opérations sont néanmoins permises : `Enable` pour activer une règle et `Disable` pour la désactiver.
- ✓ Le modèle d'exécution est dépendant de la règle dans laquelle il s'applique. Certaines règles peuvent en activer d'autres, le mode d'exécution des règles en

cascade est lui aussi dépendant de la règle dans laquelle il s'applique.

Les deux premières règles que nous spécifions (`Activate_Consistency_Checking` et `Activate_Update_Session`) ont pour but d'activer et de désactiver les étapes de vérification de la cohérence et les sessions de mise à jour. Nous avons vu que ces règles sont en exclusion mutuelle, c'est-à-dire qu'elles ne peuvent être jouées simultanément. Les deux règles possèdent le même modèle d'exécution, à savoir un modèle basé sur des déclenchements immédiats des actions et conditions. Le mode d'exécution en cascade est également le même pour ces deux règles c'est-à-dire que la règle activée s'exécute après la fin de l'exécution de la règle activante.

La règle `Activate_Consistency_Checking` constitue le point de départ de la stratégie d'intégration. Elle est déclenchée lorsqu'une évolution est insérée dans la liste des évolutions à traiter ou par la suite, à la fin d'une session de mise à jour. La règle vérifie en premier lieu que la vérification de la cohérence n'est pas déjà activée et qu'il n'y a pas de session de mise à jour en cours. Si les conditions sont remplies alors une copie de la base de données courante utilisateur est créée et le contrôle de concurrence est alors activé.

```
Règle : Activate_Consistency_Checking
  Evénement : Insert (evol) on EvolList | End (Update_Session)
  Condition : Rule(Update_Session) : disabled &
   Rule (Consistency_Checking) : disabled
  Action :
       Begin
         Create_BDCopy(Vcurrent)
         Enable(Consistency_Checking)
        End
```

Le modèle d'exécution de cette règle est le suivant :
- Exécution Événement-Condition : Immédiat
- Exécution Condition-Action : Immédiat
- L'exécution s'effectue en cascade c'est à dire que la règle `Consistency_Checking` s'exécute après la fin de l'exécution de la règle `Activate_Consistency_Checking`

La règle `Activate_Update_Session` quant à elle, est déclenchée suite à une demande utilisateur ou lorsque la date prévue pour la session de la mise à jour est la date courante ou encore lorsque la liste des évolutions à traiter est vide depuis un certain temps. La condition pour que l'action se déroule est qu'aucune session de mise à jour ne soit déjà en cours. Dans ce cas, la vérification de la cohérence est désactivée et la session de mise à jour peut commencer.

```
Règle : Activate_Upadte_Session
  Evénement : User_Demand | Date_Maj | Empty(EvolList) since time T
  Condition : if Rule(Update_Session) : disabled
  Action :
    Begin
      Disable (Consistency_Checking)
      Enable (Update_Session)
    End
```

Le modèle d'exécution de cette règle est le suivant :
- Exécution Événement-Condition : Immédiat
- Exécution Condition-Action : Immédiat
- L'exécution s'effectue en cascade c'est à dire que la règle `Update_Session` s'exécute après la fin de l'exécution de la règle `Activate_Update_Session`

Les autres règles permettant de formaliser la stratégie d'intégration selon le mécanisme « Événements-Conditions-Actions » sont détaillées dans chacune des parties traitant indépendamment les étapes. Cependant, la figure ci-dessous montre le parallèle entre les principales règles utilisées dans la stratégie d'intégration et les processus définis pour chaque étape.

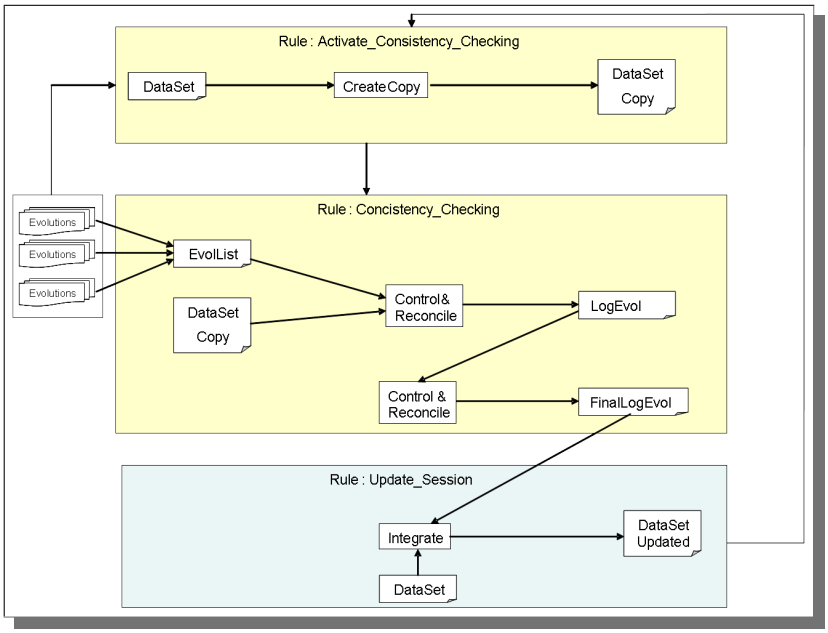

Figure 3.26 – Parallèle entre les règles ECA et les processus utilisés dans la stratégie

Ce schéma montre en particulier les enchaînements entre les différentes règles. Par ailleurs, les règles sont activées soit par un processus (interne ou externe à la règle), soit directement par une autre règle, soit par la terminaison d'une règle, soit à la demande explicite de l'utilisateur. De plus, les objets sur lesquels s'appliquent les règles constituent majoritairement les entrées et sorties des processus utilisés dans les algorithmes. Enfin les processus ne s'exécutent que lorsque les règles sont actives, optimisant ainsi la mémoire qui leur est allouée.

3.4.2 Pertinence des évolutions

Nous abordons dans cette partie la notion de pertinence des évolutions par rapport aux besoins de l'utilisateur final. Nous analysons dans un premier temps les raisons qui justifient de vérifier la pertinence des évolutions, puis nous montrons en quoi cette étape joue un rôle essentiel dans la stratégie d'intégration, enfin nous proposons une première solution basée sur les travaux de [Jeansoulin et Wilson, 2002] et [Vasseur, 2004].

Les sources utilisées dans l'infrastructure sont issues du même jeu de données de référence mais ont évolué (mise à jour et transformation) en fonction des besoins spécifiques des utilisateurs et des moyens techniques dont il dispose. Par exemple, on peut supposer qu'une unité mobile sur le terrain d'action possède des données relatives aux infrastructures routières afin de gérer au mieux les déplacements de ses troupes au sol. Il n'est donc pas nécessaire pour ce cas précis de recueillir des informations concernant d'autres couches thématiques qui dénatureraient finalement le jeu de données de ces acteurs. En revanche, le producteur situé au quartier général possède quant à lui des informations plus générales afin par exemple, de planifier certaines phases délicates de la mission. Cet utilisateur a donc besoin de toutes les informations disponibles auprès des autres acteurs de l'infrastructure pour que son jeu de données soit le plus complet possible.

Par ailleurs, les évolutions peuvent avoir été saisies de diverses façons (sur le terrain, grâce à une image satellite ...), dans des conditions différentes (période de crise, relevé périodique ...) et à différents endroits (sur le terrain d'action, en production ...). Elles sont de ce fait hétérogènes, de qualités différentes et ne concernent pas forcément la même zone de couverture géographique.

Pour toutes ces raisons, un ensemble d'évolutions provenant d'un acteur de l'infrastructure **ne couvre pas à lui seul tous les besoins** de l'utilisateur final. De même, pour un besoin en particulier, un ensemble d'évolutions contient généralement **plus d'informations que ce qui est réellement nécessaire** à l'utilisateur (Cf. Figure 3.27). Finalement, un acteur de l'infrastructure doit récupérer plusieurs ensembles d'évolutions provenant de sources différentes afin de satisfaire complètement son besoin mais doit exclure de ces ensembles les évolutions qui sont en définitive non pertinentes pour l'usage qu'il veut en faire.

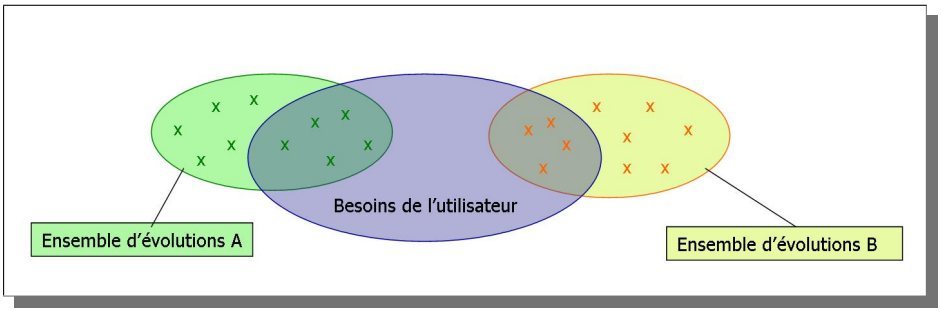

Figure 3.27 – Pertinence des evolutions multi-sources

Le but d'un processus de vérification de la pertinence est donc de **filtrer**, dans tous les ensembles d'évolutions proposés, les évolutions qui **ne sont pas pertinentes** pour l'utilisateur et qui risqueraient de **détériorer la qualité externe** du jeu de données de l'utilisateur. La solution que nous préconisons pour effectuer ce filtrage s'appuie sur les **métadonnées associées aux évolutions et aux besoins de utilisateurs** et repose sur les travaux de [Jeansoulin et Wilson, 2002] et [Vasseur, 2004]. Nous abordons dans la suite de ce paragraphe une méthode générale pour effectuer cette tache mais nous ne rentrons pas dans le détail de chacun des modules.

La figure 3.28 montre le processus de vérification de la pertinence qui permet donc de filtrer, parmi les nombreux ensembles d'évolutions proposés à un acteur de l'infrastructure, les évolutions qui ne sont pas en adéquation avec son besoin. Ce processus utilise les ensembles d'évolutions et leurs métadonnées associées, ainsi que les besoins des utilisateurs qui ont été spécifiés dans les paragraphes 3.2 et 3.3 et fournit en sortie un ensemble contenant uniquement les évolutions pertinentes pour chaque ensemble d'évolutions proposé. L'idée de cette méthode est de procéder par analyse et comparaison des métadonnées afin de déterminer si les évolutions sont pertinentes ou non pour l'utilisateur.

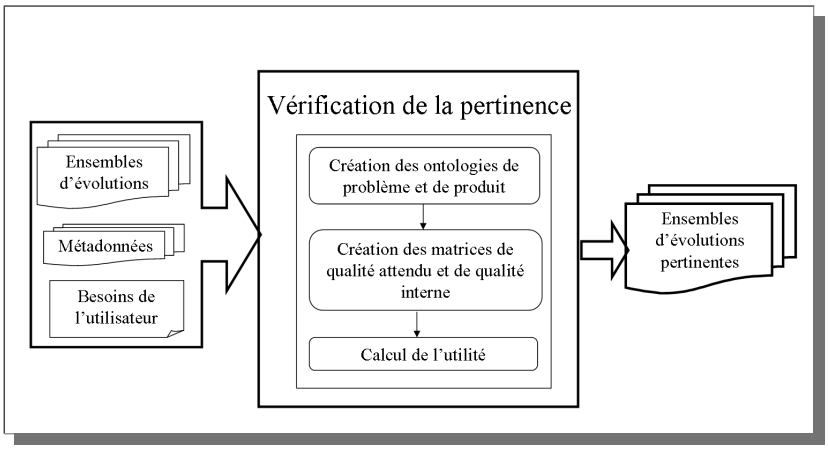

Figure 3.28 – Processus de vérification de la pertinence des évolutions

Pour ce faire, nous nous basons sur les travaux de [Jeansoulin et Wilson, 2002] et [Vasseur, 2004] dont le but est d'évaluer la qualité externe d'un jeu de données en fonction de l'emploi qui pourra en être fait par une communauté d'utilisateurs (Cf. §2.14). Nous nous inspirons de ces travaux pour évaluer la qualité externe d'un ensemble d'évolutions par rapport à l'usage que l'utilisateur final pourrait en faire.

Trois étapes sont nécessaires pour évaluer la qualité externe :
- Création d'ontologies de problème et de produit dans un référentiel commun [Jeansoulin et Wilson, 2002]
- Définition des matrices de qualité attendue et interne [Vasseur, 2004]
- Evaluation de l'adéquation aux besoins par calcul de l'utilité [Vasseur, 2004]

Les ontologies de produit et de problème fournissent la vision du monde réel

selon, respectivement, le point de vue du producteur de données et le point de vue de l'utilisateur final en formalisant les caractéristiques des données et les besoins des utilisateurs (Cf. Figure 3.29).

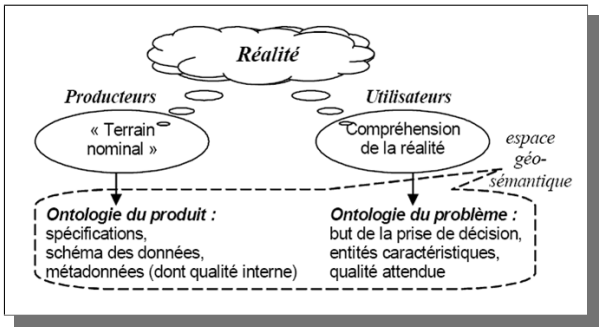

Figure 3.29 – Ontologies de problème et de produit selon [Jeansoulin et Wilson, 2002]

Dans [Jeansoulin et Wilson, 2002], les ontologies de produit correspondent à la perception qu'a le producteur de données du monde réel. Elles sont définies par les spécifications des données, le schéma des données et par les métadonnées de production et notamment celles qui renseignent sur la qualité interne des données.
Les ontologies de problème correspondent quant à elles, à la perception du monde réel vue par l'utilisateur des données. Elles sont déterminées par des entités caractéristiques du problème à résoudre, par le but de la prise de décision associée au problème et par les métadonnées de l'application qui renseignent sur la qualité attendue des données.
Dans notre étude, l'ontologie de produit pourrait être spécifiée par les évolutions proposées à l'intégration et l'ontologie de problème par les besoins de l'utilisateur. Les métadonnées associées aux évolutions et aux besoins des utilisateurs peuvent servir à définir ces ontologies.

Les ontologies fournissent alors deux modèles comparables, permettant la définition de mesures de similarité (la qualité augmentant avec la similarité). Selon [Frank *et al.*, 2004], les caractéristiques des données et les besoins des utilisateurs doivent pouvoir être comparés afin de pouvoir calculer une « utilité ». A partir des ontologies de problème et de produit, les matrices de qualité interne et attendue sont donc créées dans un référentiel commun. [Vasseur, 2004] établit ensuite par comparaison, agrégation et normalisation la matrice de qualité de l'application qui permet le calcul de l'utilité. La figure 3.30 montre un exemple des matrices qualités utilisées par [Vasseur, 2004].
Dans notre étude, la matrice de qualité interne correspondrait à la qualité des évolutions et la matrice de qualité attendue aux besoins de l'utilisateur. Les métadonnées associées aux évolutions et aux besoins des utilisateurs étant normalisées, elles peuvent être utilisées dans les différentes matrices.

Le calcul de l'utilité évalue les données en fonction des attentes de l'utilisateur et permet donc de fournir une estimation de l'adéquation des données aux besoins.

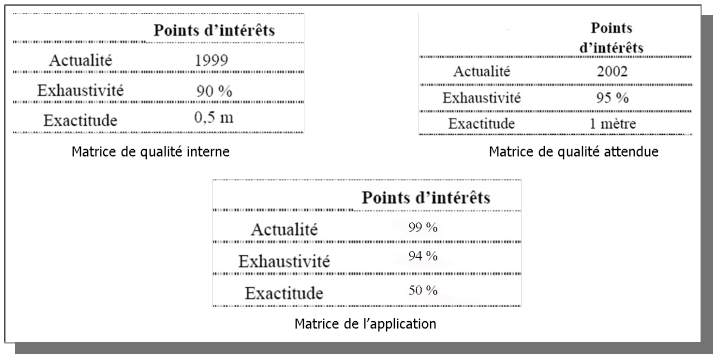

Figure 3.30 – Matrices de Qualité selon [Vasseur, 2004]

Dans [Vasseur, 2004], si la qualité est jugée insuffisante alors un retour arrière est programmé soit par reformulation de l'ontologie du problème, soit par analyse des données pour obtenir une matrice de qualité améliorée.
Dans notre étude, le calcul de l'utilité permettrait de savoir si la qualité des évolutions est jugée satisfaisante par rapport aux besoins de l'utilisateur. Dans le cas contraire, l'évolution est exclue de l'ensemble des évolutions. La difficulté ici réside dans le fait de définir un seuil à partir duquel le résultat du calcul de l'utilité est jugé mauvais et donc de considérer les évolutions comme non pertinentes.

3.4.3 Vérification de la cohérence des données et évolutions

Cette section traite de la vérification de la cohérence et constitue le coeur de la stratégie d'intégration des évolutions. Elle est constituée de deux étapes importantes, le contrôle de concurrence et la réconciliation des données conflictuelles [Pierkot et Mustiere, 2007]. Nous la divisons en trois parties. Tout d'abord, nous revenons sur la définition de la cohérence des données répliquées et celle des données spatiales, puis nous précisons la signification de la cohérence dans notre cadre de travail. Ensuite, nous abordons le contrôle de concurrence en lui-même en spécifiant d'une part les types de conflits qui peuvent se produire et d'autre part les moyens permettant de détecter ces conflits. Enfin, nous voyons comment la phase de réconciliation peut être réalisée et nous proposons des routines permettant d'effectuer un choix lorsque des évolutions sont conflictuelles.

Protocole de vérification de la cohérence

Dans cette partie, nous rappelons en premier lieu les différentes notions de cohérence telles qu'elles sont abordées en information géographique et en réplication optimiste, puis nous définissons la cohérence telle que nous la considérons dans ce travail de recherche. Nous voyons ensuite les différents niveaux de cohérence que nous pouvons rencontrer dans l'infrastructure. Enfin, nous présentons la vérification de la cohérence dans sa globalité et nous détaillons en particulier l'enchaînement entre les différentes étapes du processus.

En réplication optimiste, on dit qu'il y a **cohérence lorsque le système converge vers un état commun** c'est à dire lorsque toutes les opérations ont

été propagées sur tous les sites et que les répliques sont identiques.
En revanche, en information géographique, on dit que la **cohérence est assurée** lorsque les données produites **ne représentent pas une vision absurde du monde réel**. Nous distinguons trois types de cohérence en information géographique :

- La cohérence des modèles qui est relative aux choix de modélisation. Elle concerne par exemple le modèle topologique mis en place (spaghettis, réseau...), ou le type de données utilisé (raster, vectorielles).
- La cohérence des schémas qui est relative aux choix de représentation de l'information. Elle concerne par exemple les différences de granularité des attributs entre deux classes, ou le fait qu'une information peut être représentée comme une classe ou comme un attribut.
- La cohérence des données qui est relative aux instances elles mêmes. On en distingue deux catégories :
  - La cohérence spatiale qui concerne la géométrie et la topologie des objets géographiques. Par exemple, la précision des objets doit être de l'ordre du mètre ou une route ne doit pas traverser une maison.
  - La cohérence sémantique qui concerne les attributs non spatiaux. Par exemple, un attribut doit prendre ses valeurs dans un domaine prédéfini.

Dans le cadre de notre étude, **la cohérence ne sous entend pas convergence des copies** (dans le sens égalité) comme en réplication optimiste mais plutôt **compatibilité des copies** telle qu'elle est définie en information géographique. De surcroît, nous restreignons notre problème à l'étude de la **cohérence des données**, et non des modèles ou schémas qui est supposée traitée dans l'infrastructure.

Par ailleurs, différents **niveaux de cohérence** sont souhaités en fonction des besoins et des rôles des acteurs dans l'infrastructure. En effet, les producteurs qui doivent fournir l'information de référence et préparer les futurs jeux de données possèdent les moyens matériels nécessaires à la mise à jour et au partage de leurs données. L'objectif de ces acteurs est donc de recueillir de l'information de qualité afin d'obtenir un jeu de données fiable et précis. Cela suppose de prendre garde à limiter au maximum les incohérences qui peuvent se produire lors de l'intégration des évolutions dans leur jeu de données. Le **niveau de cohérence** doit donc être **élevé** car on privilégie **la qualité et la cohérence des données plutôt que la quantité d'information**.
Les utilisateurs ont quant à eux un rôle d'exploitation des données à des fins de prise de décision. Leur objectif premier est de pouvoir prendre rapidement des initiatives et d'être très réactif. Le but ici est donc de recueillir un maximum d'information répondant à un besoin particulier quel que soit sa qualité. On accepte donc que des incohérences apparaissent lors de l'intégration des évolutions dans le jeu de données. Le **niveau de cohérence** souhaité sera donc ici plutôt **faible** car on privilégie **la quantité à la qualité et à la cohérence des données**.
Les opérationnels doivent fournir des évolutions aux utilisateurs et faire remonter les informations recueillies aux producteurs. Ils possèdent des moyens matériels simplifiés qui leur permettent cependant de mettre à jour leurs données et de les partager ensuite avec les autres acteurs. Le but pour ces acteurs est de recueillir

un maximum d'informations et de les transmettre aux autres acteurs (producteurs et utilisateurs, en fonction de leurs besoins) tout en préservant un certain niveau de qualité. On **limite mais accepte l'existence de quelques incohérences** lors de l'intégration des évolutions dans le jeu de données. Le **niveau de cohérence** souhaité ici est donc **intermédiaire aux deux autres**.

Dans notre étude, nous devons donc gérer la cohérence à plusieurs niveaux que nous déterminons en fonction du rôle et des objectifs des acteurs de l'infrastructure. Pour cela, nous proposons un protocole de **vérification de la cohérence** à deux phases qui permet d'une part de **détecter les conflits** qui peuvent provoquer des incohérences et d'autre part, de proposer des routines de **réconciliation** des évolutions conflictuelles en fonction du niveau de cohérence souhaité. Par ailleurs, nous devons vérifier la cohérence d'une part entre les données de l'acteur et les évolutions proposées et d'autre part entre toutes les évolutions candidates à l'intégration. En effet, dans l'infrastructure, plusieurs acteurs sont en charge de la mise à jour des jeux de données et une évolution d'un même phénomène du monde réel, peut avoir été saisie plusieurs fois, à différents endroits ou à différents moments (Cf. Figure 3.31). Ces évolutions et ces données peuvent alors être en concurrence les unes avec les autres et provoquer des incohérences.

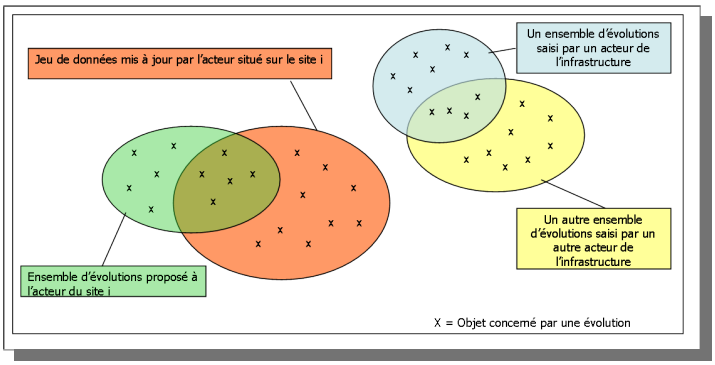

Figure 3.31 – Concurrence entre les données et les évolutions dans l'infrastructure spatiale

La figure 3.32 montre le déroulement du processus de vérification de la cohérence entre un jeu de données et un flot continu d'évolutions. Nous devons aussi mettre en place un processus similaire afin de vérifier la cohérence entre toutes les évolutions provenant des différentes sources. Notre méthode est divisée en deux phases distinctes mais fortement liées : **le contrôle de concurrence** et **la réconciliation**.

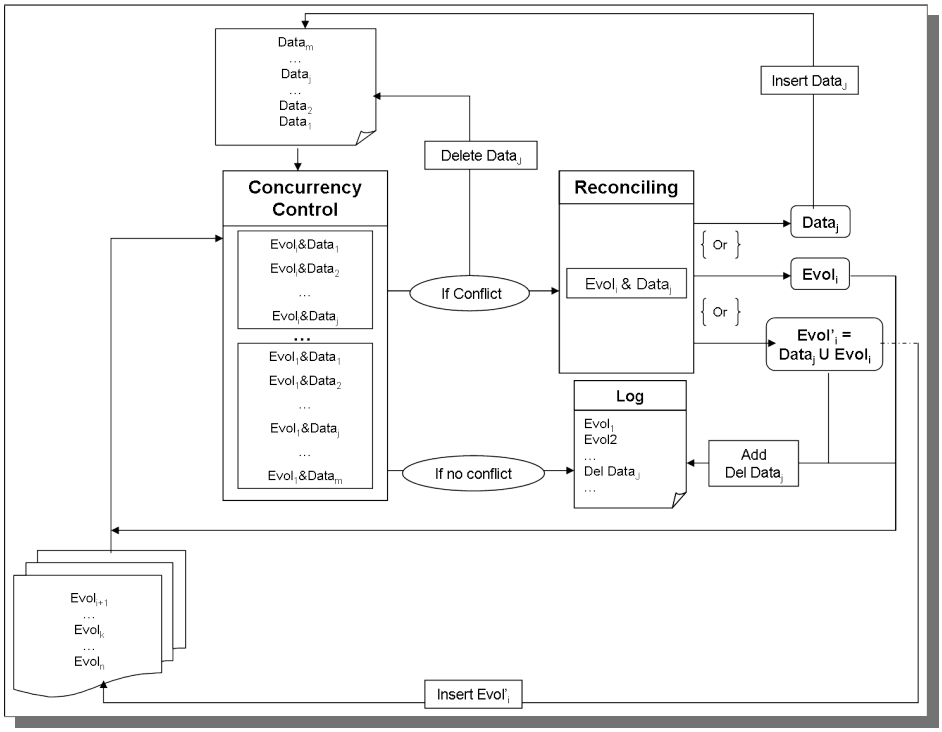

Figure 3.32 – Processus de vérification de la cohérence des données spatiales

Le module de contrôle de concurrence compare chaque évolution avec les données de l'utilisateur (respectivement, compare chaque évolution avec les autres évolutions) afin de détecter d'éventuels conflits. En réalité nous travaillons sur une copie du jeu de données afin qu'un jeu de données cohérent soit toujours disponible pour les acteurs ayant besoin d'utiliser les données. Le module de réconciliation est quant à lui activé dès qu'un conflit est détecté. Son but est de choisir la donnée qui sera finalement intégrée dans le jeu de données de l'utilisateur.

Le détail de l'algorithme `Consistency_Checking(EvolSet, DataSet)` qui est en charge de la vérification de la cohérence entre un ensemble d'évolutions et le jeu de données d'un acteur est donné ci dessous :

```
Consistency_Checking(FeatureSet EvolSet, FeatureSet DataSet) {
  While EvolSet not Empty{
    For all data from DataSet {
       If (Concurrency_Control(evolution, data)){
        If (Reconcile(evolution, data) == 0) {
          Delete(data) from DataSet;
          Delete(evolution) from EvolSet;
          Add(new evolution) in EvolSet ;
          Start again with new evol and all the data;
        }
        Else if (Reconcile(evolution, data) > 0) {
         Delete(data)from DataSet ;
         Continue with this evolution until there are no others data;
        }
        Else   {
         Delete(evolution) from EvolSet
         Start again with new evolution and all the data;
        }
       }
       Else EvolLog.add(evolution);
    }
  }
}
```

Cette méthode vérifie d'abord que la liste des évolutions à traiter n'est pas vide puis contrôle chaque évolution avec toutes les données de l'acteur grâce au prédicat `Concurrency_Control(evolution, data)` qui retourne vrai si un conflit est détecté entre deux objets (nous détaillons cette méthode dans le paragraphe consacré au contrôle de concurrence).

- S'il n'y a pas de conflit alors l'évolution est stockée dans un journal `EvolLog` en attente d'un deuxième contrôle qui sera effectué entre toutes les évolutions candidates à l'intégration afin de vérifier qu'elles ne soient pas elles-mêmes en conflit les unes avec les autres. Ce deuxième contrôle a lieu juste avant l'intégration des évolutions.
- Si un conflit est détecté alors la méthode `Reconcile(evolution, data)` est appelée et propose un choix de réconciliation des données conflictuelles (le détail de ce processus est donné dans le paragraphe consacré à la réconciliation).
  - Si l'évolution est choisie alors la donnée est supprimée de `DataSet` et le contrôle reprend à l'endroit où il s'était arrêté avec la même évolution afin de vérifier qu'il n'existe pas de conflit avec les données restantes.
  - Si une nouvelle évolution est créée (à partir de la donnée et de l'évolution en cours de traitement) alors la donnée est supprimée de `DataSet`, l'évolution en cours de traitement est supprimée de `EvolSet` et l'évolution nouvellement créée est insérée dans la liste des évolutions à traiter. Le contrôle redémarre ensuite avec une nouvelle évolution prise dans `EvolSet`.
  - Si la donnée est choisie alors l'évolution est supprimée de `EvolSet` et le contrôle de concurrence redémarre avec une nouvelle évolution.

La méthode s'exécute tant qu'il reste des évolutions à analyser ou jusqu'à ce qu'une session de mise à jour soit activée. Par ailleurs, les évolutions effectuées par l'utilisateur entre deux sessions de mise à jour sont considérées comme les autres évolutions et doivent être testées avant toute intégration.
Finalement, à la sortie de ce processus, un journal contenant les évolutions non conflictuelles avec les données de l'utilisateur est fourni.

Contrôle de concurrence

Nous allons maintenant détailler le premier des deux modules du processus de vérification de la cohérence, à savoir le contrôle de concurrence.

Le contrôle de concurrence permet de détecter les éventuels conflits qui peuvent provoquer des incohérences dans un jeu de données lors de l'intégration. Dans notre étude, **un conflit** est provoqué par une évolution qui **viole la cohérence** des données tel que nous l'avons définie dans le précédent paragraphe. Cependant, pour pouvoir détecter les conflits, il faut au préalable les caractériser.

Un conflit peut être déclenché soit par une **évolution impliquant deux objets du monde réel** et engendré par des niveaux de détails différents ou des erreurs et imprécisions de saisies (par exemple, la création d'une route sur une maison), soit par **deux évolutions relatives au même objet** du monde réel mais issues de sources distinctes et causés par des différences dans la géométrie ou dans les valeurs d'attribut ou encore par la manière dont les évolutions sont décrites. Typiquement, un utilisateur peut avoir fait une modification et un autre utilisateur, une suppression suivie d'une création pour décrire la même évolution.

Les conflits que nous détectons pendant la phase de contrôle de concurrence peuvent être de trois types : les **conflits de mise à jour** (modification ou suppression d'objet), les **conflits topologiques** (intersection ou recouvrement d'objets) et les **conflits de création** (créations multiples d'un même objet).

La figure 3.33 montre un conflit de modification du même objet du monde réel.On peut voir dans cet exemple que les identifiants fournis avec les mises à jour sont identiques. Ces deux évolutions sont donc concurrentes et leur intégration si elle n'est pas contrôlée provoquera une incohérence sémantique car la valeur de l'attribut EXS diffère.

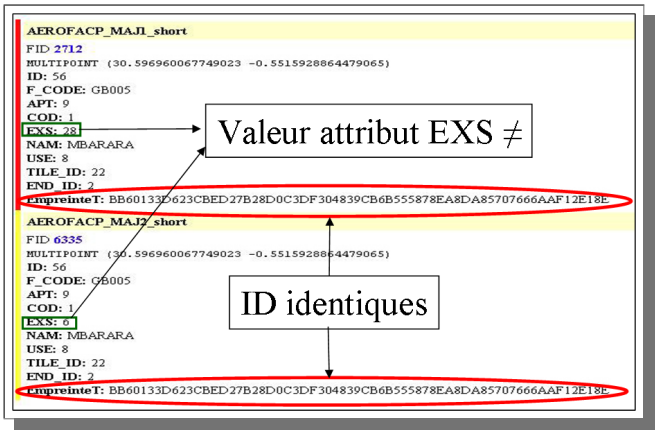

Figure 3.33 – Conflit de modification

La figure 3.34 montre un deuxième exemple qui illustre un conflit d'intersection entre une route et un bâtiment. Ici la route traverse un bâtiment ce qui risque d'entraîner une incohérence topologique lors de l'intégration si le conflit n'est pas traité au préalable.

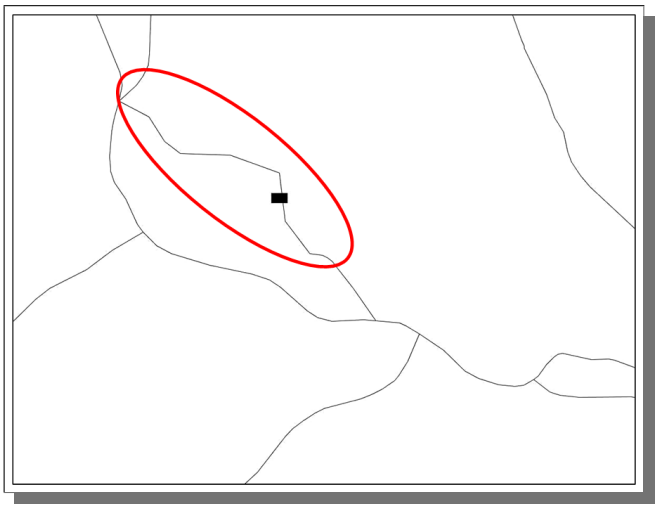

Figure 3.34 – Conflit d'intersection

La figure 3.35 montre un troisième et dernier exemple de conflit. Ici, nous avons affaire à un conflit de type créations multiples car les deux routes nouvellement créées concernent le même objet du monde réel. Par ailleurs, nous observons qu'elles sont localisées à des endroits différents, ce qui suppose que l'une au moins des deux routes est entachée d'imprécision de localisation. L'intégration sans traitement de ces deux routes occasionnera une incohérence spatiale et une incohérence sémantique.

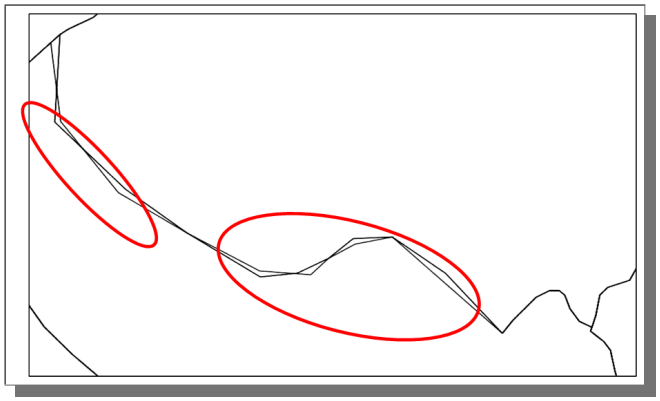

Figure 3.35 – Conflit de créations multiples

Les conflits ainsi caractérisés, nous pouvons proposer des méthodes pour les détecter :

- Pour déterminer les conflits de mise à jour, nous nous appuyons sur les **identifiants** fournis avec les évolutions et les données. En effet, dans l'infrastructure que nous avons mise en place, les données possèdent initialement des identifiants uniques et pérennes, générés à partir de la géométrie et des attributs des objets spatiaux. Les mises à jour de type modification et suppression ayant été saisies à partir de données existantes possèdent une référence à ces identifiants que nous pouvons comparer pour détecter les conflits. Cette méthode a l'avantage d'être simple à mettre en oeuvre et permet de détecter rapidement les évolutions effectuées sur des objets spatiaux qui existent dans un jeu de données utilisé dans l'infrastructure mais n'est pas utilisable pour détecter des conflits de type topologique ou sur des évolutions de type création.
- Les conflits topologiques sont quant à eux détectés grâce à des tests sur **les relations spatiales entre les objets**. Les identifiants ne sont pas ici exploitables car les objets concernés par l'évolution ne sont pas les mêmes.
- Enfin, les conflits de création sont détectés grâce à des techniques **d'appariement géométrique**. Les identifiants ne sont pas non plus exploitables ici car ils sont générés lors de l'intégration des évolutions dans le jeu de données, et n'existent donc pas encore à ce stade de la stratégie.

L'algorithme permettant d'effectuer le contrôle de concurrence entre deux objets (une donnée et une évolution ou deux évolutions) est donné ci-dessous. Il s'agit du prédicat `Concurrency_Control` qui retourne vrai si un conflit est détecté entre les deux objets passés en paramètre. Cette fonction effectue simultanément les trois tests permettant de détecter les conflits spécifiques (au niveau des instances) à la mise à jour de données spatiales :

- Premièrement une vérification de la similitude des identifiants est effectuée afin de détecter les conflits de mise à jour. Les identifiants sont stockés comme des attributs sur les objets, il est donc possible de les comparer facilement.
- Ensuite un appariement géométrique est exécuté pour détecter les conflits dus à de multiples créations du même objet. L'appariement géométrique calcule la distance (distance euclidienne ou distance de Haussdorf) entre deux objets et évalue la possibilité de correspondance en fonction d'un seuil. La difficulté ici

réside dans le choix du seuil. En effet, un seuil trop élevé conduit à détecter trop de concurrence alors qu'un seuil trop petit conduit à fournir trop peu de conflits. Le choix d'un « bon » seuil permet donc de détecter suffisamment de conflits tout en limitant les erreurs.
- Enfin, un test vérifiant une éventuelle intersection entre les objets est réalisé pour détecter des conflits topologiques. En effet, lorsqu'une relation spatiale existe entre deux objets (chevauchement, inclusion,...), il y a forcément une intersection entre ces deux objets (dans le sens mathématique du terme). Cependant, une intersection entre deux objets peut s'avérer ne pas être finalement conflictuelle (par exemple, le fait qu'un pont chevauche une rivière est une relation tout à fait légitime) mais la relation dans ce cas doit être exprimée explicitement avec les évolutions (Cf.§3.2.2). Après avoir effectué un test d'intersection, nous vérifions donc qu'il n'existe par de relations entre les objets.

Pour les conflits de types topologiques et géométriques, nous nous appuyons sur des enveloppes englobant chaque objet pour effectuer nos calculs sur la géométrie. Cela nous permet d'accélérer les calculs et permet parfois de détecter des conflits sur des objets ayant des géométries très erronées ou très imprécises, mais nous devons utiliser avec précaution ces enveloppes afin qu'elles ne donnent un résultat erroné en provoquant des conflits qui n'ont pas lieu d'être.

```
Boolean Concurrency_Control(Feature a, Feature b){
  If (a.getID() == b.getID())
    return true;
  Else if ((a.getEnveloppe().getGeom().apparie(b.getEnveloppe().getGeom()))
  ||   (b.getEnveloppe().getGeom().apparie(a.getEnveloppe().getGeom())))
    return true;
  Else if ((a.getEnveloppe().getGeom().intersect(b.getEnveloppe().getGeom()))
  || (b.getEnveloppe().getGeom().intersect(a.getEnveloppe().getGeom()))){
    if (a.link(b)) return false;
    else return true;
  }
  Else return false;
}
```

Formalisation ECA

Nous allons maintenant formaliser les différents enchaînements du processus de contrôle de concurrence grâce aux règles Événement-Condition-Action que nous avons décrites dans le paragraphe 3.4.1.

La règle `Consistency_Checking` découle directement de la règle `Activate_Consistency_Checking` (Cf. §3.4.1.) mais n'est activée qu'à la fin de cette dernière. Son but est, après avoir vérifié que la liste des évolutions à traiter n'est pas vide, d'effectuer le contrôle de concurrence (activation de la règle `Concurrency_Control`).

```
Règle : Consistency_Checking
Condition : if NotEmpty(EvolList)
   Action
    Enable(Concurrency_Control)
```

La règle `Concurrency_Control` est quant à elle activée par la règle `Consistency_Checking` ou lorsque le précédent contrôle n'a détecté aucun conflit (la règle `Control_NoConflict` est terminée) ou lorsque le processus de réconciliation a effectué un choix (règles `Choose_Evol, Choose_Data ou Create_Evol` terminées). Son but est de vérifier la concurrence entre l'évolution courante et une donnée appartenant à la copie de la base de données de l'utilisateur par le biais de la fonction `Concurrency_Control(currentEvol, data)`. L'exécution de l'action est immédiate si la condition est remplie.

```
Règle : Concurrency_Control
Evénement : Consistency_Checking enabled || end(Control_NoConflict)
|| end(Choose_Evol) || end(Choose_Data) || end(Create_Evol)
   Action
    Concurrency_Control (currentEvol, data)
```

La règle `Control_Conflict` est déclenchée par l'appel du prédicat `Concurrency_Control(currentEvol, data)` mais n'est activée que si un conflit a été détecté entre l'évolution et la donnée. Dans ce cas, le contrôle de concurrence est stoppé (règle `Concurrency_Control` désactivée) et le processus de réconciliation est lancé.

```
Règle : Control_Conflict
Evénement : Concurrency_Control (currentEvol, data)
Condition : if Conflict(currentEvol, data)
Action :
      Begin
       Disable(Concurrency_Control)
       Reconcile (currentEvol, data)
      End
```

La règle `Control_NoConflict` est également déclenchée par l'appel de la fonction `Concurrency_Control (currentEvol, data)` mais n'est activée que lorsqu'aucun conflit n'a été détecté entre l'évolution et la donnée. Dans ce cas, l'évolution est stockée dans le journal des évolutions `LogEvol` et la vérification de la concurrence est relancée avec une nouvelle évolution (réactivation de la règle `Concurrency_Control`).

```
Règle : Control_NoConflict
Evénement : Concurrency_Control (currentEvol, data)
Condition : if NOT Conflict(currentEvol, data)
Action :
      Begin
        Add (currentEvol) into LogEvol
        Enable (Concurrency_Control)
      End
```

Au début de chaque session de mise à jour, des règles équivalentes sont utilisées pour vérifier la cohérence entre toutes les évolutions contenues dans `LogEvol` car elles peuvent être en conflit les unes avec les autres. La règle `Concurrency_Control_Evol` est activée lorsqu'une session de mise à jour (règle `Update_Session`) est activée ou lorsque la règle `Control_NoConflict_Evol` est terminée. Cette règle, après avoir vérifié que le journal contenant les évolutions n'est pas vide, appelle elle aussi la fonction `Concurrency_Control` afin de vérifier la concurrence entre l'évolution courante et une autre évolution contenue dans le journal `LogEvol`

```
Règle : Concurrency_Control_Evol
Evénement : Update_Session enabled || end(Control_NoConflict_Evol)
Condition : If LogEvol not empty
   Action
    Concurrency_Control (currentEvol, evol)
```

La règle `Control_Conflict_Evol` est déclenchée par l'appel du prédicat `Concurrency_Control(currentEvol, evol)`. Si un conflit existe entre les évolutions alors la vérification de la concurrence est stoppée (règle Concurrency_Control_Evol désactivée) et le processus de réconciliation est lancé.

```
Règle : Control_Conflict_Evol
Evénement : Concurrency_Control (currentEvol, evol)
Condition : if Conflict(currentEvol, evol)
Action :
      Begin
       Disable(Concurrency_Control)
       Reconcile (currentEvol , evol)
End
```

La règle `Control_NoConflict_Evol` est également déclenchée par l'appel de la fonction `Concurrency_Control (currentEvol, evol)`. Si aucun conflit n'est détecté entre les évolutions alors l'évolution courante est stockée dans le journal des évolutions `FinalLogEvol` qui contient les évolutions qui seront finalement intégrées et la vérification de la concurrence est relancée avec une nouvelle évolution (activation de la règle `Concurrency_Control_Evol`).

```
Règle : Control_NoConflict_Evol
Evénement : Concurrency_Control (currentEvol, evol)
Condition : if NOT Conflict(currentEvol, evol)
Action :
      Begin
        Add (currentEvol) into FinalLogEvol
Enable (Concurrency_Control_Evol)
End
```

Deux contrôles de concurrence sont donc proposés dans la stratégie d'intégration des mises à jour, chacun dépendant du type de données à contrôler. Le premier contrôle est activé dès qu'une évolution est ajoutée dans la liste des évolutions à traiter et permet de **détecter les conflits entre les évolutions proposées et les données de l'utilisateur**. Le second contrôle est quant à lui effectué au début de chaque session de mise à jour et permet de **détecter les évolutions qui sont en conflit les unes avec les autres**. Les enchaînements des différents traitements permettent de réactiver le contrôle, soit avec la même évolution à l'endroit où il a été stoppé, soit avec une nouvelle évolution au début du traitement. L'issue de ce processus est quant à elle la même quel que soit le type de contrôle effectué : soit **l'évolution n'est pas conflictuelle, elle est alors stockée dans un journal**, soit **l'évolution provoque un conflit, le processus de réconciliation est alors lancé** afin qu'une solution soit proposée.

Nous détaillons dans la suite de ce paragraphe le protocole de réconciliation et en particulier nous montrons comment nous utilisons les métadonnées pour effectuer le choix le plus adéquat en fonction du niveau de cohérence voulu, des besoins de l'utilisateur et des évolutions proposées. Nous discutons ensuite de la nécessité d'automatiser au maximum ce processus et décrivons finalement les différents enchaînements du processus.

Réconciliation des données conflictuelles

Le protocole de réconciliation intervient lorsqu'un ou plusieurs conflits sont détectés durant la phase de contrôle de concurrence. L'objectif de ce processus est de proposer l'option la plus appropriée permettant de fournir une solution adéquate à la résolution du ou des conflits. Le résultat dépend du niveau de cohérence souhaité et de l'équilibre voulu entre qualité et quantité.

L'originalité du protocole de réconciliation que nous proposons vient du fait qu'il exploite les métadonnées associées aux entités du modèle DAE défini dans le paragraphe 3.1 afin de proposer un résultat conforme aux attentes de l'acteur. Nous rappelons que les métadonnées associées aux entités du modèle DAE sont les suivantes :

- **Les métadonnées des données** qui sont conformes à la norme ISO 19115 et qui fournissent des informations notamment sur l'identification et la qualité des ressources.

- **Les métadonnées des évolutions** qui sont conformes au profil de métadonnées MUMSDI que nous avons créé pour la gestion des évolutions dans un contexte d'infrastructure militaire. Ces métadonnées fournissent des renseignements sur les évolutions partagées dans une infrastructure militaire à des granularités différentes (évolution élémentaire, ensemble d'évolutions ...), ainsi que des informations concernant la qualité et la fiabilité des évolutions.
- **Les métadonnées des acteurs** qui sont conformes au format défini dans le §3.3.4 et qui se distinguent en deux catégories, les besoins des utilisateurs et les contraintes de cohérence. Cependant, les métadonnées des acteurs utilisées pour la réconciliation dépendent du niveau de cohérence souhaité. En effet, lorsque le niveau de cohérence désiré est élevé (c'est le cas pour les producteurs de l'infrastructure), alors le processus donnera un poids plus important aux contraintes de cohérence qui ont été spécifiées dès le début de la mission et qui ne peuvent évoluer pendant toute la durée de la mission. En revanche, lorsque le niveau de cohérence est faible (c'est le cas pour les utilisateurs de l'infrastructure), alors le processus considérera plutôt les métadonnées relatives aux besoins des acteurs, qui dépendent des objectifs et de l'urgence de la situation (période de crise ou campagne de relevé) et qui peuvent évoluer au cours de la mission. Pour un niveau de cohérence intermédiaire (c'est le cas pour les opérationnels de l'infrastructure), le processus utilise les contraintes de cohérence et les besoins des utilisateurs sans privilégier l'un des deux types de métadonnées.

Le principe du protocole de réconciliation repose donc sur **l'utilisation des métadonnées** pour effectuer un choix lorsqu'une évolution est en conflit avec une autre ressource (l'autre ressource est soit une donnée, soit une autre évolution). Les métadonnées fournies avec les évolutions et les données donnent des informations telles que la qualité ou l'origine. Ces informations vont permettre au processus de comparer les éléments avec les attentes de l'acteur (contraintes ou besoins). Ainsi lorsqu'un conflit doit être traité, le processus peut effectuer un choix et proposer une réconciliation qui dépende du niveau de cohérence souhaité et de l'utilisation qui est souhaitée.

Le processus de réconciliation se déroule en plusieurs étapes :

- Premièrement, une **comparaison** des métadonnées associées aux éléments en conflits est effectuée avec les métadonnées des acteurs et un **calcul de mesures qualités** pour chacune des caractéristiques des éléments (géométrie, attributs, fiabilité, ...) est réalisé. Cette partie s'inspire des travaux sur le calcul de l'utilité de [Grum et Vasseur, 2004] et [Frank *et al.*, 2004].
- Ensuite, nous calculons une mesure de qualité globale pour chaque élément en conflit afin d'obtenir un résultat qui dépende des attentes de l'utilisateur final et du niveau de cohérence souhaité. Cette partie s'inspire des travaux de [Vasseur, 2004].
- Enfin, une **comparaison** des mesures qualités globales des élements en conflit est effectuée et l'élément considéré comme étant le plus pertinent est **choisi** en vue d'une future intégration. Cette partie peut s'exécuter automatiquement, semi-automatiquement ou interactivement.

Comparaison des métadonnées et calcul des mesures qualités

La première action du processus de réconciliation est donc de **comparer les métadonnées**. Une comparaison n'est possible que si les métadonnées sont uniformisées et que des correspondances sont explicitement établies. Nous avons, au paragraphe 3.2., élaboré des ensembles de métadonnées contenant des informations caractéristiques sur les données, les évolutions et les acteurs. Ces ensembles sont dans des formats très proches car ils ont été spécifiés conformément aux exigences de la norme ISO 19115. Nous pouvons donc **établir des correspondances** entre les différents éléments de chaque ensemble de métadonnées. Le tableau 3.2 montre quelques correspondances entre les métadonnées des évolutions et celles des acteurs.

Type d'élément	**Métadonnées MUMSDI**	**Besoins des acteurs**	**Contraintes de cohérence**
Etendue	MD_Metadata .MD_DataIdentification .extent	Zone Spatiale Maximale	Contrainte .Contextuelle .Etendue
Date	MD_Metadata .MD_DataIdentification .date	Date Actualité Minimale	Contrainte .Contextuelle .DateActualité
Exhaustivité	MD_Metadata .DQ_DataQuality .DQ_Completeness	Exhaustivité Minimale	Contrainte .Contextuelle .Exhaustivité
Précision Géométrique	MD_Metadata .DQ_DataQuality .DQ_PositionnalAccuracy	Precision géométrique minimale	Contrainte .Geometrique .Precision
Résolution	MD_Metadata .DQ_DataQuality .LI_Lineage.LI_Source .scaledenominator		Contrainte .Geometrique .Résolution
Précision Attributaire	MD_Metadata .DQ_DataQuality .DQ_ThematicAccuracy	Précision attributaire minimale	Contrainte .Attributaire .Precision
Sources	MD_Metadata .DQ_DataQuality .LI_Lineage.source		Contrainte .Contextuelle .Sources
Processus de création des évolutions	MD_Metadata .DQ_DataQuality .LI_Lineage.processStep		Contrainte .Contextuelle .Processus

Table 3.2 – Tableau de correspondances entre les Métadonnées des évolutions et les métadonnées des acteurs

Par ailleurs, nous **classons les éléments de métadonnées** en fonction du type d'information qu'elles produisent. Nous distinguons cinq caractéristiques essentielles pour mesurer la qualité d'une ressource par rapport aux attentes d'un

utilisateur [2] : des caractéristiques géométriques (la précision et la résolution), des caractéristiques sémantiques (la précision quantitative et la précision qualitative), des caractéristiques de généalogie (les sources et les processus de construction des ressources), des caractéristiques de fiabilité (le type d'erreur et la confiance accordée aux acteurs) et d'autres caractéristiques plus générales comme l'actualité, l'exhaustivité et l'étendue de l'ensemble contenant les ressources. Nous calculons ensuite des mesures qualité pour chacune de ces caractéristiques prises individuellement.

Nous **calculons la mesure qualité de chaque caractéristique** grâce à un calcul de distance qui mesure l'écart entre la qualité souhaitée par l'acteur et la qualité effective de la ressource. Nous faisons ici la différence entre la valeur souhaitée par l'acteur et la valeur de la ressource. Ce calcul est réalisable pour la majorité des caractéristiques car les valeurs des qualités sont des valeurs numériques ou des pourcentages.

Pour les autres types de valeurs, nous devons créer une métrique associée aux éléments. C'est le cas pour les informations de généalogie où nous disposons d'une liste d'éléments à comparer à une autre liste (par exemple, la liste des sources ayant conduit à la mise à jour que nous devons comparer à la liste des sources acceptées par l'acteur), la mesure qualité est ici la différence entre le nombre total d'éléments souhaités et le nombre d'éléments correspondants de la liste proposée.

Calcul de la mesure de qualité globale

A l'issue de la seconde étape, nous souhaitons obtenir **une mesure de la qualité globale** qui corresponde à la mesure qualité de la ressource en conflit par rapport aux attentes de l'acteur qui en fera l'usage. Cette mesure dépend du niveau de cohérence souhaité et sera par conséquent différente d'un acteur à l'autre (même si les mesures qualités de chaque éléments caractéristiques sont identiques). Cette étape est divisée en trois phases :

- Premièrement, nous **normalisons les mesures qualités** de chaque caractéristique obtenues lors de la précédente étape afin que toutes les mesures soient dans une seule et même unité et soient de ce fait comparables. La normalisation doit conduire à obtenir des valeurs comprises entre -1 et 1.
- Ensuite, nous **affectons un poids** à chacune des mesures qualités prises individuellement afin de prendre en considération le niveau de cohérence et les attentes de l'acteur qui utilisera les données. Par exemple, le résultat des mesures qualité doivent prendre en considération qu'un producteur préfère collecter peu de données mais des données qui soient fiables géométriquement et sémantiquement alors qu'un opérationnel préfère obtenir beaucoup d'informations quel que soit sa qualité. Le résultat de ces mesures sera donc différent si on les calcule pour un acteur de type producteur ou pour un acteur de type opérationnel. Le poids est une constante que l'on multiplie à la valeur qualité. Sa valeur dépend du niveau de cohérence souhaité et des exigences de l'acteur, elle est donc contextuelle.

2. Ces éléments caractéristiques peuvent également servir lors du filtrage des évolutions non pertinentes

– Enfin, nous **faisons une agrégation** des mesures qualités de chaque caractéristique afin d'obtenir une mesure globale. Pour cela, nous effectuons ici une moyenne qui permet d'avoir une vision globale de la qualité externe de la ressource par rapport aux besoins de l'utilisateur et au niveau de cohérence attendu.

Sélection de l'élément pertinent

La troisième phase du processus de réconciliation est l'étape finale qui permet **d'effectuer un choix** entre les éléments conflictuels. La décision pour chaque évolution dépend des résultats des mesures de qualités globales des ressources conflictuelles. Trois résultats peuvent constituer le choix final :

– L'évolution correspond mieux aux attentes de l'acteur, elle est choisie. Si on se trouve dans le cas où l'on compare une évolution avec une donnée alors l'évolution remplacera la donnée à la prochaine session de mise à jour.
– La donnée correspond mieux aux attentes de l'acteur, elle est conservée et l'évolution est supprimée.
– Aucune des deux ressources prise dans leur globalité ne satisfait entièrement les attentes de l'acteur mais certaines caractéristiques prises séparément peuvent se révéler pertinentes, alors une nouvelle évolution peut être constituée à partir de ces caractéristiques. Cela arrive lorsqu'une partie de l'évolution et une partie de l'autre ressource donnent une meilleure solution si elles sont combinées que l'une ou l'autre prise individuellement. Par exemple, supposons que l'évolution ait une géométrie précise et que la donnée ait des attributs plus fiables. Si le niveau de cohérence est élevé et que la contrainte générale est d'avoir les données les plus précises alors une nouvelle évolution est créée avec la géométrie de l'évolution et les attributs de la donnée.

Par ailleurs, cette troisième étape peut se faire automatiquement, semi-automatiquement ou interactivement en fonction des résultats obtenus. En effet, il est toujours préférable d'automatiser un maximum le protocole de réconciliation afin que les résultats obtenus soient harmonisés, surtout lorsque les acteurs ont des rôles, des objectifs et des besoins communs dans l'infrastructure. Cela est possible lorsqu'il y a suffisamment d'informations (en quantité et de qualité) disponibles avec les évolutions et les données pour que les mesures qualités soient précisément calculées. Cependant, lorsqu'une nouvelle évolution peut être proposée par agrégation des caractéristiques pertinentes des deux ressources, il est parfois difficile de choisir entre plusieurs solutions qui s'avèrent toutes acceptables et le protocole propose alors une liste de choix à l'utilisateur pour validation. En dernier recours, lorsqu'aucune solution satisfaisante n'est trouvée alors le processus laisse la charge à l'utilisateur de choisir la solution qu'il considère comme étant la meilleure. Cette issue doit évidemment être évitée le plus souvent possible car les acteurs, même s'ils travaillent sur un jeu de données ayant des contraintes identiques, ont des visions différentes du monde réel et leur analyse d'une même situation peut conduire à des interprétations hétérogènes, ce qui n'est évidemment pas souhaitable.

Formalisation ECA

Comme pour les autres processus de la stratégie d'intégration, nous formalisons les enchaînements du protocole de réconciliation grâce aux règles ECA suivantes. Nous nous plaçons ici dans le cas où les ressources en conflit sont une donnée et une évolution mais une analyse analogue peut être effectuée pour la réconciliation entre deux évolutions concurrentes :

La règle `Reconcile` définit le mécanisme du processus de réconciliation. Elle est déclenchée par l'appel de la fonction `Reconcile (currentEvol, data)`. La seule action de cette règle est d'appeler la méthode qui va permettre au processus d'effectuer un choix entre la donnée et l'évolution.

```
Règle : Reconcile
Evénement :  Reconcile(currentEvol, data)
Condition :
Action :
Choose (currentEvol, data)
```

Les trois règles suivantes (`Choose_Data, Choose_Evol, Create_Evol`) sont déclenchées en parallèle par le même événement, à savoir l'appel de la fonction `Choose (evol, data)` mais leurs actions diffèrent en fonction du résultat de la condition.

Si la donnée est choisie alors l'évolution est supprimée de la liste des évolutions à traiter et la vérification de la concurrence reprend avec l'évolution courante.

```
Règle : Choose_Data
Evénement : Choose(currentEvol,data)
Condition : if data is choosen
Action
Begin
Delete (currentEvol) from EvolList
Enable (Concurrency_Checking)
End
```

Si l'évolution est choisie alors la donnée est supprimée de la copie de la base de données de l'utilisateur et l'action « supprimer la donnée » est ajoutée au journal des évolutions traitées. La vérification de la concurrence reprend avec l'évolution courante afin de vérifier qu'il n'existe pas d'autres conflits avec les autres données.

```
Règle : Choose_Evol
Evénement : Choose(currentEvol,data)
Condition if currentEvol is choosen
Action
Begin
  AddAction(Delete data) into LogEvol
  Delete (data) from BDCopy
    Enable (Concurrency_Checking)
End
```

Si une nouvelle évolution est créée (grâce à une partie de la donnée et une partie de l'évolution) alors la donnée est supprimée de la copie de la base de données, l'évolution est également supprimée de la liste des évolutions à traiter, l'action « supprimer la donnée » est ajoutée au journal des évolutions, la nouvelle évolution est stockée dans la liste des évolutions à traiter afin de vérifier qu'elle ne provoque pas de nouveaux conflits avec les autres données et le contrôle de concurrence reprend avec une nouvelle évolution.

```
Règle : Create_Evol
Evénement : Choose(currentEvol,data)
Condition if  new Evol is created
Action
Begin
  AddAction(Del data) into LogEvol
  Delete(data) from BDCopy
  Delete (currentEvol) from EvolList
  Create(newEvol)and add(newEvol) into EvolList
  Enable (Concurrency_Checking)
End
```

Nous avons proposé dans cette partie une méthode complète pour vérifier la cohérence entre des évolutions provenant de sources multiples et un jeu de données d'un acteur particulier. Cette méthode s'appuie sur un contrôle de la concurrence entre les ressources et sur des méthodes de réconciliation afin de vérifier et de traiter les éventuels conflits. Le contrôle de la concurrence que nous proposons se base sur le type de conflit et sur les caractéristiques des évolutions. Les techniques de réconciliation s'appuient sur les métadonnées associées aux différentes entités pour proposer des solutions qui ne remettent pas en cause la cohérence du jeu de données et qui tiennent compte des exigences des acteurs qui en feront l'usage.

Finalement, lorsque toutes les évolutions ont été contrôlées avec les données de l'utilisateur et avec les autres évolutions, nous obtenons un journal contenant uniquement des évolutions qui ne sont pas conflictuelles et qui correspondent aux attentes et aux besoins de l'acteur. Nous pouvons donc intégrer ses évolutions sans remettre en cause la cohérence de la base de données de l'acteur, cette étape se déroule pendant les sessions de mises à jour.

3.4.4 Sessions de mise à jour

L'existence de sessions de mise à jour par rapport à une intégration en continu des mises à jour apporte deux bénéfices majeurs. Premièrement, elles permettent de limiter les actions sur les données de l'utilisateur. En effet, les produits d'évolutions multi-sources arrivant en continu, il est possible de recevoir successivement plusieurs évolutions concernant le même objet. Si on intégrait immédiatement l'évolution dans le jeu de données utilisateur, il faudrait parfois défaire et refaire les actions quand une nouvelle évolution se présenterait, avec tous les problèmes de recalage que cela implique. Ensuite, elles nous permettent d'avoir des états du jeu de données considérés comme stables à des instants donnés, états dans lesquels on peut facilement revenir en cas de besoin si les journaux des modifications sont conservés.

La fréquence des sessions de mise à jour est étroitement liée à la cohérence. En effet, les utilisateurs ont besoin d'obtenir rapidement de l'information quelle que soit sa qualité. Il est donc souhaitable que la durée entre les sessions de mises à jour soit courte afin que les jeux de données soient mis à jour le plus souvent possible quitte à défaire et refaire souvent les opérations, ce qui est un moindre mal étant donné que la cohérence souhaitée est faible. En revanche, au niveau des producteurs, les informations doivent être cohérentes et de qualité, et des actions interactives peuvent être réalisées pour cela. Il est dont préférable que le temps entre les sessions de mise à jour soit plus élevé afin de minimiser les opérations sur les données.

La figure 3.36 montre la progression d'une session de mises à jour. La méthode récupère les évolutions non conflictuelles provenant du journal des évolutions (FinalEvolLog) qui ont été traitées entre le temps Ti et le temps Tj et les intègre dans la version Vi du jeu de données de l'acteur. Au final, si le traitement s'est déroulé sans problème, le processus crée une nouvelle version Vj du jeu de données et le journal permettant de passer de la version Vi à la version Vj est stocké.

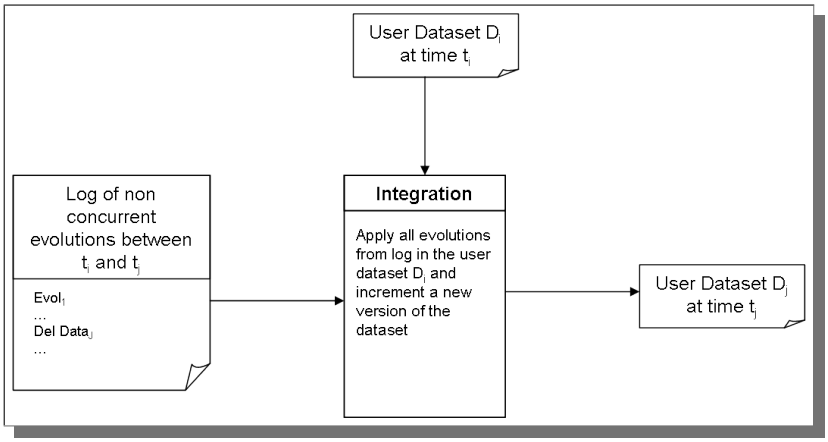

Figure 3.36 – Organisation d'une session de mise à jour

La dernière règle ECA que nous détaillons ici est celle qui lance le processus d'intégration des évolutions dans la base de données de l'utilisateur c'est-à-dire la règle `Update_Session`. Elle découle directement de la règle `Activate_Update_Session` mais les actions ne sont déclenchées qu'à la condition

que le journal contenant les évolutions déjà traitées ne soit pas vide. Si la condition est remplie alors la règle active dans un premier temps le contrôle de concurrence entre toutes les évolutions présentes dans `LogFile` afin de vérifier qu'elles ne sont pas concurrentes les unes avec les autres. Enfin, lorsque cette vérification est terminée, l'intégration des évolutions contenues dans le journal des évolutions traitées (`FinalLogFile`) est effectuée dans la base de données courante de l'utilisateur. La règle s'auto-désactive lorsque la transaction effectuant l'intégration est terminée.

```
Règle : Update_Session
Condition : if NotEmpty(LogEvol)
Action :
Begin
enable(Concurrency_Control_Evol)
T = Integrate(FinalLogEvol) into BDCurrent
End
```

Finalement, après chaque session de mise à jour, nous avons un jeu de données cohérent, dans une nouvelle version, obtenu suite à l'intégration des évolutions non conflictuelles provenant de différentes sources de l'infrastructure. Ce jeu de données constitue la nouvelle référence pour l'acteur qui utilise les données à des fins de prise de décision.

Chapitre 4

Mise en oeuvre et évaluation de la stratégie d'intégration des mises à jour

4.1 Introduction

Ce chapitre constitue la mise en oeuvre de ce travail de thèse et une évaluation des performances de la stratégie d'intégration des mises à jour. Nous supposons que le filtrage des évolutions non adéquates est effectué dès la réception des évolutions et nous nous concentrons en particulier sur la phase de vérification de la cohérence.

Le contexte de simulation que nous avons choisi est celui d'un acteur situé sur le terrain d'action qui possède un jeu de données dérivé d'un jeu de référence qu'il a transformé en fonction de son environnement et de ses besoins. Nous nous situons dans la quatrième phase de la mission militaire (déploiement de longue durée) où les acteurs sont déployés depuis un certain temps et où des mises à jour locales ont déjà été effectuées en fonction de relevés de terrain ou de collectes de nouvelles données. A ce stade de la mission, des évolutions provenant des autres acteurs de l'infrastructure (producteurs ou opérationnels, situés au quartier général ou sur le terrain d'action) sont proposées à notre acteur de référence en vue d'une éventuelle intégration.

Les données tests (données, évolutions et métadonnées) utilisées dans ce contexte de simulation sont décrites dans la partie 4.2. Nous décrivons en premier lieu les données militaires constituant le jeu de données de référence. Ensuite, nous précisons quelles sont les évolutions qui sont proposées à l'acteur de référence. Enfin, nous voyons quelles sont les métadonnées attachées aux données, acteurs et évolutions.

Les résultats de la simulation sont donnés au paragraphe 4.3. En particulier, nous donnons dans la partie 4.3.1, le bilan du contrôle de concurrence entre les évolutions proposées et les données de l'acteur. Puis, nous examinons dans la partie 4.3.2., l'issue de la réconciliation des données conflictuelles. Nous effectuons ce test pour deux types d'acteurs distincts (producteur ou opérationnel) afin de montrer que les effets obtenus dépendent des besoins et des attentes de l'utilisateur final.

Enfin, nous discutons de ces résultats et nous concluons.

4.2 Données tests

Nous avons limité notre étude aux données vectorielles et en particulier aux données vectorielles militaires qui sont structurées dans le format VMap (Vector Map). Nous discutons de ce type de données dans le paragraphe 4.2.1.

Nous voyons ensuite quelles sont les évolutions qui sont proposées à l'acteur. Ne disposant pas de mises à jour réelles, nous avons simulés les évolutions pour effectuer nos tests. Les évolutions sont livrées dans des fichiers XML qui contiennent uniquement les évolutions selon la classification établie dans la section 3.2 [W3C, 2006]. Des explications concernant cette simulation et l'implémentation du format de livraison des évolutions sont fournies au paragraphe 4.2.2.

Enfin, nous discutons des métadonnées associées aux données, évolutions et acteurs dans la partie 4.2.3. Les métadonnées utilisées dans cette stratégie d'intégration sont formatées selon les schémas exposés au paragraphe 3.3. et leurs mises en oeuvre s'appuient sur la norme ISO 19139 publiée par l'ISO [ISO19139, 2003].

4.2.1 Données vectorielles militaires

Les données de référence utilisées dans une infrastructure de données spatiales sont constituées par un ensemble de jeux de données nécessaires à l'intérêt commun.

Dans le contexte militaire, les données sont conformes au standard DIGEST qui est un standard réglementaire au sein des armées de l'OTAN pour l'échange de données géographiques numériques [DIGEST, 2000]. Il est principalement destiné à la formalisation des échanges de données géographiques entre les différentes unités (services d'un producteur, producteurs nationaux ou utilisateurs) et a fait l'objet de nombreux profils d'implémentation. En particulier, les spécifications des produits vectoriels résultant de ce standard sont les produits VMap [VMAP0, 1999],[VMAP1, 1995], [VMAP2, 1993], [VMAPUrban, 2000] qui sont construits sur les formats VPF et VRF [VPF, 1998],[VRF, 2000].

Les formats VPF (Vector Product Format) et VRF (Vector Relational Format) définissent une norme d'échange et un format de stockage des données vectorielles. Ils permettent de structurer, documenter et localiser des données géographiques dans une structure homogène [VPF, 1998], [VRF, 2000].

Les produits VMap constituent une collection de bases de données qui fournit les données géographiques vectorielles à petite, moyenne et grande échelle. Les données sont séparées en neuf couches thématiques (par exemple, les limites administratives, l'hydrographie, les transports, les zones habitées, la végétation ...) et sont structurées topologiquement. Les produits VMap sont édités à différentes échelles :

- VMap0 constitue un ensemble de données au niveau mondial qui peut être visualisé à une échelle de 1/1 000 000 [VMAP0, 1999]. La base de données ainsi constituée est destinée à fournir des informations géoréférencées à petite échelle. En fonction de la fiabilité des sources, la précision du VMAP0 varie de 4000 à 2000 m.
- VMap1 constitue un ensemble de données dont le contenu correspond à une carte à l'échelle 1/250 000 [VMAP1, 1995]. En fonction de la fiabilité des sources, la précision du VMAP1 varie de 500 à 125 m.
- VMap2 constitue un ensemble de données dont le contenu correspond à une carte thématique à l'échelle 1/50 000 [VMAP2, 1993]. En fonction de la fiabilité des sources, la précision du VMAP2 varie de 200 à 50 m.
- VMapUrbain constitue un ensemble de données au niveau urbain à une échelle urbaine [VMAPUrban, 2000]. Il consiste en une représentation des plans de villes.

Par ailleurs, un produit VMap est constitué de plusieurs ensembles de données contenant chacun l'information concernant une couche thématique particulière (par exemple la couche `ROADL` référence uniquement les données relatives au réseau routier dont la forme géométrique est linéaire).

Les données de référence utilisées dans notre contexte de simulation sont des données vectorielles, au format VMAP1 couvrant la région de Fort Portal en Ouganda (Cf. Figure 4.1). L'étude 1 du projet Envol VDC a permis d'ajouter des identifiants uniques et pérennes à ces données originales [Raynal, 2005]. Le schéma a donc été transformé pour prendre en compte ces identifiants.

Ces données sont initialement fournies à notre acteur de référence dans un format Shape qui est un format simplifié d'échange de données vectorielles [ESRI, 1998]. Un fichier Shape stocke uniquement les attributs (enregistrements dans une base de données) et la géométrie des objets géographiques (forme géométrique de type point, ligne ou polygone, décrite par un ensemble de coordonnées) et ne considère pas la topologie. De ce fait, les données sont accessibles plus rapidement et requièrent moins d'espace disque mais des calculs additionnels doivent être effectués pour retrouver les relations entre les objets géographiques.

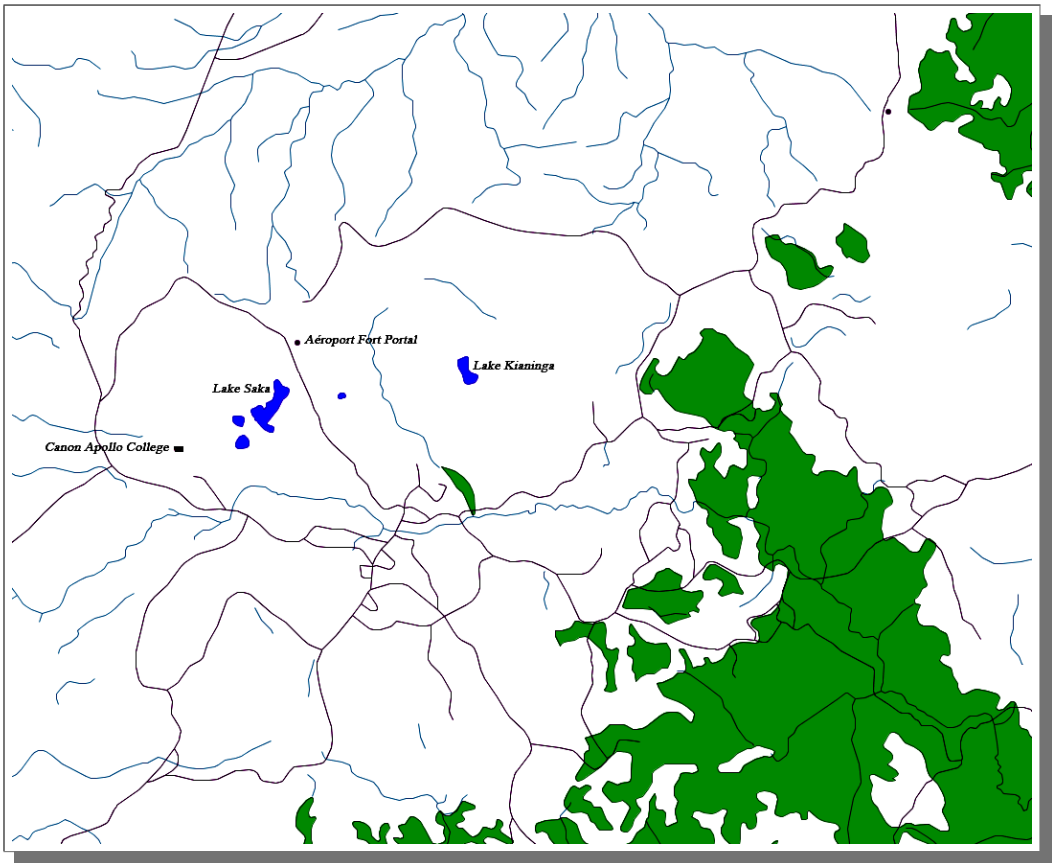

Figure 4.1 – Extrait du jeu de données tests de référence

Par ailleurs, nous considérons uniquement les données relatives aux couches thématiques concernant l'hydrographie, les transports, les zones habitées et la végétation. En particulier, nous nous intéressons aux routes, rivières, aéroports, chemin ferrés et lacs.

4.2.2 Ensembles d'évolutions

Ne disposant pas de mises à jour réelles, les évolutions ont été simulées par plusieurs chercheurs du laboratoire Cogit.

Protocole de simulation des évolutions

La saisie a été réalisée indépendamment par chaque expert, grâce au système d'information géographique Jump [Aquino et Kim, 2003]. La mise à jour a été effectuée sur le même produit de référence mais la liberté a été laissée aux utilisateurs concernant le choix des mises à jour ainsi que les couches thématiques à mettre à jour. Les évolutions acceptées sont celles qui ont été définies dans la partie 3.2.1. de ce manuscrit à savoir les créations, les suppressions, les modifications géométriques, attributaires ou mixtes. Par ailleurs, nous avons supposé que les personnes en charge de la mise à jour étaient des militaires en mission. Nous avons donc demandé aux personnes effectuant la saisie de se placer dans des conditions se rapprochant des conditions réelles d'un acteur d'une infrastructure militaire. En particulier, nous

avons demandé à un expert de prendre le rôle d'un producteur situé au quartier général, à un autre celui d'un opérationnel sur le terrain. Par ailleurs, pour certains d'entre eux, la période de saisie était une période de crise, pour d'autres une simple campagne de mise à jour. Pour chaque acteur, la précision des évolutions est donc fonction des conditions que nous avons imposées dans le protocole de simulation et les évolutions ne possèdent de ce fait pas la même qualité.

La figure 4.2 montre un extrait d'un produit mis à jour par un acteur. En particulier, nous distinguons sur cet aperçu, la création d'un nouveau bâtiment proche du collège Canon Apollo, la création d'un aéroport ainsi que la modification d'une partie du réseau routier.

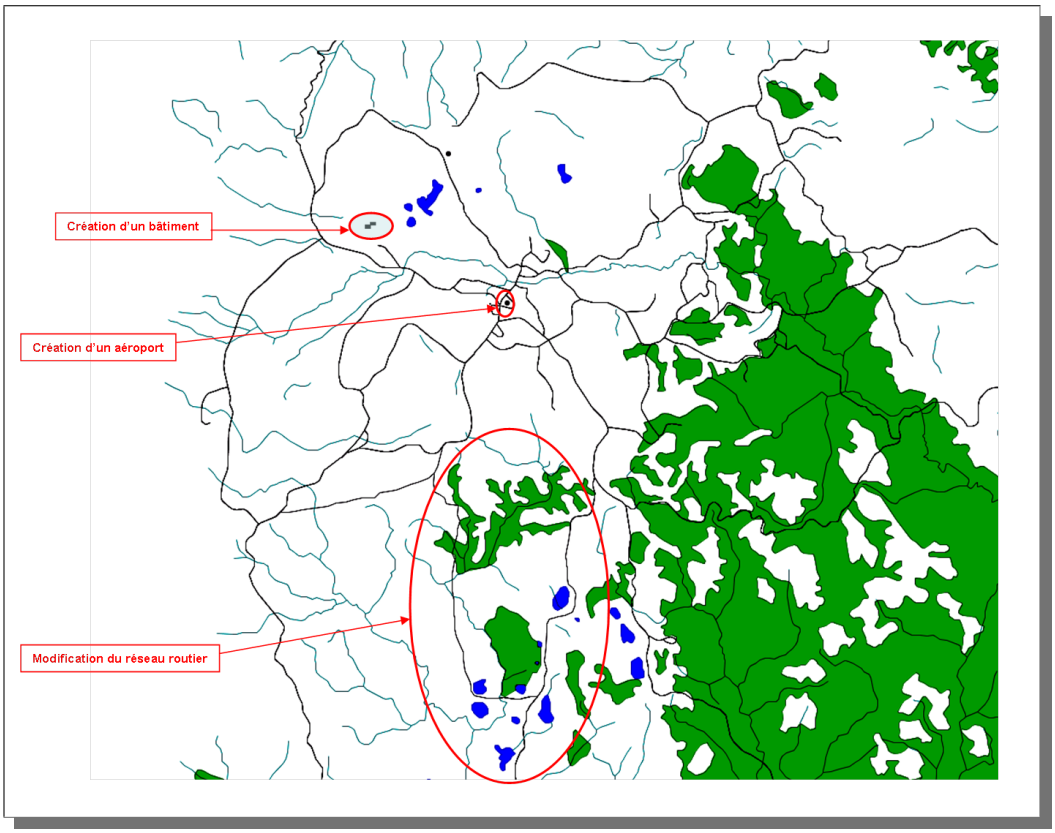

Figure 4.2 – Exemple de mises à jour simulées

Parallèlement, le jeu de données de l'acteur de référence a aussi été mis à jour. La figure 4.3 montre un extrait de la mise à jour du jeu de données de l'acteur, en particulier, la mise à jour d'une partie du réseau routier.

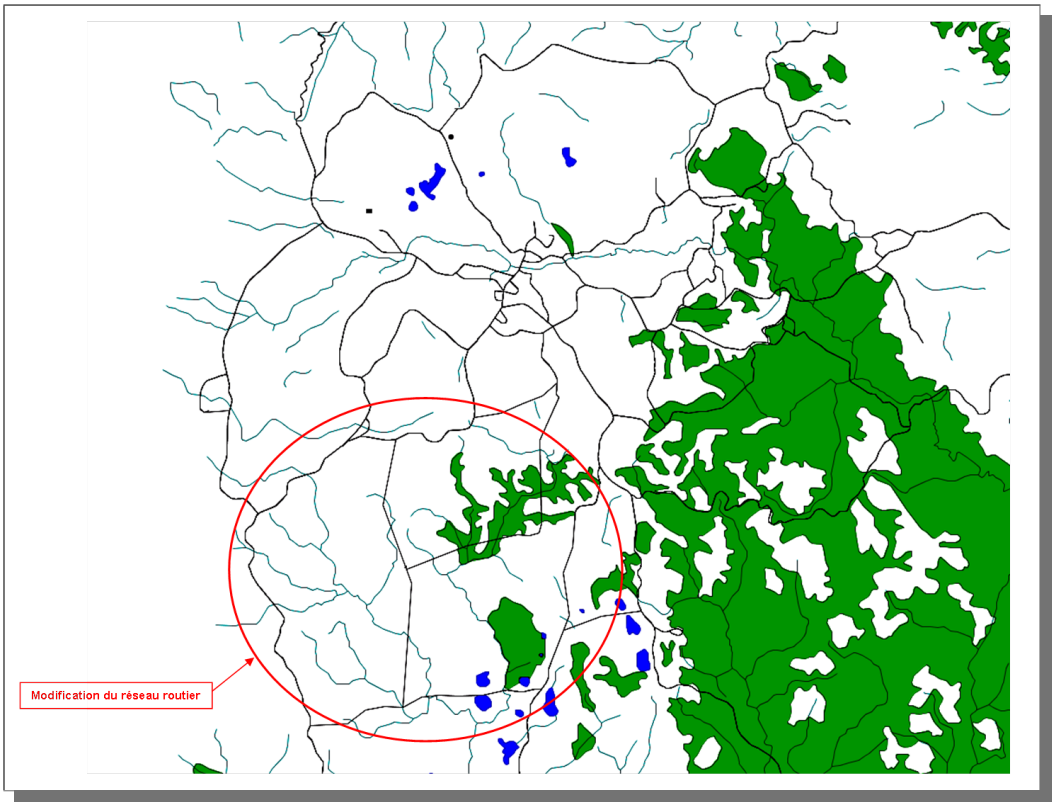

Figure 4.3 – Extrait du jeu de données de l'acteur de référence mis à jour

Lorsque la saisie est terminée, chaque couche thématique ayant été mise à jour est exportée au format GML [GML, 2007]. Un fichier GML correspond à l'ensemble des données d'une couche thématique (par exemple la couche concernant le réseau routier).

Extraction des évolutions

Nous devons ensuite extraire les évolutions contenues dans les différents fichiers GML afin d'obtenir uniquement l'information de mise à jour. Nous détectons les changements entre les fichiers de référence et chaque fichier mis à jour grâce au logiciel DeltaXML ([Fontaine, 2001] qui est un comparateur de fichiers XML. Pour cela, nous avons comparé chaque fichier GML issu de l'exportation des couches thématiques mises à jour avec le fichier GML correspondant à la couche de référence ayant servi pour la mise à jour. La structure d'un fichier DeltaXML est la suivante :

```
Espaces de noms
<schemaFeatures deltaxml:delta="unchanged"> : Schéma des fichiers XML testés
et indication d'un éventuel changement
<dataFeatures deltaxml:delta="WFmodify"> : Ensemble des données
et indication d'un éventuel changement

<!-- Puis pour chaque donnée : -->
<gml:featureMember deltaxml:delta="unchanged"> : pas de changement
<gml:featureMember deltaxml:delta="WFmodify"> : mise à jour de type modification
   <deltaxml:PCDATAold> : ancienne(s) valeur(s) de la donnée
   <deltaxml:PCDATAnew> : nouvelle(s) valeur(s) de la donnée
<gml:featureMember deltaxml:delta="delete"> : mise à jour de type suppression
<gml:featureMember deltaxml:delta="add"> : création d'une nouvelle donnée
```

L'entête du fichier est constituée des divers espaces de noms.
Puis le détail du schéma décrivant la structure des fichiers XML qui ont été testés est fourni. Si les schémas sont différents alors DeltaXML le signale.
Vient ensuite, l'ensemble des données contenues dans les fichiers XML. DeltaXML signale toutes les différences entre les fichiers analysés, en indiquant s'il s'agit d'une mise à jour (modification ou suppression) ou d'une création.

L'extrait suivant montre un exemple de modification détectée par le logiciel DeltaXML :

```
<!-- Exemple de mise à jour de donnée, ici modification géométrique -->
<gml:featureMember deltaxml:delta="WFmodify">
   <Feature deltaxml:delta="WFmodify">
       <featureType deltaxml:delta="unchanged">JCSOutput</featureType>
       <property deltaxml:delta="unchanged" name="gml2_coordsys"/>
       <gml:LineStringProperty deltaxml:delta="WFmodify">
          <gml:LineString deltaxml:delta="WFmodify">
             <gml:coordinates deltaxml:delta="WFmodify">
                <deltaxml:PCDATAmodify>
                   <deltaxml:PCDATAold>
                   30.267528533935547,0.6448313593864441
                   30.267356872558594,0.6453275680541992
                   30.26719093322754,0.647222638130188
                   30.267290115356445,0.6474847197532654
                   30.267658233642578,0.647607147693634
                   30.270709991455078,0.6472131609916687
                   </deltaxml:PCDATAold>
                   <deltaxml:PCDATAnew>
                   30.267528533935547,0.6448313593864441
                   30.267356872558594,0.6453275680541992
                   30.26719093322754,0.647222638130188
                   30.267290115356445,0.6474847197532654
                   30.267426421327205,0.6476169593600929
                   30.267658233642578,0.647607147693634
                   30.270709991455078,0.6472131609916687
                   </deltaxml:PCDATAnew>
                </deltaxml:PCDATAmodify>
             </gml:coordinates>
          </gml:LineString>
       </gml:LineStringProperty>
       <property deltaxml:delta="unchanged" name="ID">4527.0</property>
```

```
        <property deltaxml:delta="unchanged" name="F_CODE">AP030</property>
        <property deltaxml:delta="unchanged" name="ACC">1.0</property>
        <property deltaxml:delta="unchanged" name="EXS">28.0</property>
        <property deltaxml:delta="unchanged" name="LOC">8.0</property>
        <property deltaxml:delta="unchanged" name="LTN">0.0</property>
        <property deltaxml:delta="unchanged" name="MED">2.0</property>
        <property deltaxml:delta="unchanged" name="NAM">UNK</property>
        <property deltaxml:delta="unchanged" name="RST">2.0</property>
        <property deltaxml:delta="unchanged" name="RTT">0.0</property>
        <property deltaxml:delta="unchanged" name="USE">0.0</property>
        <property deltaxml:delta="unchanged" name="WTC">2.0</property>
        <property deltaxml:delta="unchanged" name="WD1">0.0</property>
        <property deltaxml:delta="unchanged" name="TILE_ID">16.0</property>
        <property deltaxml:delta="unchanged" name="EDG_ID">201.0</property>
        <property deltaxml:delta="unchanged" name="EmpreinteT">
        716B292DFFACD841207A9EF3E6129F099D42461A5A05FDC5B4A19E177861E49F
        </property>
    </Feature>
</gml:featureMember>
```

Les fichiers résultant de cette analyse fournissent toutes les données contenues dans les fichiers analysés agrémentées d'une information sur les évolutions éventuelles qu'elles ont subies (grâce aux mots clés `unchanged, WFModify, delete et add`). Cependant, la politique de gestion des évolutions mise en place dans l'infrastructure nous impose de livrer uniquement les évolutions dans un produit structuré.

Restructuration des évolutions

Nous devons donc restructurer ces ensembles afin de ne recueillir que les informations utiles c'est-à-dire uniquement les évolutions. Ces ensembles sont ensuite stockés dans un produit d'évolutions afin que ce dernier soit livré structuré aux autres acteurs, conformément à ce qui a été décidé dans la politique de gestion établie dans l'infrastructure.

Par ailleurs, c'est lors de cette étape que nous vérifions qu'une modification n'a pas été saisie comme une suppression, suivie de la création d'une nouvelle donnée. Ce cas est détecté lorsqu'une évolution de type création possède un identifiant (ce qui n'est normalement pas possible car les identifiants ne sont pas créés lors de la saisie de l'évolution) et qu'une autre évolution de type suppression possède le même identifiant. Si tel est le cas, nous regardons si l'évolution concerne la géométrie, les attributs ou les deux éléments et regroupons les deux évolutions en une seule de type modification (géométrique, attributaire ou mixte) afin de respecter le format imposé dans l'infrastructure.

Au final, nous obtenons des ensembles d'évolutions structurés contenant uniquement les évolutions organisées selon le schéma commun choisi pour être le format de livraison des évolutions au sein de l'infrastructure lors de la mise en place de la politique de gestion des évolutions. Ces ensembles structurés sont ensuite fournis à l'acteur de référence en XML. Ce sont ces ensembles qui seront utilisés dans la stratégie d'intégration des mises à jour. En particulier, lors de la phase de vérification de la

cohérence. L'extrait ci-dessous montre un exemple de produit d'évolutions structuré selon le format prédéfini dans l'infrastructure :

```
<Product name="EvolutionsActor1">
   <EvolutionSet name="ROADL">
      <Evolution name="Modification Geometrique1">
         <Object ID="716B292DFFACD841207A9EF3E6129F099D42461A5A05FDC5B4A19E177861E49F" />
         <Geometry type="Line" coordinates= "
         30.267528533935547,0.6448313593864441
         ...
         30.270709991455078,0.6472131609916687 " />
      </Evolution>
      <Evolution name="Modification Geometrique2">
         <Object ID="4883D93877D8435CCCE5934DE0379C218F7832B93DFA3F414D7B86110CF03E2C" />
         <Geometry type="Line" coordinates= "
         30.283443450927734,0.6697490811347961
         ...
         30.290632247924805,0.6701707243919373 " />
      </Evolution>
      <Evolution name="Creation1">
         <Geometry type="Line" coordinates= "
         30.247220993041992,0.5897606015205383
         ...
         30.179962158203125,0.561992883682251  " />
         <Attribut name="ACC" value="0.0" />
         <Attribut name="EXS" value="28.0" />
         <Attribut name="RST" value="3.0" />
         <Attribut name="USE" value="6.0" />
         <LinkEvol name="Creation2" />
      </Evolution>
      <Evolution name="Creation2">
         <Geometry type="Line" coordinates= "
         30.243764283762896,0.5783806737121173
         ...
         30.270288467407227,0.48546797037124634  " />
         <Attribut name="ACC" value="2.0" />
         <Attribut name="EXS" value="5.0" />
         <Attribut name="RST" value="1.0" />
         <Attribut name="USE" value="5.0" />
         <LinkEvol name="Creation1" />
      </Evolution>
      <Evolution name="Creation3">
         <Geometry type="Line" coordinates= "
         30.29962730407715,0.5539653301239014
         ...
         30.29023551940918,0.49694615602493286   " />
         <Attribut name="ACC" value="2.0" />
         <Attribut name="EXS" value="5.0" />
         <Attribut name="RST" value="0.0" />
         <Attribut name="USE" value="5.0" />
      </Evolution>
   </EvolutionSet>
   <EvolutionSet name="AEROFACP">
   <!-- Toutes les évolutions de la couche thématique AEROFACP -->
   </EvolutionSet>
   <!-- Toutes les évolutions des autres couches thématiques -->
</Product>
```

4.2.3 Métadonnées

Nous supposons ici que les métadonnées sont correctement remplies et attachées à chacune des entités du modèle Données Acteurs Évolutions.

Le principe général de la mise en oeuvre des métadonnées liées aux données et aux évolutions repose sur la recommandation de la norme ISO 19139 et sur les schémas XML produits par l'application de cette norme [ISO19139, 2003].
La mise en oeuvre des métadonnées attachées aux acteurs repose sur l'utilisation d'un fichier XML structuré selon le format défini dans la partie 3.3.4. de ce manuscrit.

Les données utilisées pour nos tests ne disposant pas d'informations, nous avons simulé un ensemble de métadonnées en essayant d'être au plus proche de la réalité. Nous nous sommes pour cela inspirés des métadonnées de la BDCarto et des spécifications des produits VMAP1.

Le tableau 4.1 montre une synthèse des métadonnées que nous avons définies pour les données de l'acteur de référence (données qui ont été mises à jour) à différentes granularités (le produit, l'ensemble des données du thème ROADL et une donnée particulière du thème ROADL). La valeur `null` signifie qu'on ne dispose pas de l'information ou que l'information n'est pas nécessaire pour ce champ.

Type d'information	**Le produit**	**La couche thématique ROADL**	**Une donnée**
Titre	Produit résultant de la mise à jour du produit de référence par l'acteur	Ensemble des données de la couche thématique ROADL mis à jour	la route 4472
Couches thématiques	ROADL, TREESA, RAILRDL, AEROFACP, LAKERESA, BUILDA, WATRCSRL	ROADL	ROADL
Étendue	30.092968139546496 0.152717484699114, 30.092968139546496 1.039281141645268, 31.048775262028194 1.039281141645268, 31.048775262028194 0.152717484699114	30.08183972544257 0.1588999369790732, 30.08183972544257 1.0528825366611785, 31.08710646616394 1.0528825366611785, 31.08710646616394 0.1588999369790732	null
Date	10/01/2008	5/01/2008	5/01/2008
Exhaustivité : Omission	2%	10%	null
Précision géométrique	de 125m à 500m	150m	150m
Résolution	1/250 000	idem produit	idem produit
Précision attributaire	bien documentés	à moitié documentés	tous documentés

Table 4.1 – Exemple de métadonnées fournies avec les données de l'acteur de référence

Concernant les évolutions, nous avons demandé à chaque personne ayant effectué la saisie de se mettre à la place d'un acteur de l'infrastructure militaire et de nous fournir des informations attachées aux évolutions telles que la qualité, la fiabilité, ... Le tableau 4.2 montre un exemple de métadonnées définies par un utilisateur, informations fournies avec les évolutions à différentes granularités (le produit, l'ensemble des évolutions du thème ROADL et une évolutions particulière du thème ROADL). La valeur `null` signifie qu'on ne dispose pas de l'information ou que l'information n'est pas nécessaire pour ce champ.

Type d'information	**Le produit**	**La couche thématique ROADL**	**Une donnée**
Titre	Produit d'évolutions	Ensemble des évolutions de la couche thématique ROADL	la création d'une route
Couches mises à jour	ROADL, AEROFACP, BUILDA, WATRCSRL	ROADL	ROADL
Étendue	30.092968139546496 0.152717484699114, 30.092968139546496 1.039281141645268, 31.048775262028194 1.039281141645268, 31.048775262028194 0.152717484699114	30.14861021006613 0.4618400986970757, 30.14861021006613 0.9032671914861651, 31.003025115156497 0.9032671914861651, 31.003025115156497 0.4618400986970757	null
Date	01/04/2008	15/03/2008	13/03/2008
Rôle de l'acteur	Opérationnel	Opérationnel	Opérationnel
Source	JDD de référence	Couche ROADL du JDD de référence	Couche ROADL du JDD de référence
Localisation de l'acteur	Terrain d'action	Terrain d'action	Terrain d'action
Conditions de saisie	Besoin rapide d'évolutions	Saisie rapide	Géométrie peu fiable, Valeurs d'attributs non renseignées
Précision géométrique	entre 100 et 500m	entre 150m et 200m	150m
Précision attributaire	a moitié documentés	à moitié documentés	pas documentés
Type erreur	toutes	attributaire et géométrique	attributaire et géométrique

Table 4.2 – Exemple de métadonnées fournies avec les évolutions

Enfin, pour les données des acteurs, nous avons choisi de simuler les deux cas que nous pouvons rencontrer dans l'infrastructure, à savoir les métadonnées relatives aux besoins des acteurs et les métadonnées relatives aux contraintes.
Le tableau 4.3 montre un exemple de métadonnées liées à un acteur de l'infrastructure dont le rôle est opérationnel. Les besoins de ce type d'acteur ne sont pas fixes et peuvent évoluer au cours de la mission. Les métadonnées présentées ici à différentes granularités (produit d'évolutions et évolutions du thème `ROADL`) peuvent être différentes selon la période donnée.

Type d'information	**Produit d'évolutions**	**Les évolutions de la couche thématique ROADL**
Besoins opérationnel n°125	Couches ROADL, AEROFACP, WATRCSRL	Tous types d'évolutions
Etendue	30.171036974903966 0.4572163983835835, 30.171036974903966 0.7103159044523032, 30.402013709331126 0.7103159044523032, 30.402013709331126 0.4572163983835835	30.171036974903966 0.4572163983835835, 30.171036974903966 0.7103159044523032, 30.402013709331126 0.7103159044523032, 30.402013709331126 0.4572163983835835
Date	01/02/2008	01/02/2008
Précision géométrique	500m maxi	300m maxi
Précision attributaire	a moitié documentés	pas documentés
Fiabilité géométrique	50%	50%
Fiabilité attributaire	70%	100%

Table 4.3 – Exemple de besoins d'un opérationnel de l'infrastructure

4.3 Mise en oeuvre et évaluation de la vérification de la cohérence

Dans cette partie, nous abordons la mise en oeuvre du processus de vérification de la cohérence et analysons les résultats issus de cette mise en oeuvre. En particulier, nous évaluons les processus de contrôle de concurrence et de réconciliation. Pour réaliser ces tests, nous avons utilisé les données, évolutions et métadonnées que nous avons décrites dans la section précédente. Par ailleurs, nous nous appuyons sur le système d'information géographique Jump pour effectuer nos tests [Aquino et Kim, 2003]. Jump est implémenté en Java et nous choisissons par conséquent de réaliser nos tests dans ce langage de programmation.
Dans Jump, les entités géographiques sont représentées par des objets de la classe Java `Feature`, les jeux de données (un jeu de données est constitué d'un ensemble de données d'une même couche thématique) sont représentés par la classe `FeatureDataset` qui implémente l'interface `FeatureCollection` et les produits (un produit est constitué d'un ensemble de jeux de données) par la classe `FeatureSuperCollection`. Nous avons spécialisé la classe `Feature` pour obtenir d'une part la classe `Evolution` qui permet de gérer les évolutions élémentaires et la classe `Data` pour la gestion des données, notamment pour le stockage des identifiants. De la même façon, nous gérons les ensembles de données et d'évolutions grâce à des classes dérivées de la classe `FeatureDataset` et les produits grâce à des classes dérivées de la classe `FeatureSuperCollection`.

Nous utilisons ensuite ces classes pour effectuer la vérification de la cohérence. Dans cette section, nous donnons tout d'abord, dans la partie 4.3.1, le détail de la

mise en oeuvre du contrôle de concurrence et analysons les résultats obtenus. Puis nous voyons la mise en oeuvre du mécanisme de réconciliation des données conflictuelles et nous évaluons les résultats obtenus dans la partie 4.3.2.

4.3.1 Contrôle de concurrence

Pour valider le contrôle de concurrence, nous avons considéré trois produits d'évolutions (que nous désignerons dans la suite par P1, P2 et P3). Ces produits sont issus de la mise à jour d'un jeu de données de référence et résultent de la simulation décrite dans la partie 4.2.2. Les produits que nous utilisons sont structurés selon le format défini dans l'infrastructure militaire. Par ailleurs, nous considérons que le jeu de données de l'acteur a déjà été mis à jour par l'acteur lui-même, il diffère donc du jeu de données de référence. La première phase de la stratégie consiste à extraire les données et évolutions des fichiers XML afin de stocker les instances dans leur classe respective. Nous travaillons ensuite exclusivement sur les instances de classe ainsi obtenues.

Processus automatique : Mise en oeuvre et résultats

Nous contrôlons dans un premier temps les produits d'évolutions avec les données de l'acteur de référence. Toutes les évolutions traitées (non concurrentes ou résultantes du protocole de réconciliation) sont stockées dans un journal qui est contrôlé avant chaque session de mise à jour afin de vérifier une éventuelle concurrence entre les évolutions elles mêmes. Le fonctionnement du contrôle de concurrence est le même quel que soit les données en entrée du processus car il repose sur un principe basé sur le contrôle des instances (l'instance pouvant être une donnée ou une évolution).

En effet, la fonction qui est en charge du contrôle de concurrence est un prédicat qui retourne vrai si un conflit est détecté entre deux objets (une donnée et une évolution ou deux évolutions). Cette fonction effectue trois tests permettant de détecter les conflits spécifiques (au niveau des instances) à la mise à jour de données spatiales :

- Vérification de la non similitude des identifiants pour détecter les conflits de mise à jour.
- Appariement géométrique pour détecter les conflits de créations.
- Tests sur les relations entre objets pour détecter les conflits topologiques.

Nous avons testé le processus de contrôle de concurrence entre les produits d'évolutions et les données de l'acteur de référence. Nous donnons dans la suite de cette partie, le résultat de ces tests pour chacun des produits d'évolutions.

Le tableau 4.4 donne les résultats globaux que nous avons obtenu avec le processus de contrôle de concurrence, pour chacun des produits proposés. Pour chaque produit, nous donnons le nombre d'évolutions saisies et le nombre total de conflits que le processus à détecté.

Produit d'évolutions	Nombre d'évolutions saisies	Nombre de conflits détectés
Produit P1	14 évolutions	19 conflits
Produit P2	21 évolutions	36 conflits
Produit P3	41 évolutions	45 conflits

Table 4.4 – Contrôle de concurrence entre les produits d'évolutions et les données de l'acteur

Nous allons maintenant détailler les résultats relatifs à la couche thématique `ROADL` (instances du réseau routier) car nous avons observé que la majorité des évolutions qui ont été saisies concernent les objets de ce thème et les trois types de conflits que nous recherchons sont représentés dans cette catégorie.

Le tableau 4.5 montre les résultats obtenus suite au contrôle de concurrence automatique entre les évolutions et les données de l'acteur de référence pour la couche `ROADL`. En particulier, nous donnons pour chaque produit d'évolutions, le nombre total de conflits détectés, puis nous détaillons le nombre de conflits en fonction de leur nature.

Produit d'évolutions	Nombre d'évolutions	Nombre de conflits de mise à jour	Nombre de conflits de créations	Nombre de conflits topologiques
Produit P1	5 évolutions	2 conflits	1 conflits	4 conflits avec des objets de la couche thématique `WATRCRSL`
Produit P2	9 évolutions	5 conflits	2 conflits	7 conflits avec des objets de la couche thématique `WATRCRSL`
Produit P3	34 évolutions	15 conflits	6 conflits	14 conflits avec des objets de la couche thématique `WATRCRSL` et 2 conflits avec des objets de la couche thématique `LAKERESA`

Table 4.5 – Contrôle de concurrence pour les objets de la couche `ROADL`

Vérification interactive : mise en oeuvre et résultats

Pour valider le processus automatique de contrôle de concurrence, nous nous sommes mis à la place d'un expert et avons effectué un contrôle visuel. Ce contrôle permet de déterminer précisément les conflits (conflits de mise à jour, conflits de

créations et conflits topologiques) qui doivent être détectés. Le contrôle est réalisé grâce au logiciel Jump, en superposant les produits d'évolutions avec le jeu de données de l'acteur de référence.
Nous cherchons dans un premier temps à détecter les conflits de mise à jour en analysant les évolutions de type modification et suppression. Pour cela, nous cherchons les objets dans le jeu de données de l'acteur qui possèdent le même identifiant que celui de l'évolution testée.
Ensuite, nous cherchons à détecter les conflits de création. Pour cela, nous prenons chaque couche thématique séparément et nous cherchons les correspondances entre les évolutions et les données en faisant un appariement visuel.
Enfin, nous repérons les conflits topologiques en trouvant les évolutions qui provoquent des intersections non souhaitées avec des objets d'autres couches thématiques. Nous obtenons ainsi un résultat optimal qui va nous servir de base à l'évaluation des résultats obtenus avec le processus automatique de contrôle de concurrence.

Le tableau 4.6 montre le résultat global que nous avons obtenu suite à cette analyse. Dans ce tableau, nous avons reporté, pour chaque produit d'évolutions, le nombre d'évolutions qui ont été saisies et le nombre total de conflits que nous avons détectés.

Produit d'évolutions	**Nombre d'évolutions saisies**	**Nombre de conflits détectés**
Produit P1	14 évolutions	17 conflits
Produit P2	21 évolutions	36 conflits
Produit P3	41 évolutions	43 conflits

Table 4.6 – Contrôle visuel entre les produits d'évolutions et les données de l'acteur

Le tableau 4.7 montre le résultat obtenu suite au contrôle visuel entre les évolutions et les données de la couche `ROADL`. Pour chaque produit, nous donnons tout d'abord le nombre total de conflits détectés. Puis, nous présentons le nombre de conflits trouvés en fonction du type de conflit recherché.

Produits Évolutions	Nb Évolutions	Nb Conflits de mise à jour	Nb Conflits de création	Nb Conflits topologiques
Produit P1	5 évolutions	2 conflits	1 conflit	4 conflits avec des objets de la couche thématique WATRCRSL
Produit P2	9 évolutions	5 conflits	2 conflits	7 conflits avec des objets de la couche thématique WATRCRSL
Produit P3	34 évolutions	15 conflits	4 conflits	14 conflits avec des objets de la couche thématique WATRCRSL et 2 conflits avec des objets de la couche thématique LAKERESA

Table 4.7 – Contrôle visuel pour les objets de la couche ROADL

Analyse des résultats et conclusions

Pour évaluer les résultats du processus automatique par rapport au contrôle interactif, nous utilisons les notions mathématiques de précision [1] et de rappel [2].

Les tableaux suivants montrent une synthèse des résultats obtenus. Nous présentons en premier lieu les résultats globaux obtenus pour chacun des produits P1, P2 et P3.

Produit d'évolutions	Nombre d'évolutions	Nombre de conflits à détecter	Nombre de conflits détectés	Précision	Rappel
Produit P1	14 évolutions	17	19	89%	100%
Produit P2	21 évolutions	36	36	100%	100%
Produit P3	41 évolutions	43	45	93%	98%

Table 4.8 – Evaluation du contrôle de concurrence entre les produits d'évolutions et les données de l'acteur

Nous présentons dans le tableau 4.9 les résultats obtenus sur la couche ROADL pour chacun des produits P1, P2 et P3. Nous détaillons en particulier, les résultats obtenus pour les conflits de mise à jour et les conflits de création du produit P3.

1. La précision est le nombre de bons résultats obtenus sur le nombre total de résultats obtenus
2. Le rappel est le nombre de bons résultats obtenus sur le nombre de bons résultats souhaités

Produit d'évolutions	**Nombre d'évolutions**	**Nombre de conflits à détecter**	**Nombre de conflits détectés**	**Précision**	**Rappel**
Produit P1	5 évolutions	7	7	100%	100%
Produit P2	21 évolutions	14	14	100%	100%
Produit P3	34 évolutions	33	35	91%	97%
Produit P3	15 mises à jour	15	15	100%	100%
Produit P3	19 créations	4 conflits	6 conflits mais 3 inexacts	50%	75%

Table 4.9 – Evaluation du contrôle de concurrence pour les objets de la couche `ROADL`

L'analyse des résultats montre que, comme nous l'avions supposé, l'usage d'identifiants facilite fortement le contrôle de concurrence pour des conflits de mise à jour intervenant sur des évolutions de type modification ou suppression. Cependant, cela suppose un effort de transformation des jeux de données militaires pour lesquels les identifiants ne sont pas à ce jour considérés. Nous pensons néanmoins que cela apporte un bénéfice non négligeable lors de la réalisation de la stratégie d'intégration au sein de l'infrastructure militaire et nous conseillons par conséquent d'adapter les jeux de données utilisés dans l'infrastructure dès le début de la mission afin que des identifiants existent et soient exploitables par la suite.

Par ailleurs, nous devons effectuer des tests complémentaires concernant les conflits topologiques, car il s'avère que toutes les intersections détectées entre les différents thèmes ne provoquent pas systématiquement un conflit (une route peut très bien croiser une rivière si un pont existe au point d'intersection). Nous devons donc examiner, avant toute réconciliation, l'authenticité ou non du conflit, grâce aux spécifications et aux métadonnées attachées aux jeux de données et aux évolutions. En effet, si nous ne faisions pas cette vérification, nous serions amenés à supprimer des données ou des évolutions qui s'avéreraient ne pas être conflictuelles, ce qui fausserait considérablement le résultat.

Enfin, nous constatons que le processus de contrôle de concurrence que nous avons mis en place détecte quelques conflits inexacts par rapport au contrôle visuel effectué par un expert (précisions des produits P1 et P3). En particulier, nous voyons que le processus détecte plus de conflits de création qu'il n'en existe et que ces conflits n'ont parfois pas lieu d'être dans la réalité (la valeur de la précision obtenue pour les évolutions de type création effectuées sur la couche `ROADL` du produit P3 montre ce phénomène). Cela est du au fait que les conflits de création sont repérés grâce à une technique d'appariement géométrique pour lequel nous devons définir un seuil à partir duquel nous considérons les données appariées. En fonction de cette valeur, nous obtenons plus ou moins d'objets appariés.

La figure 4.4 montre un exemple de conflit de création correctement détecté entre une évolution du produit P2 et une donnée du jeu de données de référence

(appariement correct).

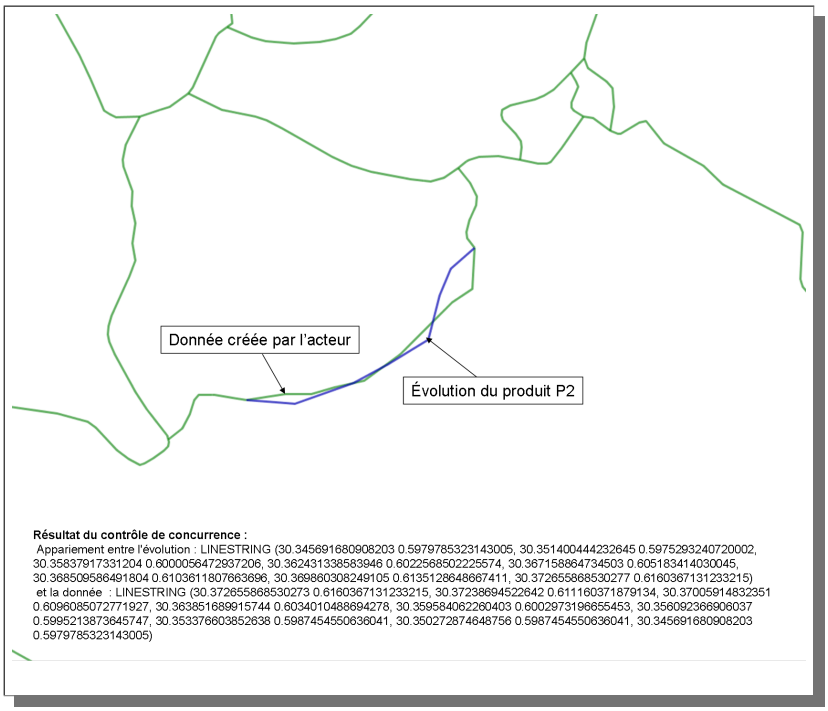

Figure 4.4 – Exemple d'un conflit de création correctement détecté par le processus

La figure 4.5 montre quant à elle, un exemple d'un conflit de création incorrectement détecté entre une évolution du produit P3 et une donnée du jeu de données de référence (appariement incorrect). En effet, nous voyons que le processus effectue un appariement entre l'évolution du produit P3 et la donnée créée par l'acteur alors que ces deux routes sont en réalité distinctes.

La difficulté réside ici dans la définition de la valeur du seuil qui permet d'obtenir des résultats optimaux. En particulier, nous devons trouver des valeurs qui reconnaissent un maximum d'appariements cohérents sans ajouter des appariements qui s'avère être faux (la valeur du rappel obtenue pour les évolutions de type création effectuées sur la couche `ROADL` du produit P3 montre ce type de problème).

Nous avons utilisé les résultats de l'étude du projet Envol VDC qui traite des mécanismes d'appariement pour les données vectorielles militaires [Raynal et Ruffier, 2005], pour fixer et ajuster les seuils des appariements dans ces tests. Nous avons pour cela effectué plusieurs tests en fonction de seuils différents avant d'obtenir des appariements qui nous semblaient apporter les meilleurs résultats en fonction de notre contexte d'application.

La solution que nous avons finalement choisie permet de détecter un nombre acceptable de conflits réels tout en limitant le nombre d'appariements incorrects. Cependant, nous pouvons assurer à la vue de ces résultats que la détection de conflits par appariement s'avère être la partie la plus difficile à automatiser.

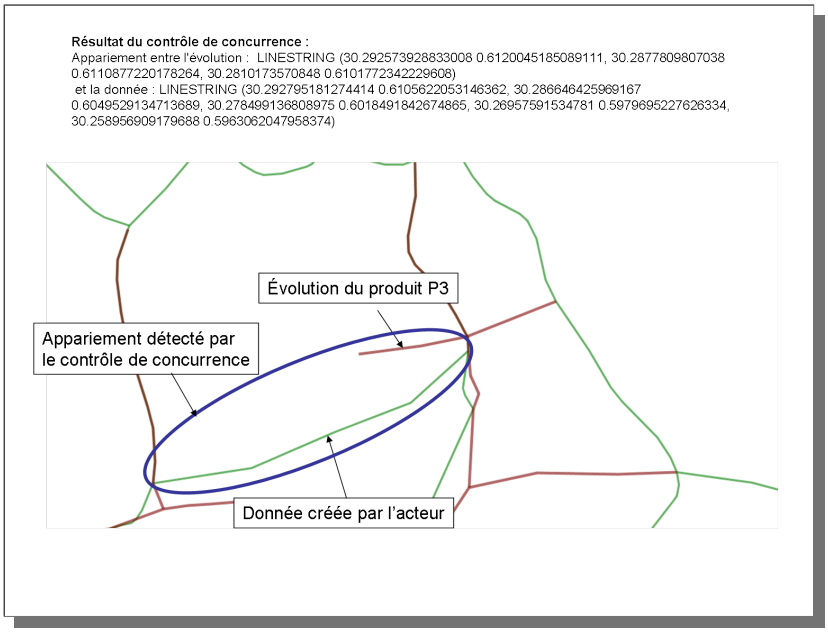

Figure 4.5 – Exemple de conflit de création incorrectement détecté par le processus

4.3.2 Réconciliation des données conflictuelles

Le rôle du processus de réconciliation est de traiter les données concurrentes qui ne peuvent par conséquent pas être insérées conjointement dans un jeu de données au risque de créer un conflit d'intégration. Son but est de déterminer le meilleur choix au vue de différents paramètres tels que les besoins de l'acteur qui utilisera les données ou encore les contraintes à respecter.

Mise en oeuvre

Nous nous plaçons ici dans la situation où une donnée et une évolution ont été déclarées conflictuelles et doivent être traitées. Pour être au plus proche de la réalité contextuelle de notre étude et estimer au mieux les résultats de la réconciliation, nous simulons les deux cas suivants : premièrement, l'acteur de référence est un opérationnel sur le terrain d'action qui a des besoins qui peuvent évoluer dans le temps. Dans la seconde situation, l'acteur de référence est un producteur situé sur le terrain d'action dont les besoins ont été fixés dès le début de la mission et ne peuvent changer. Cette façon de procéder nous permet d'illustrer le processus de réconciliation en fonction des contraintes propres à chaque acteur et en fonction des niveaux de cohérence inhérents à leur rôle dans l'infrastructure.

L'évolution conflictuelle est fournie dans un produit d'évolutions contenant plusieurs thèmes mis à jour (`ROADL, AEROFACP, BUILDA, WATRCRSL`). C'est une évolution de type création effectuée sur le thème `ROADL`. Elle a été saisie à partir du produit de référence, par un opérationnel situé sur le terrain d'action, pendant une campagne de mise à jour. Les informations suivantes caractérisent l'ensemble des évolutions de la couche `ROADL` :

- La précision géométrique des évolutions est comprise entre 150 et 200m,
- 50% des attributs ont été renseignés,
- Les évolutions saisies pour ce thème concernent 42% du nombre total d'objets contenus dans le produit de référence,
- La fiabilité de l'acteur ayant effectué la mise à jour est estimée à 50%,
- La date d'actualité est le 15/03/2008,

Par ailleurs, la donnée conflictuelle appartient à la couche `ROADL` du jeu de données de l'acteur de référence. Cette donnée est une mise à jour qui a été effectuée par l'acteur et qui a déjà été intégrée à son jeu de données. Les informations suivantes caractérisent les données de la couche `ROADL` :

- La précision géométrique des données est de 150 m,
- Tous les attributs ont été renseignés,
- Les données de cette couche thématique représente 10% du nombre total d'objets contenus dans le produit de référence,
- La fiabilité est estimée à 50% si l'acteur qui a saisi la donnée est un opérationnel, à 75% si l'acteur est un producteur,
- La date d'actualité est le 05/01/2008,

La première étape du processus de réconciliation consiste à récupérer les informations contenues dans les métadonnées de chacune des données conflictuelles, puis de calculer une mesure de qualité pour chacune d'entre elle. La mesure de qualité est calculée en fonction des besoins de l'acteur de référence et du niveau de cohérence dépendant de son rôle dans l'infrastructure.

Les besoins de l'acteur de référence que nous utilisons pour notre simulation sont les suivants :

- La précision géométrique des données doit être inférieure ou égale à 500 m,
- Au moins la moitié des attributs doit être renseignée,
- Les données ou évolutions de cette couche thématique doivent représenter au moins 10% du nombre total d'objets que l'acteur manipule,
- La fiabilité accordée à l'acteur en charge de la saisie doit être supérieure ou égale à 40%
- La date d'actualité doit être supérieure au 01/02/2008,

Pour chaque caractéristique, le processus calcule une mesure de qualité, puis normalise cette mesure afin d'obtenir un résultat compris entre -1 et 1. Plus la mesure qualité est proche de -1, meilleure est la qualité externe.

Le tableau 4.10 montre le résultat des mesures qualités normalisées obtenues pour chacune des caractéristiques attachées à la donnée conflictuelle.

Caractéristique	**Valeur effective**	**Valeur souhaitée**	**Mesure qualité normalisée**
Précision géométrique	150m	$\leqslant$ 500m	-0.777
Précision attributaire	100%	$\geqslant$ 50%	-0.33
Fiabilité si opérationnel	50%	$\geqslant$ 40%	-0.1
Fiabilité si producteur	75%	$\geqslant$ 40%	-0.35
Exhaustivité	10%	$\geqslant$ 10%	0
Actualité	05/01/08	$\geqslant$ 01/02/08	+0.06

Table 4.10 – Mesures qualités des caractéristiques relatives à la donnée conflictuelle

Le tableau 4.11 montre le résultat des mesures qualités normalisées obtenues pour chacune des caractéristiques attachées à l'évolution conflictuelle.

Caractéristique	**Valeur effective**	**Valeur souhaitée**	**Mesure qualité normalisée**
Précision géométrique	$150m < x < 200m$	$\leqslant$ 500m	-0.43
Précision attributaire	50%	$\geqslant$ 50%	0
Fiabilité	50%	$\geqslant$ 40%	-0.1
Exhaustivité	42%	$\geqslant$ 10%	-0.32
Actualité	15/03/08	$\geqslant$ 01/02/08	-0.4

Table 4.11 – Mesures qualités des caractéristiques relatives à l'évolution conflictuelle

Le processus calcule ensuite la mesure qualité globale pour chaque donnée conflictuelle. Pour cela, le processus affecte dans un premier temps un poids à chaque caractéristique en fonction du niveau de cohérence qui est souhaité. Pour cette simulation, nous avons considéré deux types d'acteurs de référence : un producteur et un opérationnel. Le niveau de cohérence du producteur est plus élevé que celui de l'opérationnel. D'autre part, le producteur souhaite plutôt obtenir des données fiables avec une précision géométrique proche de son besoin plutôt qu'un grand nombre de données qui ne seraient pas de qualité. Le poids affecté à la précision géométrique et à la fiabilité sera donc plus important que celui de l'exhaustivité. En revanche, l'opérationnel souhaite obtenir rapidement de l'information récente quel que soit la qualité de celles ci. Le poids attribué à l'exhaustivité et à l'actualité sera donc plus important que celui de la fiabilité ou des précisions géométriques et attributaires. Le processus calcule enfin la moyenne pondérée afin d'obtenir une mesure qualité globale pour chacune des données conflictuelles. Cette mesure dépend donc des besoins de l'acteur, du niveau de cohérence souhaité et permet au processus de proposer un choix entre les données concurrentes. Avec les valeurs prises pour notre

simulation, nous obtenons les résultats suivants :

	Producteur	**Opérationnel**
Évolution	-0.2246	-0.3118
Donnée	-0.437	-0.17

Table 4.12 – Mesures de qualité globales en fonction de l'acteur de référence

Analyse des résultats et conclusions

Les mesures de qualité globales obtenues suite à cette simulation montrent des résultats qui diffèrent selon le rôle de l'acteur dans l'infrastructure (et donc le niveau de cohérence attendu). En effet, nous voyons en particulier que l'évolution sera mieux adaptée aux besoins d'un opérationnel alors que la donnée conviendra mieux à un producteur. Le processus aura donc une préférence pour l'une ou l'autre donnée conflictuelle en fonction de l'acteur qui utilisera finalement la donnée après intégration.

Ces résultat prouve qu'il est possible de développer un processus de réconciliation qui utilise les informations présentes dans les métadonnées afin de proposer un choix entre des données conflictuelles en considérant d'une part les besoins et les contraintes de l'acteur fixés en fonction de leur rôle dans l'infrastructure, et d'autre part les niveaux de cohérence souhaités pour les différents jeux de données qui seront exploités.
Nous devons toutefois nuancer cette affirmation car les résultats que nous avons obtenus pour ces données sont issus d'une simulation et sont donc probablement biaisés. Des tests supplémentaires sur des données et des métadonnées réelles permettraient d'appuyer les résultats obtenus par simulation et de renforcer notre conviction sur la faisabilité de ce processus.

Chapitre 5

Conclusions et perspectives

5.1 Conclusions

Le contexte d'étude de ce travail de thèse était celui d'un environnement militaire réparti dans lequel des acteurs sont déployés sur plusieurs sites et doivent coopérer les uns avec les autres. Des données géographiques répliquées sur chacun des sites sont utilisées pendant toute la durée de l'opération et peuvent évoluer en fonction des besoins de chaque unité. Les évolutions sont saisies dans différentes conditions, par différents auteurs, à différentes périodes. Elles sont par conséquent plus ou moins fiables et possèdent de ce fait des qualités qui divergent. Les mises à jour et nouvelles données doivent être régulièrement mises à disposition des autres acteurs afin qu'une collaboration efficace puisse avoir lieue pendant toute la durée de la mission.

5.1.1 Analyse de la problématique

L'analyse du scénario de démonstration utilisé dans le projet Envol VDC a montré la complexité des échanges intervenant entre les acteurs participant à une mission opérationnelle et a permis d'exposer la difficulté de la gestion de la mise à jour des jeux de données spatiaux dans un univers militaire réparti. En particulier, nous avons souligné qu'à cause de la mise à jour simultanée, l'intégration sans ménagement des évolutions peut provoquer des incohérences et remettre en cause l'intégrité des bases de données spatiales.

Plus généralement, notre travail de recherche se situe dans le cadre d'une application collaborative asynchrone utilisant un protocole de réplication optimiste de données géographiques dans un contexte de prise de décision. Plus particulièrement, nous nous intéressons à la mise à jour d'un jeu de données géographique par des évolutions provenant de sources multiples. Ces évolutions ne sont pas nécessairement pertinentes pour l'utilisateur final, peuvent être en conflits et créer des incohérences si elles sont intégrées sans précautions.

La problématique générale de ce travail concerne la gestion dans un contexte militaire de la mise à jour de données géographiques répliquées. Pour y parvenir,

nous avons montré qu'il était nécessaire de définir une **infrastructure de données spatiales** appropriée au contexte militaire, **vérifier la pertinence** des évolutions en fonction des attentes des différents acteurs, proposer un mécanisme de **détection des mises à jour conflictuelles lié au type de données** et utiliser des procédures de **réconciliation des écritures divergentes adaptées aux besoins des unités** participant à la mission. Notre objectif final étant de proposer des solutions permettant l'intégration cohérente et autant que possible automatique, des mises à jour de données spatiales dans un environnement de réplication optimiste, multimaîtres et asynchrone.

Nous avons dans un premier temps montré qu'il était nécessaire de définir une politique de gestion des évolutions échangées entre les différents acteurs de la mission. En particulier, nous avons spécifié la nature des évolutions saisies par les différents acteurs, le format d'échange utilisé entre les différentes unités et les informations nécessaires pour que les évolutions puissent être correctement exploitées.

Par ailleurs, nous avons vu qu'il est possible qu'un utilisateur ait à intégrer de nombreux ensembles d'évolutions provenant de sources multiples, ce qui implique que les mises à jour ne sont pas forcément toutes pertinentes pour son application. Une étude permettant d'évaluer la pertinence des mises à jour multi-sources était alors nécessaire pour exclure les évolutions non adéquates et ne retenir que celles qui soient indispensables à l'utilisateur.

Ensuite, nous avons souligné que nous devions être en mesure de détecter les évolutions qui risqueraient de perturber la cohérence du jeu de données. Pour cela, nous avons d'abord défini ce qu'est la cohérence des données spatiales et ensuite répertorié les éventuels conflits qui pourraient remettre en cause cette cohérence.

Enfin, notre étude se devait de répondre aux problèmes liés à la réconciliation des données conflictuelles. En particulier, nous souhaitions prendre en considération les différents points de vue des personnes ayant saisi les évolutions afin de proposer le meilleur choix en fonction des besoins de l'utilisateur final. Le but étant de nous nous assurer qu'après l'intégration des évolutions, l'utilisateur possédera finalement les « bonnes » données dans le sens les moins erronées possibles et les plus adaptées à son besoin immédiat.

5.1.2 SDI et métadonnées

La solution que nous avons proposée dans ce travail pour traiter les évolutions provenant de sources multiples dans un contexte de mission militaire s'appuie sur un **infrastructure de données spatiales** et sur une **stratégie d'intégration des évolutions** utilisant des métadonnées normalisées. En effet, nous pensons que la mise en place d'une infrastructure spécifique au contexte militaire apporte un cadre nécessaire à l'élaboration de politiques permettant de gérer au mieux les échanges entre les acteurs prenant part à l'infrastructure. Par ailleurs, nous pensons que l'utilisation des métadonnées permet d'aider l'intégration des évolutions multi-sources, notamment lors de la vérification de la cohérence.

Nous avons donc proposé d'utiliser une infrastructure militaire dans laquelle nous avons déterminé les acteurs, spécifié les données et caractérisé les évolutions. Nous avons d'abord précisé les rôles et objectifs de chacun des acteurs participant à une mission opérationnelle. Puis, nous avons déterminé les types et schémas des données utilisés dans l'infrastructure. En particulier, nous avons choisi d'**ajouter des identifiants** qui n'existaient pas initialement sur les données. Ensuite, nous avons établi une **politique de gestion des évolutions** et avons défini un format de livraison permettant l'échange structuré des produits contenant uniquement les changements survenus sur les données. Enfin, nous avons formalisé les échanges possibles entre ces trois entités dans le **modèle « Données, Acteurs, Évolutions »** .

A partir de ce modèle, nous avons ensuite stipulé, pour chaque entité, l'information supplémentaire qu'il faut ajouter pour une exploitation optimale. Nous avons donc spécifié les métadonnées qui doivent être attachées à chacune des entités. Nous avons pour cela montré que l'utilisation de **métadonnées normalisées** favorise l'interopérabilité. Pour les données, nous avons choisi de nous appuyer sur le format METAFOR qui est le format de métadonnées utilisé par l'armée française. Puis, pour la gestion des évolutions de données militaires, nous avons créé un **profil de métadonnées MUMSDI** conforme aux recommandations de la norme ISO 19115. Enfin, pour les acteurs, nous avons défini un ensemble de métadonnées, en distinguant les informations relatives aux besoins de celles relatives aux contraintes.

5.1.3 Stratégie d'intégration des évolutions

Grâce à toutes ces informations, nous avons pu mettre en place une **stratégie d'intégration des évolutions multi-sources** qui peut être utilisée sur tous les sites et par tous les acteurs de l'infrastructure militaire. Cette stratégie est divisée en plusieurs étapes, chacune permettant de traiter un problème relatif à l'échange et à l'intégration de multiples évolutions.
Nous avons premièrement montré le besoin d'**évaluer la pertinence** des évolutions pour filtrer celles qui ne correspondent pas aux attentes de l'utilisateur final.
Ensuite, nous avons défini un processus de **vérification de la cohérence**. Pour cela, nous avons d'abord défini ce qu'est la cohérence dans notre contexte d'étude et nous avons déterminé les différents niveaux de cohérence à atteindre en fonction des rôles des acteurs. Puis, nous avons développé le processus de vérification de la cohérence en détaillant d'une part le **contrôle de concurrence** qui permet de détecter les données conflictuelles et d'autre part, la **réconciliation** qui permet de choisir entre deux données conflictuelles celle qui sera finalement intégrée. Nous avons montré que l'utilisation des identifiants facilitait le contrôle de concurrence et permettait d'obtenir des résultats satisfaisants. Nous avons également montré l'intérêt pendant la phase de réconciliation d'utiliser des métadonnées normalisées afin de fournir l'information utile pour effectuer le meilleur choix en fonction de critères diverses et propres à chaque unité.
Enfin, couplé au processus de vérification de la cohérence, nous avons choisi d'utiliser des **sessions de mises à jour** pour effectuer une intégration cohérente

des évolutions dans le jeu de données de l'acteur de référence. Nous avons en outre montré les enchaînements entre ces deux dernières étapes de la stratégie d'intégration.

5.1.4 Résultats

Nous avons effectué une série de tests afin de valider nos choix théoriques. Nous avons utilisé des données vectorielles militaires et avons expérimenté essentiellement la vérification de la cohérence et spécialement le contrôle de concurrence et la réconciliation. Nous avons en particulier, prouvé que le contrôle de concurrence est efficace pour les évolutions de type mise à jour et suppression, notamment grâce à l'utilisation des identifiants, mais qu'il reste problématique pour les évolutions de type création du fait de l'utilisation de l'appariement. Par ailleurs, nous avons aussi montré que le mécanisme de réconciliation fournit des résultats cohérents lorsque suffisamment d'informations sont disponibles avec les données et les évolutions. Cependant, les tests ayant été effectués sur des évolutions et avec des métadonnées simulées, nous ne pouvons malheureusement pas nous assurer de l'exactitude de ces résultats dans une situation réelle. Néanmoins, les résultats de ces expérimentations tendent à nous conforter dans notre analyse et dans la décision des choix effectués pour traiter la problématique de cette thèse.

Finalement, nous pouvons dire que les principaux bénéfices apportés par cette méthode sont :

- La mise en place d'un SDI adapté au contexte militaire dans lequel une politique des gestion des évolutions est applicable.
- La formalisation d'un modèle de métadonnées pour la gestion des évolutions, normalisé favorisant ainsi l'interopérabilité
- La prise en compte des besoins des utilisateurs afin de mesurer l'adéquation des évolutions à l'utilisation qui va en être faite.
- L'utilisation des identifiants pour faciliter le contrôle de concurrence
- L'élimination des conflits grâce à un protocole de réconciliation utilisant les informations contenues dans les métadonnées
- L'intégration des évolutions pertinentes et dont on est assuré qu'elles ne provoqueront pas d'incohérences

5.2 Perspectives

Les améliorations que nous pourrions apporter à ce travail concernent plusieurs points de la stratégie d'intégration.
En effet, nous pensons qu'une étude approfondie de l'utilisation des mécanismes d'appariement pour la détection de conflits corrigerait la tendance du processus en charge du contrôle de concurrence à détecter des conflits en excès. Nous pourrions par exemple utiliser des outils d'appariements qui analysent la structure d'un réseau comme ceux définis par [Devogele, 1997].
Ensuite, nous croyons que l'automatisation du remplissage des métadonnées lors

de la saisie des évolutions permettrait d'obtenir un nombre optimal d'informations exploitables par le processus de réconciliation. Nous pourrions étudier les travaux de [Libourel, 2003] et en particulier la thèse de [Barde, 2005] qui s'appuie sur un SGBDR afin de contrôler ou d'automatiser la saisie de certains éléments de métadonnées.
Le développement d'une IHM pour gérer l'interactivité nous semble également être une bonne piste pour améliorer la stratégie d'intégration. Nous pourrions par exemple nous inspirer des travaux de [Devillers, 2004] et utiliser des indicateurs de qualité qui permettraient à l'acteur d'effectuer un choix pertinent lors de la réconciliation des données conflictuelles.
Enfin, nous supposons que les calculs effectués dans la phase de réconciliation mériteraient d'être approfondis afin d'obtenir de meilleures mesures de qualité. Par exemple, une étude approfondie auprès des militaires concernant l'importance qu'il faudrait accorder aux éléments caractéristiques en fonction du type d'acteurs et du niveau de cohérence souhaité serait utile pour exploiter au mieux la notion de pondération.

Par ailleurs, des tests avec des évolutions et des métadonnées non simulées permettraient de valider les choix de conception dans un cadre réel. Il serait également intéressant de mettre en place le profil MUMSDI et de l'utiliser lors d'une mission militaire afin de montrer l'utilité de ces métadonnées pour la stratégie d'intégration dans un cas réel.

Enfin, les perspectives plus générales de ce travail sont multiples car beaucoup de pistes restent à explorer concernant la mise à jour des bases de données géographiques réparties. Nous avons choisi de nous appuyer sur un SDI pour gérer la mise à jour des jeux de données spatiaux répartis car nous pouvons contextuellement déterminer un cadre dans lequel nous connaissons un certain nombre d'informations telles que les acteurs ou les données. Cette façon de faire ne pourrait être utilisée dans un contexte plus général et d'autres alternatives comme la définition de contraintes évoquée dans [Kermarrec *et al.*, 2001] pourraient être utilisées pour permettre une réconciliation efficace.
Enfin, de nombreux travaux restent à faire concernant l'adéquation aux besoins des utilisateurs et permettraient d'améliorer nos résultats concernant le filtrage des évolutions non pertinentes et la réconciliation des données conflictuelles.

Annexe A

La norme ISO 19115

A.1 Sections et entités de métadonnées

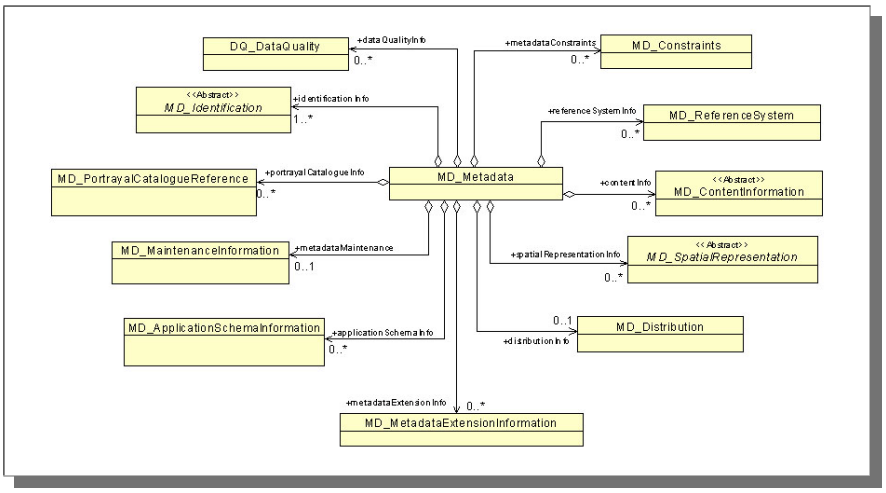

FIGURE A.1 – Principales entités de l'ISO 19115

Le point d'entrée de la norme est le package `Metadata entity set information`. Ce package dont l'unique classe est la classe `MD_Metadata` est obligatoire (Cf. figure A.1). Cette classe fournit l'ensemble des informations relatives aux métadonnées (domaine d'application, date de création, nom et version du standard de métadonnées utilisé..). Toutes les autres informations présentes dans la norme sont agrégées à cette classe.

Les informations nécessaires à l'identification des données sont situées dans le package `Identification Information` qui a pour point d'entrée la classe abstraite `MD_Identification` qui est obligatoire. On y trouve entre autres, la description des données, un aperçu, le mode de représentation spatial utilisé ...

Le package `Data quality information` contient l'évaluation générale de la qualité des données et des jeux de données. La classe `DQ_Quality` qui est optionnelle, permet d'accéder à ces informations. Cette classe se dérive en deux sous classes pour fournir d'une part des informations de généalogie (`LI_Lineage`), et d'autre

part des informations quantitatives sur la qualité telles que la précision ou encore la cohérence des données (`DQ_Element`).

La section `Maintenance information` donne la fréquence et la portée des mises à jour. Elle est accessible par le biais de la classe `MD_MaintenanceInformation` qui est optionnelle. On y trouve des informations sur la fréquence, l'étendue et la date de la prochaine mise à jour. Cette classe permet aussi à l'utilisateur de choisir la période envisagée pour les futures mises à jour.

Les informations sur le distributeur des données et sur les moyens mis à disposition pour obtenir une ressource sont fournies dans le package `Distribution information` dont le point d'entrée est la classe `MD_Distribution` qui est optionnelle. Cette classe permet de connaître le média de stockage des données, utile pour savoir si les ressources sont disponibles sur le réseau ou non. Elle renseigne également sur le distributeur, le coût et la disponibilité d'un jeu de données.

Les informations sur l'étendue spatiale, temporelle et verticale du jeu de données sont données dans la section `Extent information` via la classe `EX_Extent` qui est optionnelle.

Les contraintes associées aux données se trouvent dans le package `Constraint information` et sont accessibles par la classe `MD_Constraint` (optionnelle) qui renseigne sur les restrictions d'usage appliquées aux jeux de données (copyright, licence,...) ainsi que sur le niveau de confidentialité des données (confidentiel, top secret, ...).

La section `Citation and responsible party information` contient les classes `CI_Citation` et `CI_ResponsibleParty` qui permettent de citer une ressource mais aussi de donner des informations sur l'organisation responsable de la ressource. Cette section est optionnelle.

La classe `MD_SpatialRepresentation` présente dans le package `Spatial representation information` décrit les mécanismes utilisés pour représenter l'information spatiale dans un jeu de données. Cette section est optionnelle.

La section `Reference system information` par le biais de la classe `MD_ReferenceSystem` fournit la description des systèmes de référence spatiaux et temporels utilisés dans le jeu de données. Cette section est optionnelle.

Le package `Content information` a pour point d'entrée la classe `MD_ContentInfo` qui contient l'information permettant d'identifier le catalogue d'objets utilisé et toute autre information concernant la couverture du jeu de données. Cette section est optionnelle.

La classe `MD_PortrayalCatalogueReference` permet d'identifier le catalogue de référence utilisé. Cette classe est optionnelle et appartient à la section `Portrayal catalogue information`.

Le schéma d'application utilisé pour construire le jeu de données est contenu dans le package `Application schema information` et spécifié dans la classe `MD_ApplicationSchemaInformation`. Cette section est optionnelle.

Enfin, il existe le package `Metadata extension information` permettant d'étendre la norme en fonction du besoin spécifique de l'utilisateur qui est accessible via la classe `MD_MetadataExtensionInformation` qui est optionnelle. Il fournit en particulier le nom, la définition et les conditions d'utilisation des nouveaux éléments de métadonnées.

A.2 Les métadonnées de qualité

Les métadonnées de qualité sont accessibles via la section `Data quality information`. Un ensemble de métadonnées peut contenir plusieurs instances de la classe `DQ_DataQuality` (Cf. figure A.2). Chaque instance de la classe `DQ_DataQuality` est caractérisée par un champ d'application (attribut `scope` de type `DQ_Scope`) qui spécifie la nature des données cibles, en particulier le niveau d'application des métadonnées (attribut `level` dont les valeurs possibles sont fournies dans la liste de code `MD_ScopeCode`) et la zone géographique concernée (attribut `extent` de type `EX_Extent`).

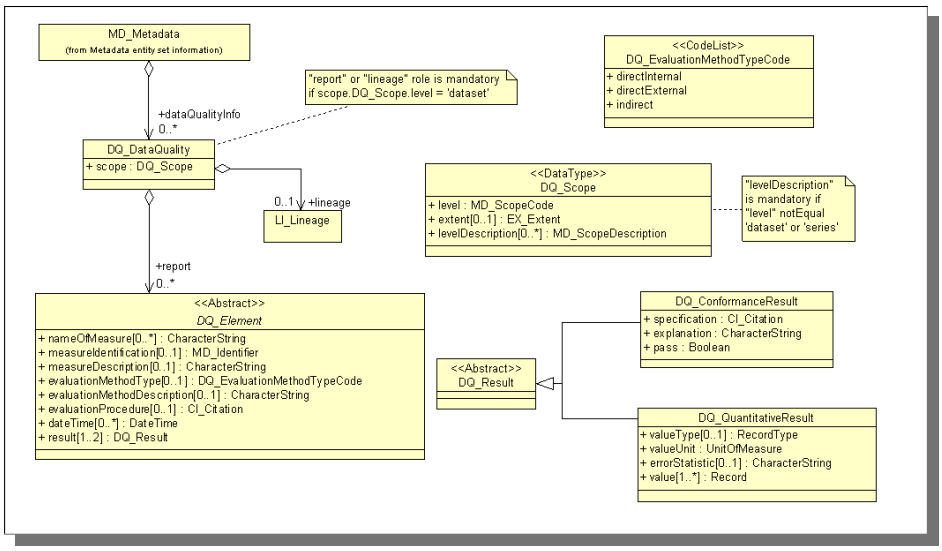

Figure A.2 – Informations de qualité dans ISO 19115

`DQ_DataQuality` se spécialise en `LI_Lineage` pour fournir les informations de généalogie et en `DQ_Element` pour présenter les mesures de qualité (Cf. figure A.3).

La classe `LI_Lineage` renseigne sur la nature des données sources (`LI_Source`) et sur le processus de production (`LI_ProcessStep`) ayant conduit à la création du

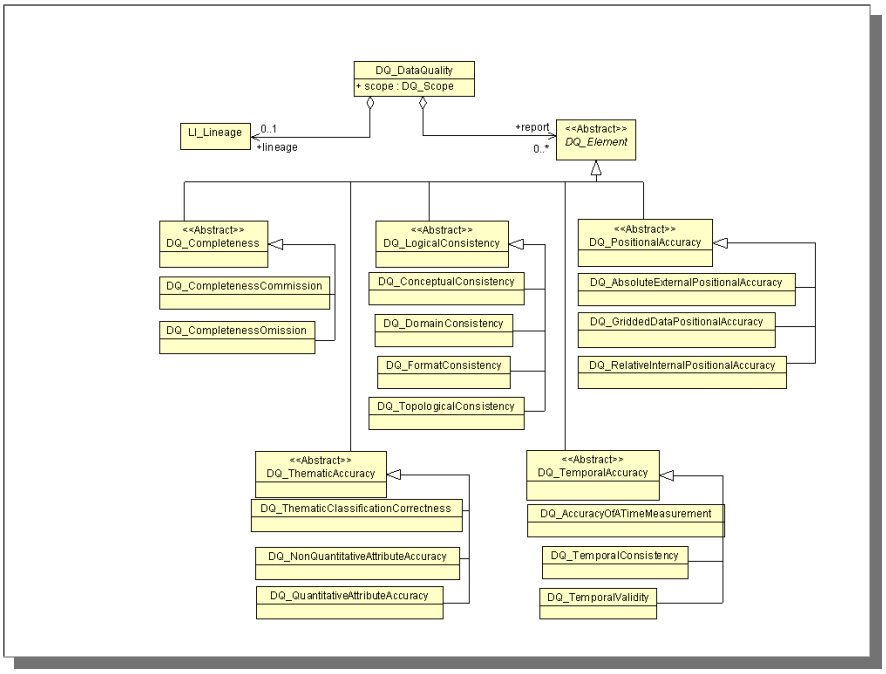

Figure A.3 – Classes pour la représentation de qualité dans ISO 19115

jeu de données.

La classe `DQ_Element` fournit un certain nombre d'informations sur les tests effectués pour mesurer la qualité telle que la méthode utilisée, la période pendant laquelle le test a été fait et surtout les résultats obtenus.
Elle est divisée en sous-classes représentant chacune un critère de qualité :

`DQ_Completeness` qualifie l'exhaustivité du jeu de données. Elle se décompose en deux sous-classes :
- `DQ_CompletnessCommission` décrit les données excédentaires du jeu de données.
- `DQ_CompletnessOmission` décrit les données manquantes du jeu de données.

`DQ_ThematicAccuracy` fournit l'information sur la qualité des attributs. Elle se décompose en trois sous-classes :
- `DQ_ThematicClassificationCorrectness` décrit la cohérence des attributs
- `DQ_NonQuantitativeAttributeAccuracy` décrit la justesse des attributs non-quantitatifs
- `DQ_QuantitativeAttributeAccuracy` décrit la précision des attributs quantitatifs

`DQ_LogicalConsistency` donne le degré d'adhésion aux règles logiques. Elle se décompose en quatre sous-classes :
- `DQ_ConceptualConsistency` indique la conformité par rapport au schéma conceptuel
- `DQ_DomainConsistency` indique la conformité par rapport au domaine de va-

leurs
- DQ_FormatConsistency indique la conformité par rapport au format
- DQ_TopologicalConsistency informe sur le degré de cohérence topologique.

DQ_TemporalAccuracy fournit l'information sur la précision temporelle. Elle se décompose en trois sous-classes :
- DQ_AccuracyOfTimeMeasurement donne la précision d'une mesure temporelle
- DQ_TemporalConsistency définit le degré de cohérence temporelle
- DQ_TemporalValidity définit la validité temporelle.

DQ_PositionalAccuracy fournit l'information sur la précision de position. Elle se décompose en trois sous-classes :
- DQ_AbsoluteExternalPositionalAccuracy indique la précision absolue
- DQ_GriddedDataPositionalAccuracy indique la précision absolue pour les données maillées
- DQ_RelativeInternalPositionalAccuracy indique la précision relative

A.3 Le noyau de la norme ISO 19115

On dénombre sept éléments de la norme ISO 19115 obligatoires dans le noyau :
- MD_Metadata > MD_DataIdentification.citation > CI_Citation.title qui spécifie le nom avec lequel la ressource est généralement connue.
- MD_Metadata > MD_DataIdentification.citation > CI_Citation.date qui donne la ou les dates de référence pour la ressource en question (date de publication, de création et/ou de révision).
- MD_Metadata > MD_DataIdentification.language indique la langue utilisée dans le jeu de données.
- MD_Metadata > MD_DataIdentification.topicCategory qui fournit le ou les principaux thèmes relatifs au jeu de données.
- MD_Metadata > MD_DataIdentification.abstract qui précise le contenu des ressources par un court résumé.
- MD_Metadata.dateStamp qui signale la date de création des métadonnées
- MD_Metadata.contact > CI_ResponsibleParty qui donne les renseignements relatifs à l'organisme responsable des informations que contiennent les métadonnées.

Il y a quatre éléments de la norme qui sont conditionnels dans le noyau :
- MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicBoundingBox ou EX_GeographicDescription. L'un ou l'autre devient obligatoire si l'élément de métadonnées MD_Metadata.hierarchyLevel prend la valeur « dataset ».
- MD_Metadata > MD_DataIdentification.characterSet est rendu obligatoire si le jeu de caractère utilisé n'est pas défini par le standard ISO 10646-1.
- MD_Metadata.characterSet est rendu obligatoire si le jeu de caractère utilisé n'est pas défini par le standard ISO 10646-1, ni par le standard de codage.

– MD_Metadata.language est rendu obligatoire si la langue utilisée pour documenter les métadonnées n'est pas définie dans le standard de codage.

Enfin, le noyau de la norme ISO 19115 contient douze éléments optionnels :
– MD_Metadata > MD_DataIdentification.pointOfContact > CI_ResponsibleParty qui permet d'identifier les personnes ou organisations associées aux ressources et de spécifier les moyens de communications pour les contacter.
– MD_Metadata > MD_DataIdentification.spatialResolution > MD_Resolution.equivalentScale ou MD_Resolution.distance qui donne une indication générale sur la densité des données spatiales dans le jeu de données.
– MD_Metadata > MD_DataIdentification.extent.EX_Extent > EX_TemporalExtent ou EX_VerticalExtent qui établit les étendues temporelles ou verticales du jeu de données.
– MD_Metadata > MD_DataIdentification.spatialRepresentationType qui révèle la méthode utilisée pour représenter spatialement l'information géographique.
– MD_Metadata > MD_Distribution > MD_Format.name qui donne le nom du ou des formats utilisés pour le transfert de données.
– MD_Metadata > MD_Distribution > MD_Format.version qui donne la version du ou des formats utilisés pour le transfert des données.
– MD_Metadata > MD_Distribution > MD_DigitalTransferOption.onLine > CI_OnlineResource qui fournit les informations sur les sources en ligne depuis lesquelles la ressource peut être obtenue.
– MD_Metadata > DQ_DataQuality > LI_Lineage qui contient l'information sur les données sources utilisées ou sur les étapes du processus ayant servies à construire la ressource.
– MD_Metadata > MD_ReferenceSystem qui décrit les systèmes de références spatiaux et temporels utilisés dans le jeu de données.
– MD_Metadata.fileIdentifier qui est l'identifiant unique du fichier de métadonnées
– MD_Metadata.standardName qui spécifie le nom du standard de métadonnées utilisé
– MD_Metadata.standardVersion qui définit la version du standard de métadonnées utilise.

A.4 Normes associées à la norme ISO 19115

Les normes associées à l'ISO 19115 sont présentées dans la figure A.4, où :

ISO 639 : Codes pour la représentation des noms de langages.
ISO 3166 : Codes pour la représentation des noms de pays et de leurs subdivisions
ISO 4217 : Codes pour la représentation des monnaies et types de fonds
ISO 8859 (parties 1 à 16) : Technologies de l'information - Jeux de caractères graphiques codés sur un seul octet

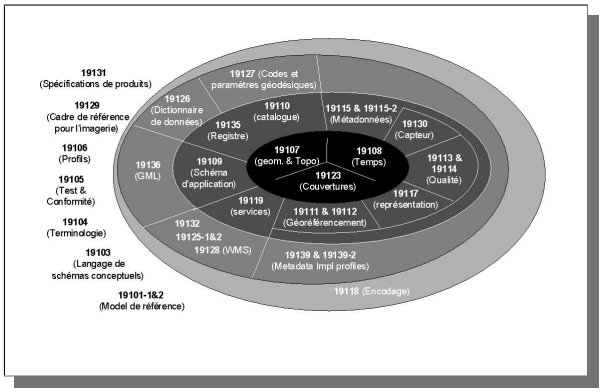

Figure A.4 – Normes associées de l'ISO 19115 (source [ADAE, 2006])

ISO 8879 : SGML
ISO/IEC 10646-1 : Technologies de l'information - Jeu universel de caractères codés sur plusieurs octets (JUC)
ISO/IEC 11179 : Technologies de l'information - Spécification et standardisation des éléments de données
ISO 19106 : Information géographique - Profils
ISO 19107 : Information géographique - Schéma spatial
ISO 19108 : Information géographique - Schéma temporel
ISO 19109 : Information géographique - Règles de schémas d'application
ISO 19110 : Information géographique - Méthodologie de catalogage des objets
ISO 19111 : Information géographique - Référencement spatial par les coordonnées
ISO 19112 : Information géographique - Référencement spatial par les identificateurs géographiques
ISO 19113 : Information géographique - Principes de qualité
ISO 19114 : Information géographique - Procédures d'évaluation de la qualité
ISO 19117 : Information géographique - Présentation
ISO 19118 : Information géographique - Encodage
ISO 19139 : Information géographique - Implémentation XML

Annexe B

Le profil MUMSDI

B.1 Métadonnées ISO19115 maintenues dans le profil MUMSDI

Par défaut, nous considérons que les sections, entités et éléments de métadonnées sont obligatoires dans le profil MUMSDI, excepté si une indication contraire est spécifiée.

Les **sections de métadonnées** qui ont été conservées dans le profil sont :
- Le package contenant les informations sur les métadonnées qui est obligatoire dans la norme.
- Le package contenant les informations sur l'identification des données qui est obligatoire dans la norme.
- Le package contenant les informations de références et sur les parties responsables qui sont obligatoires dans le noyau.
- Le package contenant les informations de qualité que nous rendons obligatoire dans le profil.
- Le package contenant les informations sur l'étendue des données que nous rendons obligatoire dans le profil.

Les **entités de métadonnées** qui ont été conservées dans le profil sont :
- La classe `MD_Metadata`, obligatoire dans la norme
- La classe `MD_Identification` et sa classe dérivée `MD_DataIdentification`, obligatoires dans la norme.
- La classe `DQ_DataQuality`, sa classe agrégée `DQ_Element` et ses classes dérivées `DQ_Completeness`, `DQ_PositionalAccuracy`, `DQ_ThematicAccuracy`, que nous rendons obligatoires dans le profil.
- La classe `EX_Extent`, sa classe agrégée `EX_GeographicExtent` spécialisée en `EX_BoundingPolygon` et en `EX_GeographicBoundingBox` qui est conditionnelle dans le noyau et que nous rendons obligatoire dans le profil.
- La classe `CI_Citation` qui est obligatoire dans le noyau.
- La classe `CI_ResponsibleParty` qui est obligatoire dans le noyau

Les **éléments de métadonnées** que nous conservons sont :

- Pour la classe `MD_Metadata` :
  - `hierarchyLevel` qui indique le domaine d'application des métadonnées grâce à une valeur issue de la liste `MD_ScopeCode`
  - `contact` qui renseigne sur l'organisme pouvant donner de l'information sur les métadonnées. Cet élément est de type `CI_ResponsibleParty`.
  - `dateStamp` qui donne la date de création des métadonnées. Cet élément est de type `Date`

- Pour la classe `MD_Identification` :
  - `citation` qui fournit des informations sur les ressources. Cet élément est de type `CI_Citation`
  - `abstract` qui décrit brièvement le contenu des ressources.
  - `purpose` qui résume les intentions pour lesquelles les ressources ont été développées
  - `credit` qui détermine le crédit accordé aux personnes ayant contribuées à l'élaboration des sources. Cet élément est optionnel.
  - `pointOfContact` qui identifie les moyens de communication avec la ou les personnes ou organisations associées aux ressources. Cet élément de type `CI_ResponsibleParty` est optionnel.

- Pour la classe `MD_DataIdentification` :
  - `language` qui détermine la langue utilisée dans la ressource.
  - `topicCategory` qui recense les principales couches thématiques présentes dans la ressource grâce aux valeurs définies dans `MD_TopicCategoryCode`
  - `extent` qui donne les informations sur l'étendue spatiale de la ressource. Cet élément est de type `EX_Extent` et est obligatoire si `hierarchyLevel = evolutionSet ou tranfertAggregate`
  - `supplementalInformation` qui fournit tout autre information permettant de décrire plus précisément la ressource. Cet élément est optionnel.

- Pour la classe `DQ_DataQuality` :
  - `Scope` qui indique le domaine d'application des informations de qualité. Cet élément est de type `DQ_Scope`

- Pour la classe `LI_Source` :
  - `description` qui fournit une description détaillée des données sources. Cet élément est optionnel.
  - `scaleDenomimnator` qui donne le dénominateur de l'échelle des données sources. Cet élément de type `MD_RepresentativeFraction` est optionnel.
  - `sourceCitation` qui spécifie les informations de référence des données sources. Cet élément est de type `CI_Citation` et nous le rendons obligatoire.

- Pour la classe `LI_ProcessStep` :
  - `description` qui décrit les étapes du processus ayant conduit à la donnée

spécifiée par l'attribut `scope`.
  - `rational` qui indique les buts à atteindre pour une étape particulière du processus. Cet élément est optionnel.
  - `processor` qui identifie les personnes ou organisations associées avec l'étape du processus. Cet élément de type `CI_ResponsibleParty` est optionnel.

- Pour la classe `DQ_Element` :
  - `nameOfMeasure` qui indique le nom de la mesure qualité. Cet élément est optionnel.
  - `result` qui fournit les résultats obtenus pour la mesure qualité considérée. Cet élément de type `DQ_Result` est obligatoire dans le profil.

- Pour la classe `DQ_ConformanceResult` :
  - `explanation` qui décrit le type de conformité attendu. Cet élément est optionnel
  - `pass` qui indique le résultat de la conformité.

- Pour la classe `DQ_QuantitativeResult` :
  - `valueType` qui fournit le domaine de valeur possible des résultats. Cet élément est optionnel.
  - `valueUnit` donne l'unité de valeur du résultat.
  - `value` indique la liste des valeurs obtenues.

- Pour la classe `EX_BoundingPolygon` :
  - `polygone` qui fournit le polygone englobant déterminant l'étendue de l'ensemble d'évolutions. Cet élément est de type `GM_Object`.

- Pour la classe `EX_GeographicBoundingBox` :
  - `westBoundLongitude` qui donne la coordonnée la plus à l'ouest de l'étendue de l'ensemble d'évolutions.
  - `eastBoundLongitude` qui donne la coordonnée la plus à l'est de l'étendue de l'ensemble d'évolutions.
  - `southBoundLatitude` qui donne la coordonnée la plus au sud de l'étendue de l'ensemble d'évolutions.
  - `northBoundLatitude` qui donne la coordonnée la plus au nord de l'étendue de l'ensemble d'évolutions.

- Pour la classe `CI_Citation` :
  - `title` qui indique le nom avec lequel les ressources sont habituellement connues.
  - `date` qui fournit l'ensemble des dates de référence de la ressource. Cet élément est de type `CI_Date` et au moins une date doit être renseignée.
  - `citedResponsibleParty` qui donne l'identité de la personne responsable de la ressource. Cet élément de type `CI_ResponsibleParty` est optionnel.

- Pour la classe `CI_ResponsibleParty` :
  - `individualName` qui donne le nom de la personne associée aux

évolutions. Cet élément est conditionnel et doit être renseigné si l'élément `organisationName` n'a pas été rempli.
- `organisationName` qui donne le nom de l'organisation associée aux évolutions. Cet élément est conditionnel et doit être renseigné si l'élément `individualName` n'a pas été rempli.
- `role` qui renseigne sur le rôle de la personne ou de l'organisme associé aux évolutions. Cet élément prend sa valeur dans la liste `CI_RoleCode` et est obligatoire.

B.2 Dictionnaire de données du profil MUMSDI

Le catalogue des données du profil MUMSDI est fourni sous la forme d'un tableau composé de deux parties.

La première partie donne l'ensemble des entités et éléments de métadonnées utilisés dans le profil MUMSDI et précise pour chaque donnée :
- Le nom,
- La traduction française du nom,
- Le type de données,
- Le domaine d'application,
- L'obligation imposée dans le profil : M pour obligatoire, C pour conditionnel et O pour optionnel,
- La cardinalité,
- La présence ou l'absence de l'élément ou de l'entité dans le noyau ISO 19115
- Si l'élément ou l'entité est une extension dans le profil MUMSDI

La seconde partie du tableau fournit la liste des codes utilisés dans le profil MUMSDI et précise pour chaque catégorie et chaque code :
- Le nom,
- Le domaine d'application, en particulier pour les catégories, nous indiquons s'il s'agit d'une liste de code (qui est par définition extensible) ou d'une énumération (qui est par définition une liste non modifiable).
- La définition
- La présence ou l'absence de la liste et des codes dans la norme ISO 19115
- Si la catégorie ou le code est une extension dans le profil MUMSDI

Catalogue des classes et objets du profil MUMSDI

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
MD_METADATA	Classe racine qui définit les métadonnées sur la ou les ressources	Classe		M	1	☑	☐
hierarchyLevel	Domaine d'application des métadonnées	Classe : CodeList	MD_ScopeCode	M	1..*	☐	☐
contact	Organisme pouvant fournir l'information sur les métadonnées	Classe : DataType	**CI_ResponsibleParty**	M	1..*	☑	☐
dateStamp	Date de création des métadonnées	Classe définie dans ISO 8601	**Date**	M	1	☑	☐
Role name : identificationInfo	Informations de base sur la ou les ressources pour lesquelles les métadonnées sont décrites	Association	**MD_Identification**			☑	☐
Role name : dataQualityInfo	Qualité relative aux ressources	Association	**DQ_DataQuality**			☑	☐
MD_IDENTIFICATION	Informations nécessaires à l'identification de la ou des ressources	Classe abstraite agrégée à MD_Metadata		M	1..*	☑	☐
citation	Informations sur les ressources	Classe : DataType	**CI_Citation**	M	1	☑	☐
abstract	Bref résumé du contenu des ressources	CharacterString	Texte libre	M	1	☑	☐
purpose	Résumé des intentions pour lesquelles les ressources ont été développées	CharacterString	Texte libre	M	1	☐	☐
credit	Crédit accordé aux personnes qui ont contribués à l'élaboration de ces ressources	CharacterString	Texte libre	O	0..*	☐	☐
pointOfContact	Identification et moyen de communication avec la ou des personnes et la ou les organisations associées aux ressources	Classe : DataType	**CI_ResponsibleParty**	O	0..*	☑	☐
MD_DATA_IDENTIFICATION	Informations nécessaires pour l'identification d'un jeu de données	Classe dérivée de MD_Identification		M	1..*	☑	☐
language	Langue utilisée dans le jeu de données	CharacterString	Défini dans ISO 639-3	M	1	☑	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
topicCategory	Principaux thèmes présents dans le jeu de données	Classe : Enumération	MD_TopicCategoryCode	M	1..*	☑	☐
extent	Informations sur l'étendue spatiale du jeu de données	Classe : DataType	**EX_Extent**	C si level =evolution Set ou level = tranferAggregate	0..1	☑	☐
supplementalInformation	Toute autre information permettant de décrire le jeu de données	CharacterString	Texte libre	O	0..1	☐	☐
MD_REPRESENTATIVEFRACTION	Echelle utilisée	Classe dérivée de ISO 19103		M	1	☑	☐
denominator	Dénominateur de l'échelle	Entier	Entier strictement positif	M	1	☑	☐
EX_EXTENT	Informations sur l'étendue horizontale, verticale et temporelle	Classe : DataType		M	1	☑	☐
Role name : geographicElement	Fournit les composants géographiques de l'étendue pour l'objet référencé	Association	**EX_GeographicExtent**			☑	☐
EX_GEOGRAPHICEXTENT	Zone géographique constituant le jeu de données	Classe abstraite agrégée à EX_Extent		M	1	☑	☐
EX_BOUNDINGPOLYGON	Rectangle englobant le jeu de données, exprimé par l'ensemble fermé des coordonnées du polygone	Classe dérivée de EX_GeographicExtent		C si EX_GEOGRAPHICBOUNDINGBOX n'est pas renseigné	0..1	☐	☐
polygon	Ensemble de points définissant le polygone	Classe	**GM_Object**	M	1	☑	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
EX_GEOGRAPHICBOUNDINGBOX	Position géographique du jeu de données. Valeur approximative, il n'est pas nécessaire de spécifier le système de coordonnées de référence	Classe dérivée de EX_GeographicExtent		C si EX_BOUNDINGPOLYGON n'est pas renseigné	0..1	☑	☐
westBoundLongitude	Coordonnée la plus à l'ouest de l'étendue du jeu de données. Longitude exprimée en degré décimal, valeurs positives à l'est	Décimal	-180.0 à + 180.0. La valeur de westBoundLongitude est inférieure à celle de eastBoundLongitude	M	1	☑	☐
eastBoundLongitude	Coordonnée la plus à l'est de l'étendue du jeu de données. Longitude exprimée en degré décimal, valeurs positives à l'est	Décimal	-180.0 à + 180.0 La valeur de eastBoundLongitude est supérieure à celle de westBoundLongitude	M	1	☑	☐
southBoundLatitude	Coordonnée la plus au sud de l'étendue du jeu de données. Latitude exprimée en degré décimal, valeurs positives au nord	Décimal	- 90.0 à 90.0 La valeur de southBoundLatitude est inférieure à la valeur de northBoundLatitude	M	1	☑	☐
northBoundLatitude	Coordonnée la plus au nord de l'étendue du jeu de données. Latitude exprimée en degré décimal, valeurs positives au nord	Décimal	- 90.0 à 90.0 La valeur de northBoundLatitude est supérieure à la valeur de southBoundLatitude	M	1	☑	☐
CI_RESPONSIBLEPARTY	Identification et moyens de communication avec la ou les personnes ou organisations associées au jeu de données et aux métadonnées	Classe : DataType		M si objet référencé	0..*	☑	☐
individualName	Nom de la personne responsable	CharacterString	Texte Libre.	C si organisationName non renseigné	0..1	☑	☐
organisationName	Nom de l'organisation responsable	CharacterString	Texte Libre	C si individualName non renseigné	0..1	☑	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
role	Role de l'organisme ou de la personne responsable	Classe : CodeList	CI_RoleCode	M	1	☑	☐
CI_CITATION	Description standardisée des informations de référence concernant la ressource	Classe : DataType		M	1..*	☑	☐
title	Nom avec lequel les ressources sont habituellement connues	CharacterString	Texte Libre	M	1	☑	☐
date	Date de référence de la ressource en question	Classe : DataType	**CI_Date**	M	1..*	☑	☐
citedResponsibleParty	référence de la personne responsable de la ressource en question	Classe	**CI_ResponsibleParty**	0	0..*	☐	☐
CI_DATE	Date de référence et événement qui est lié à cette date	Classe : DataType		M	1..*	☑	☐
Date	Date de référence de la ressource	Classe définie dans ISO 8601	Date	M	1	☑	☐
dateType	Evénement associé à la date de référence	Classe : CodeList	CI_DateTypeCode	M	1	☑	☐
DQ_DATAQUALITY	Informations sur la qualité des données spécifiées par l'attribut scope	Classe agrégée à MD_Metadata		M	1..*	☑	☐
Scope	Données cibles des informations de qualité	Classe : DataType	**DQ_Scope**	M	1	☐	☐
Role name : report	Informations quantitatives sur la qualité des données spécifiées par l'attribut scope	Association	**DQ_Element**			☐	☐
Role name : lineage	Informations non quantitatives sur la qualité des données spécifiées par l'attribut scope	Association	**LI_Lineage**			☐	☐
DQ_SCOPE	Caractéristiques des données pour lesquelles les informations de qualité sont fournies	Classe : DataType		M	1	☐	☐
level	Niveau hiérarchique des données	Classe : CodeList	MD_ScopeCode	M	1	☐	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
levelDescription	Description détaillée du niveau hiérarchique des données	Classe : Union	**MD_ScopeDescription** :	C si level n'est pas egal à evolutionSet ou transferAggregate	0..*	☐	☐
MD_SCOPEDESCRIPTION	Description détaillée de la classe considérée dans le niveau hiérarchique	Classe : Union		C	0..*	☐	☐
attributes	Attributs sur lesquels l'information s'applique	Ensemble	**GF_AttributeType** de ISO 19109	C si aucun des autres n'est renseigné	0..1	☐	☐
features	Objets sur lesquels l'information s'applique	Ensemble	**GF_FeatureType** de ISO 19109	C si aucun des autres n'est renseigné	0..1	☐	☐
elementaryUpdates	Type d'évolution élémentaire sur laquelle l'information s'applique	Classe : CodeList	**MU_EvolTypeCode**	C si aucun des autres n'est renseigné	0..1	☐	☐
DQ_ELEMENT	Informations quantitatives de la qualité	Classe abstraite agrégée à DQ_DataQuality		M	1..*	☐	☐
nameOfMeasure	Nom de la mesure Ex : Synthetic expression of the horizontal quality or Synthetic expression of the reliability	CharacterString	Texte Libre	O	0..*	☐	☐
result	Résultats obtenus pour la mesure qualité considérée	Classe	**DQ_Result**	M	1..3	☐	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
DQ_COMPLETENESS	Exhaustivité des objets, attributs et des relations	Classe abstraite dérivée de DQ_Element		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_COMPLETENESSOMISSION	Absence de données dans le jeu défini par l'attribut scope	Classe dérivée de la classe DQ_Completeness		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_POSITIONALACCURACY	Précision de la position des objets	Classe abstraite dérivée de DQ_Element		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_ABSOLUTEEXTERNALPOSITIONALACCURACY	Précision de position absolue des objets	Classe dérivée de DQ_PositionalAccuracy		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_RELATIVEINTERNALPOSITIONALACCURACY	Précision de position relative entre les objets	Classe dérivée de DQ_PositionalAccuracy		C si aucune autre mesure n'est effectuée	0..1	☐	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
DQ_THEMATICACCURACY	Précision des attributs quantitatifs et exactitude des attributs qualitatifs	Classe abstraite dérivée de DQ_Element		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_NONQUANTITATIVEATTRIBUTEACCURACY	Exactitude des attributs qualitatifs	Classe dérivée de DQ_ThematicAccuracy		C si aucune autre mesure n'est effectuée	0..1	☐	☐
DQ_QUANTITATIVEATTRIBUTACCURACY	Précision des attributs quantitatifs	Classe dérivée de DQ_ThematicAccuracy		C si aucune autre mesure n'est effectuée	0..1	☐	☐
MU_USABILITY	Information sur la fiabilité accordée aux mises à jour	Classe dérivée de DQ_Element		C si aucune autre mesure n'est effectuée	0..1	☐	☑
finalUserCode	Rôles des acteurs pour lesquels les données sont potentiellement utiles	Classe : CodeList	**CI_RoleCode**	M	1	☐	☑
finalUserLocation	Localisation des acteurs pour lesquels les données sont potentiellement utiles	CharacterString	Texte libre	M	1	☐	☑
DQ_RESULT	Résultats détaillés des mesures de qualité	Classe abstraite		M	1..3	☐	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
DQ_CONFORMANCERESULT	Evaluation de la conformité du résultat avec une valeur de qualité connue et considérée comme acceptable	Classe dérivée de DQ_Result				☐	☐
explanation	Signification de la conformité pour ce résultat	CharacterString	Texte Libre	O	0..1	☐	☐
pass	Indication de la conformité du résultat. 0 = non-conforme, 1 = conforme	Boolean	1 = oui 0 = non	M	1	☐	☐
DQ_QUANTITATIVERESULT	Valeur (ou ensemble de valeurs) ou informations sur la valeur résultante de la mesure qualité appliquée	Classe dérivée de DQ_Result		M	1. Combien de resultat possible ?	☐	☐
valueType	Domaine de valeur du résultat	Classe	Type	0	0..1	☐	☐
valueUnit	Unité de valeur du résultat	Classe	UnitOfMeasure défini dans ISO 19103	M	1	☐	☐
value	Valeurs quantitatives du résultat	Classe	Record	M	1..*	☐	☐
MU_QUALITATIVERESULT	Informations qualitatives sur la qualité des évolutions	Classe dérivée de DQ_Result		O	0..*	☐	☑
documentation	Qualité de documentation des attributs	Classe : CodeList	MU_DocumentationCode	0	0..1	☐	☑
ErrorType	Types d'erreurs pouvant être rencontrés dans l'ensemble d'évolutions	Classe	**MU_Error**	O	0..3	☐	☑
MU_ERROR	Informations sur le type d'erreur qui a pu être effectué lors de la saisie des évolutions	Classe : DataType		M	1..3	☐	☑
geometricError	Liste des erreurs de type géométrique	CharacterString	Texte libre	C si aucun autre résultat n'est donné	0..*	☐	☑

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
attributeError	Liste des erreurs portant sur les attributs	CharacterString	**MU_Error**	C si aucun autre résultat n'est donné	0..*	☐	☑
topologicError	Liste des erreurs de type topologique	CharacterString	**MU_Error**	C si aucun autre résultat n'est donné	0..*	☐	☑
LI_LINEAGE	Informations sur les étapes du processus de production des données ou sur les données sources utilisées pour construire le jeu de données	Classe agrégée à DQ_DataQuality	Est-ce utile pour les évolutions ?	M	1	☑	☐
Role name : source	Informations sur les données sources utilisées pour construire le jeu de données	Association	**LI_Source**			☐	☐
Role name : processStep	Informations sur les processus utilisés pour construire le jeu de données	Association	**LI_ProcessStep**			☐	☐
LI_SOURCE	Informations sur les données sources utilisées pour construire le jeu de données	Classe agrégée à LI_Lineage		C si LI_ProcessStep non renseigné	0.. *	☐	☐
description	Description détaillée des données sources	CharacterString	Texte Libre	C si sourceExtent n'est pas fourni	0..1	☐	☐
scaleDenominator	Dénominateur de l'échelle des données sources	Classe : DataType	**MD_RepresentativeFraction**	O	0..1	☐	☐
sourceCitation	Informations de référence sur les données sources	Classe : DataType	**CI_Citation**	O	0..1	☐	☐
LI_PROCESSSTEP	Informations sur les processus sources utilisés pour construire le jeu de données	Classe agrégée à LI_Lineage		C si LI_Source non renseignée	0.. *	☐	☐

Nom anglais	*Traduction définition ISO*	*Type de Données*	*Domaine*	*Obligation MUMSDI*	*Card*	*Noyau ISO*	*Extension MUMSDI*
description	Description détaillée des étapes du processus	CharacterString	Texte Libre	M	1	☐	☐
rationale	Objectifs du processus	CharacterString	Texte Libre	O	0..1	☐	☐
processor	Informations de référence sur les personnes ayant conduit le processus	Classe : DataType	**CI_ResponsibleParty**	O	0..1	☐	☐

Listes de codes utilisés dans le profil MUMSDI

Nom	Domaine	Définition	ISO 19115	Extension MUMSDI
MD_SCOPE_CODE	« CodeList »	Types de données sur lesquels les métadonnées s'appliquent	☑	☐
attribute	001	Les informations s'appliquent sur une valeur d'attribut	☑	☐
feature	009	Les informations s'appliquent sur une instance d'objet géographique	☑	☐
evolutionSet	017	Les informations s'appliquent sur un ensemble d'évolutions	☐	☑
elementaryUpdate	018	Les informations s'appliquent sur une mise à jour élémentaire	☐	☑
transferAggregate	019	Les informations s'appliquent sur un produit	☐	☑

MD_TOPICCATEGORYCODE	« Enumeration »	Classification des principaux thèmes de données géographiques	☑	☐
boundaries	003	Frontières. Ex : Frontières politiques et administratives …	☑	☐
imageryBaseMapsEarthCover	010	Cartes de base. Ex : cartes topographiques, images ...	☑	☐
intelligenceMilitary	011	Bases et structures militaires. Ex : routes militaires, casernes …	☑	☐
inlandWaters	012	Hydrographie hors mer et océans. Ex : rivières, réservoirs, qualité de l'eau …	☑	☐
location	013	Informations sur la localisation et les services. Ex : adresse, nom de places, code postaux …	☑	☐
structure	017	Constructions édifiées par l'homme. Ex : immeubles, musées, églises, écoles, tours …	☑	☐
transportation	018	Moyens de communication. Ex : Route, voie ferrée …	☑	☐
CI_ROLECODE	« CodeList »	Rôle des individus ou organismes	☑	☐
resourceProvider	001	Fournisseur	☑	☐
owner	003	Propriétaire	☑	☐
user	004	Utilisateur	☑	☐
originator	006	Créateur	☑	☐
pointOfContact	007	Personne ou organisme qui possède les renseignements sur la manière d'acquérir les ressources	☑	☐
producer	012	Producteur dans la mission	☐	☑
complexOperational	013	Opérationnel complexe	☐	☑
simpleOperational	014	Opérationnel simple	☐	☑
allied	015	Allié	☐	☑

CI_DATETYPECODE	« CodeList »	Evénement associé à une date	☑	☐
création	001	Création des données	☑	☐
publication	002	Emission des données	☑	☐
revision	003	Révision des données	☑	☐
Update	004	Mise à jour des données	☐	☑
MU_DOCUMENTATIONTYPECODE	« CodeList »	Estimation qualitative de la documentation des attributs lors de la mise à jour	☐	☑
noDocumented	001	Les attributs ne sont pas renseignés correctement (0% de valeurs correctes)	☐	☑
badDocumented	002	Les attributs sont plutôt mal renseignés (25% de valeurs correctes)	☐	☑
halfDocumented	003	Les attributs sont à moitié renseignés correctement (50% de valeurs correctes)	☐	☑
goodDocumented	004	Les attributs sont plutôt bien renseignés (75% de valeurs correctes)	☐	☑
allDocumented	005	Les attributs ont tous été renseignés correctement (100% de valeurs correctes)	☐	☑
MU_EVOLTYPECODE	« CodeList »	Type des évolutions élémentaires	☐	☑
creation	001	Evolution de type création	☐	☑
suppression	002	Evolution de type suppression	☐	☑
modificationGeometrique	003	Evolution de type modification géométrique	☐	☑
modificationAttributaire	004	Evolution de type modification attributaire	☐	☑
modificationMixte	005	Evolution de type modification mixte	☐	☑

Annexe C

Le formalisme ECA : Evénements-Conditions-Actions

Dans ce formalisme, un groupe d'actions est défini par **une règle** qui doit être exécutée lorsqu'un certain prédicat est vérifié.
Une **action** est composée d'un groupe d'opérations. Par exemple, une mise à jour, une annulation, un déclenchement, ...
Une **condition** est un énoncé qui doit être vérifié. Par exemple une interrogation du SGBD, une fonction booléenne, ...
Un **événement** est spécifié par une opération (le type de l'événement) et par la liste des objets sur lesquels il s'applique. Plusieurs types d'événements sont définis :

- ✓ Les **événements primitifs** qui peuvent être :
  - des événements liés à l'exécution d'opérations de mise à jour.
  - des événements temporels.
  - des événements externes.
- ✓ Les **événements combinés** qui peuvent être construits à partir :
  - d'une conjonction d'événements primitifs.
  - d'une disjonction de différents événements qui déclenchent une même règle.
  - d'une disjonction de plusieurs événements qui expriment sémantiquement une même règle.
  - d'une fermeture c'est à dire que l'événement doit avoir eu lieu au moins une fois pour que la règle soit appliquée.

Trois types de règles peuvent être utilisées dans ce formalisme. Elles dépendent de la définition des événements, conditions et actions :

- ✓ Règle 1 : **Les événements sont dérivés implicitement des conditions**
  Syntaxe :
  ```
  Si <conditions>
  Alors <actions>
  [Attributs {priorité, mode de connexion, ..}]
  ```

- ✓ Règle 2 : **L'action est vue comme une transaction**. La condition est alors incluse dans la transaction
  Syntaxe :

```
Quand <événements>
Alors <actions>
[Attributs {priorité, mode de connexion, ..}]
```

✓ Règle 3 : **Les événements, conditions et actions sont définis séparément**
Syntaxe :

```
Quand <événements>
Si <conditions>
Alors <actions>
[Attributs {priorité, mode de connexion, ..}]
```

Le **modèle d'exécution d'une règle** est quant à lui spécifié par les trois critères suivants :

- Le **mode de connexion** qui définit le moment où la condition est évaluée et où l'action est exécutée par rapport à l'occurrence de l'événement. Il se distingue selon trois formes :
  - Immédiat, c'est à dire juste après l'événement ou la condition
  - Différé, c'est à dire exécuté dans la même transaction mais juste avant la validation
  - Séparé, c'est à dire exécuté dans une transaction indépendante

- Le **mode d'exécution transactionnel** qui définit le comportement transactionnel de l'exécution d'une règle par rapport à l'événement qui a déclenché cette exécution. Il peut prendre les trois formes suivantes :
  - La règle est exécutée dans la transaction de déclenchement
  - La règle est exécutée dans une transaction imbriquée à la transaction de déclenchement
  - La règle est exécutée dans une transaction indépendante de la transaction de déclenchement

- La **séquence d'exécution** des règles qui est définie par l'enchaînement qui a lieu entre les différentes règles :
  - Il peut être séquentiel ou parallèle lorsque plusieurs règles possèdent le même événement
  - Il peut être en cascade lorsque l'exécution d'une règle en active une autre. Dans ce cas, soit l'exécution de la règle première s'interrompt en faveur de l'exécution des règles qu'elle active, soit les règles activées s'exécutent après la fin de l'exécution de la règle première

Bibliographie

[ADAE, 2006] ADAE (2006). *Information Géographique. Recommandation relative aux métadonnées*. Agence pour le developpement de l'administration electronique. République Française, Ministere du budget et de la réforme de l'état. Projet 8 DT. TN/05.002,Version 1.0.

[Agumya et Hunter, 1998] Agumya, A. et Hunter, G. (1998). Fitness for use : Reducing the impact of geographic information uncertainty. *In URISA 98 Proceedings*.

[Allen *et al.*, 1995] Allen, L., Fernandez, G., Kane, K., Leblang, D., Minard, D. et Posner, J. (1995). Clearcase multisite : Supporting geographically-distributed software development. *In Software Configuration Management : selected papers of the ICSE SCM-4 and SCM-5 Workshops*.

[ANZLIC, 1998] ANZLIC (1998). *National Spatial Data Infrastructure for Australia and New Zealand*. The Australian and New Zealand Land Information Council. http ://www.anzlic.org.au.

[Aquino et Kim, 2003] Aquino, J. et Kim, D. (2003). Jump, the unified mapping platform. Developper's guide, Vivid Solutions. www.vivissolutions.com.

[Arcangeli *et al.*, 2004] Arcangeli, J., Hameurlain, A., Migeon, F. et Morvan, F. (2004). Mobile agent based self adaptative join for wide area distributed query processing. *Journal of Database Management*, 15(4):25–44.

[Badard, 1998] Badard, T. (1998). Extraction des mises à jour dans les BDG : de l'utilisation des méthodes d'appariement. *Revue Internationale de Géomatique*, 8(1-2):121–147.

[Badard, 2000] Badard, T. (2000). *Propagation des mises à jour dans les bases de données géographiques multi-représentations par analyse des changements géographiques*. Thèse de doctorat, Université de Marne la Vallée.

[Badard et Lemarié, 1999] Badard, T. et Lemarié, C. (1999). *Propagating updates between geographic databases with different scales*, chapitre 10. Atkinson and Martin editions, Taylor and Francis publication. Innovations in GIS VII : GeoComputation.

[Badard et Richard, 2001] Badard, T. et Richard, D. (2001). Using XML for the exchange of updating information between geographical information systems. *Computers Environment and Urban Systems (CEUS)*, 25:17–31.

[Barde, 2005] Barde, J. (2005). *Mutualisation de données et de connaissances pour la Gestion Intégrée des Zones Côtières. Application au projet SYSCOLAG*. Thèse de doctorat, Université Montpellier II.

[Bedard *et al.*, 1997] Bedard, Y., van Chestein, Y. et Poupart-Lavoie, G. (1997). Actualisation des données à référence spatiale. Rapport technique, Centre de recherche en géomatique, Université Laval, Québec. volet échange et intégration.

[Bel-Hadj-Ali, 2001] Bel-Hadj-Ali, A. (2001). *Qualité géométrique des entités géographiques surfaciques : Application à l'appariement et définition d'une typologie des écarts géométriques*. Thèse de doctorat, Université de Marne la Vallée.

[Benslimane *et al.*, 1999] Benslimane, D., Jouanot, F., Laurini, R., Yetongnon, K., Cullot, N. et Savonnet, M. (1999). Interopérabilité des sig : un état de l'art. *Revue Internationale de géomatique*, 9(3):279–316.

[Berliner, 1990] Berliner, B. (1990). Cvs ii : Parallelizing software development. *In USENIX Winter 1990 Technical Conference*.

[Bernard *et al.*, 2003] Bernard, G., Ben-Othman, J., Bouganim, L., Canals, G., Defude, B., Ferrie, J., Gançarski, S., Guerraoui, R., Molli, P., Pucheral, P., Roncancio, C., Serrano-Alvarado, P. et Valduriez, P. (2003). Mobilité et bases de données : état de l'art et perspectives. *Technique et science informatiques*, 22(3 et 4).

[Bernstein et Andgoodman, 1983] Bernstein, B. et Andgoodman, N. (1983). The failure and recovery problem for replicated databases. *In 2nd Symposium on Principles of Distributed Computing (PODC)*.

[Bernstein *et al.*, 1987] Bernstein, P., Hadzilacos, V. et Goodman, N. (1987). *Concurrency Control and Recovery in Databases*. Addison-Wesley, Reading, Massachusetts.

[Bernstein et Newcomer, 1997] Bernstein, P. et Newcomer, E. (1997). *Principles of Transaction Processing*. Morgan Kaufmann.

[Bertolotto *et al.*, 1994] Bertolotto, M., Floriani, L. D. et Puppo, E. (1994). Multiresolution topological maps. *In Advanced Geographic Data Modelling (AGDM)*.

[Bicking, 1994] Bicking, B. (1994). *A Formal Approach to Automate Thematic Accuracy Checking for Cartographic Data Sets*. Thèse de doctorat, Université du Maine, USA,.

[Bishr, 1997] Bishr, Y. (1997). *Semantics Aspects of Interoperable GIS*. Thèse de doctorat, International Institute of Aerospace Survey and Earth Sciences, Enschede, the Netherlands.

[Bouziani, 2003] Bouziani, M. (2-5 dec 2003). Définition d'une méthode d'extraction des mises à jour de l'information spatiale dans un réseau routier en milieu urbain. *In 2nd FIG Regional Conference*.

[Box, 1976] Box, G. (1976). Science and statistics. *Journal of the American Statistical Association*, 71:791–799.

[Branki et Defude, 1997] Branki, T. et Defude, B. (1997). A terminological Canonical Data Model for Cooperating Heterogeneous Geographical Information Systems. *In DEXA conference*.

[Brassel *et al.*, 1995] Brassel, K., Bucher, F., Stephan, E. et Vckovski, A. (1995). *Elements of Spatial Data Quality*, chapitre Completeness. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Braun, 2003] Braun, A. (2003). Conception d'outils d'aide à l'intégration des mises à jour dans les bases de données géographiques utilisateur. Rapport technique, IGN.

[Breitbart et Korth, 1997] Breitbart, Y. et Korth, H. F. (1997). eplication and consistency : Being lazy helps sometimes. *In ACM PODS International Conférence*.

[Brodeur, 2004] Brodeur, J. (2004). Interoperability of Geographic Information : A Communication-Based Prototype. *In 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI)*, pages 327–332.

[Brodeur *et al.*, 2003] Brodeur, J., Bedard, Y., Edwards, G. et Moulin, B. (2003). Revisiting the concept of geospatial data interoperability within the scope of human communication process. *Transactions in GIS*, 7(2):243–265.

[Bruin *et al.*, 2001] Bruin, S. D., Bregt, A. et de Ven, M. V. (2001). Assessing fitness for use : the expected value of spatial data sets. *International Journal of Geographical Information Science*, 15(5):457–471.

[Bucher, 2002] Bucher, B. (2002). *L'aide à l'accès à l'information géographique : un environnement de conception coopérative d'utilisations de données géographiques*. Thèse de doctorat, Université Paris 6.

[CARGENE, 2004] CARGENE (2004). *Spécifications des produits DNG3D - Caractéristiques générales (CARGENE 1.0)*. République Française,Ministère de la défense, IGN/DT.TN/03.055.

[Cederqvist *et al.*, 2001] Cederqvist, P., Pesch, R. et al. (2001). Version management with cvs. Rapport technique, Ximbiot LLC.

[Cellary et Jomier, 1990] Cellary, W. et Jomier, G. (1990). Consistency of versions in object-oriented databases. *In 16th VLDB conference*.

[CEN, 1998] CEN (1998). *Geographic Information European Prestandards, Euronorme Voluntaire for Geographic Information –Data description- Metadata*. European Committee for Standardization – CEN/TC287.

[Chan et Williamson, 1999] Chan, T. et Williamson, I. (1999). The different identities of gis and gis diffusion. *International journal of Geographic Information Science*, 13(3):267–281.

[Cheylan *et al.*, 1994] Cheylan, J., Lardon, S., Mathian, H. et Sanders, L. (1994). Les problématiques liées au temps dans les SIG. *Revue internationale de géomatique*, 4(3-4):287–305.

[Chrisman, 1983] Chrisman, N. (1983). The role of quality in the long-term functioning of geographic information system. *In AUTO-CARTO 6*.

[Chrisman, 1989] Chrisman, N. (1989). *Accuracy of Spatial Databases*, chapitre Modelling Error in Overlaid Categorial Maps. Taylor et Francis.

[Christensen, 2001] Christensen, A. (2001). *Issues in the Conceptual Modeling of Geographic Data*. Thèse de doctorat, The Danish Research Agency and The National Survey and Cadastre.

[Claramunt *et al.*, 1994] Claramunt, C., Séde, M., Prelaz-Droux, R. et Vidale, L. (1994). Sémantique et logique spation-temporelles. *Revue internation de géomatique*, 4(2):165–180.

[Clarke et Clark, 1995] Clarke, D. et Clark, D. (1995). *Elements of Spatial Data Quality*, chapitre Lineage. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Coleman et Nebert, 1999] Coleman, D. et Nebert, D. (1999). Building a north american spatial data infrastructure. *Cartography and Geographic Information Systems*, 25(3):151–160.

[Craglia et Johnston, 2004] Craglia, M. et Johnston, A. (2004). Assessing the impacts of spatial data infrastructures : Methods and gaps. *In 7th AGILE Conference on Geographic Information Science*.

[Danko, 2005] Danko, D. (2005). Metadata and related standards : Overview / demonstration. *In ISO Standard Workshop in the 22th International Cartographic Conference*.

[Dassonville *et al.*, 2002] Dassonville, L., Vauglin, F., Jakobsson, A. et Luzet, C. (2002). *Spatial Data Quality*, chapitre Quality Management, Data Quality and Users, Metadata for Geographical Information. Taylor et Francis.

[David, 1991] David, B. (1991). *Modélisation, représentation et gestion d'information géographique*. Thèse de doctorat, Université de Paris 6.

[David et Fasquel, 1997] David, B. et Fasquel, P. (1997). Qualité d'une base de données géographique : concepts et terminologie. Rapport technique, IGN. Bulletin d'information n°67.

[de Cambray, 1994] de Cambray, B. (1994). *Etude de la modélisation de la représentation de l'information spatiale 3D dans les bases de données géographiques*. Thèse de doctorat, Université Paris 6.

[Dedieu, 2002] Dedieu, M. (2002). *Réplication optimiste pour les applications collaboratives asynchrones*. Thèse de doctorat, Université de Marne la Vallée.

[Defude, 2005] Defude, B. (2005). Bases de données : de l'objet à l'intéropérabilité. Mémoire d'HDR.

[Demers *et al.*, 1994] Demers, A., Petersen, K., Spreitzer, M., Terry, D., Theimer, M. et Welch, B. (1994). The bayou architecture : Support for data sharing among mobile users. *In IEEEWorkshop on Mobile Computing Systems and Applications*.

[Devillers, 2004] Devillers, R. (2004). *Conception d'un système multidimensionnel d'information sur la qualité des données géospatiales*. Thèse de doctorat, Université de Laval, Québec et Université de Marnes la Vallée, France.

[Devillers et Jeansoulin, 2005] Devillers, R. et Jeansoulin, R. (2005). *Qualité de l'information géographique*. Traités en Information Géographique et Aménagement du Territoire, IGAT, Hermès Sciences, Lavoisier. ISBN 2-7462-1097-5.

[Devogele, 1997] Devogele, T. (1997). *Processus d'intégration et d'appariement des bases de données géographiques. Application à une base de données routière multi-échelles*. Thèse de doctorat, Université de Versailles.

[Dietterich, 1994] Dietterich, D. (1994). Dec data distributor : For data replication and data warehousing. *In International Conference on Management of Data (SIGMOD)*.

[DIGEST, 2000] DIGEST (2000). *DIgital Geographic information Exchange STandard : STANAG 7074*. Digital Geographic Information Working Group, members of OTAN.

[Ding *et al.*, 2004] Ding, J., Ahearn, S. et Cooper, E. (2004). An Incremental Geographic Update System (IGUS) for a large Geographic Database in New York City. *In 7th AGILE Conference on Geographic Information Science*.

[Doucet *et al.*, 1996] Doucet, A., Gançarski, S., Jomier, G. et Monties, S. (1996). Maintien de la Cohérence dans une Base de données Multiversion. *In 12èmes Journées Bases de Données Avancées*.

[Drummond, 1995] Drummond, J. (1995). *Elements of Spatial Data Quality*, chapitre Positional Accuracy. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Dupont, 1995] Dupont, Y. (1995). *Une méthode flexible pour l'intégration de schémas dans les bases de données à objets complexes*. Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne.

[Edwards *et al.*, 1997] Edwards, W., Mynatt, E., Petersen, K., Spreitzer, M., Terry, D. et Theimer, M. (1997). Designing and implementing asynchronous collaborative applications with bayou. *In ACM Symp. on User interface software and technology*.

[Egenhofer, 1999] Egenhofer, M. (1999). *Introduction : Theory and Concepts*, pages 1–4. Goodchild M. and al. editions, Kluwer Academic Publisher. Interoperating Geographic Information Systems.

[Egenhofer *et al.*, 1989] Egenhofer, M., Frank, A. et Jackson, J. (1989). A topological data model for spatial databases. *In Symposium Design an Implementation of Large Spatial Databases*, pages 271–286.

[Ellis *et al.*, 1991] Ellis, C., Gibbs, S. et Rein, G. (1991). Groupware : Some issues and experiences. *Communications of the ACM*, 34(1):38–58.

[ESRI, 1998] ESRI (1998). Esri shapefile technical description. Tehnical paper, ESRI. http ://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[ESRI, 2004] ESRI (2004). Versioning. Tehnical paper, ESRI. www.esri.com.

[Fagan et Soehngen, 1987] Fagan, G. et Soehngen, H. (1987). Improvement of gbf/dime file coordinates in a geobased information system by various transformation methods and rubbersheeting based on triangulation. *In Auto Carto 8*.

[Faïz, 1996] Faïz, T. (1996). *Modélisation, Exploitation et Visualisation de l'Information Qualité dans les Bases de Données Géographiques*. Thèse de doctorat, Université Paris sud.

[FGDC, 1997] FGDC (1997). *Framework, introduction and guide*. Federal Geographic Data Committee.

[FGDC, 1998] FGDC (1998). *Content Standard for Digital Geospatial Metadata, version 2.0. Document FGDC-SDT-001-1998*. Federal Geographic Data Committee, Metadata Ad Hoc Working Group.

[Fontaine, 2001] Fontaine, R. L. (2001). A delta format for xml : Identifying changes in xml files and representing the changes in xml. *In XML Europe, Graphic Communications Association*.

[Frank, 1998] Frank, A. (1998). *Metamodels for data quality Description*, pages 15–29. Data Quality in Geographic Information. From Error to Uncertainty. Editions Hermès.

[Frank *et al.*, 2004] Frank, A., Grum, E. et Vasseur, B. (2004). Procedure to select the best dataset for a task. *In International Conference on Geographic Information Science*.

[Gançarski, 1994] Gançarski, S. (1994). *Versions et Bases de données : modèle formel, supports de langage et d'interface utilisateur*. Thèse de doctorat, Université Paris Sud.

[Gançarski, 2006] Gançarski, S. (2006). Cohérence et fraîcheur dans les bases de données réparties. Mémoire d'HDR.

[Gançarski *et al.*, 2002] Gançarski, S., Naacke, H., Pacitti, E. et Valduriez, P. (2002). Parallel processing with autonomous databases in a cluster system. *In Cooperative Information Systems (CoopIS*.

[Gardarin, 1999] Gardarin, G. (1999). *Bases de données*. Eyrolles.

[Gesbert, 2005] Gesbert, N. (2005). *Etude de la formalisation des spécifications de bases de données géographiques en vue de leur intégration*. Thèse de doctorat, Université Marnes la Vallée.

[Gilgen, 1999] Gilgen, M. (1999). Méta-information de données géoréférencées. Rapport technique, Ecole Polytechnique Fédérale de Lausanne.

[GINIE, 2004] GINIE (2004). *Geographic Information Network In Europe*. Projet de recherche européen. http ://www.ec-gis.org/ginie/.

[GML, 2007] GML (2007). *Geography Markup Language (GML) Encoding Standard*. OpenGIS®. Reference number : OGC 07-036 Version : 3.2.1.

[Günter, 1989] Günter, O. (1989). Database support for multiple representations. *In Workshop on Multiple Representations Initiative 3, National Center for Geographic Information and Analysis (NCGIA)*, pages 50–51.

[Günther et Voisard, 1997] Günther, O. et Voisard, A. (1997). *Metadata in Geographical and Environmental Data Management*. Managing Mutimedia Data : Using Metadata to Integrate and Apply Digital Data.

[Goodchild *et al.*, 1992] Goodchild, M., Guoquing, Y. et Y.Shiren (1992). Development and test of an error model for categorical data. *International Journal of Geographical Information Systems*, 6(2):87–107.

[Gotthard *et al.*, 1992] Gotthard, W., Lockemann, P. et Neufeld, A. (1992). A system-guided view integration for object oriented databases. *IEEE Transactions on Knowledge*, 4(1):1–22.

[Gray, 1980] Gray, J. (1980). A transaction model. Rapport technique, IBM Research Laboratory.

[Gray *et al.*, 1996] Gray, J., Helland, P., O'Neil, P. et Shasha, D. (1996). the dangers of replication and a solution. *In ACM SIGMOD International Conference on Management of Data*, pages 173–182.

[Gray, 1978] Gray, W. (1978). Notes on database operating systems. *Lecture Notes in Computer Science*, (60):393–481.

[Grum et Vasseur, 2004] Grum, E. et Vasseur, B. (2004). How to select the best dataset for a task ? *In Fourth International Symposium on Spatial Data Quality, ISSDQ'04*, pages 197–206.

[Guptill, 1995] Guptill, S. (1995). *Elements of Spatial Data Quality*, chapitre Temporal Information. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Guting, 1994] Guting, R. (1994). An introduction to spatial database systems. *VLDB Journal*, 3:357–399.

[Harding, 2005] Harding, J. (2005). *Qualité des données vectorielles : perspective d'un producteur de données*, chapitre 10. Qualité de l'information géographique, Traités IGAT, Hermès Sciences, Lavoisier. ISBN 2-7462-1097-5.

[Helal *et al.*, 1996] Helal, A. A., Heddaya, A. A. et Bhargava, B. B. (1996). *Replication Techniques in Distributed Systems*. Kluwer Academic Publishers.

[Herdorfer et Bianchin, 1998] Herdorfer, M. et Bianchin, A. (1998). Un modèle structurel pour métadonnées. *In Journées Cassini*.

[Herlihy et Wing, 1990] Herlihy, M. et Wing, J. (1990). Linearizability : A correctness condition for concurrent objects. *ACM Trans. Program. Lang. Syst.*, 12(3):463–492.

[Heuvelink, 2005] Heuvelink, G. (2005). *Geographical Information Systems : Principles, Techniques, Management and Applications*, volume 1, chapitre Propagation of error in spatial modelling with GIS, pages 207–217. Wiley and Sons. ISBN 978-0-471-73545-8.

[Hornsby et Engenhofer, 2000] Hornsby, K. et Engenhofer, M. J. (2000). Identity-Based Change : A Foundation for Spatio-Temporal Knowledge Representation. *International Journal of Geographical Information Science*, 14(3):207–224.

[Hunter, 2001] Hunter, G. (2001). Spatial data quality revisited. *In GeoInfo*.

[INSPIRE, 2007] INSPIRE (2007). *INfrastructure for SPatial InfoRmation in Europe*. Directive européenne.

[ISO19115, 2003] ISO19115 (2003). *Geographic Information : Metadata*. ISO/TC 211.

[ISO19139, 2003] ISO19139 (2003). *Geographic Information – Metadata. Implementation specification. ISO/WD 19139*. International Organisation for Standardisation.

[ISO8402, 1994] ISO8402 (1994). *Quality management and quality assurance- Vocabulary*. International Organization for Standardization (ISO).

[ISO9000, 2000] ISO9000 (2000). *Quality management systems*. International Organization for Standardization (ISO).

[Jeansoulin, 1997] Jeansoulin, R. (1997). *Data Quality in Geographic Information : From error to Uncertainty*, chapitre sing Spatial Constraints as Redundancy Information to Improve Geographical Knowledge. M.Goodchild and R.Jeansoulin. Hermes.

[Jeansoulin et Wilson, 2002] Jeansoulin, R. et Wilson (2002). Model-based semantics for ontologies of geographic information. *In Gisciences*.

[J.Pouliot *et al.*, 2001] J.Pouliot, Bedard, Y., Nadeau, M. et Larrivee, S. (2001). Extraction, diffusion and integration of geospatial data updates (majic project). *In Workshop : geo information fusion and revision.*

[Juran *et al.*, 1974] Juran, J., Gryna, F. et Bingham, R. (1974). *Quality Control Handbook.* McGrawHill, New York.

[Kadri-Dahmani, 2005] Kadri-Dahmani, H. (2005). *Mise à jour incrémentale des bases de données géographiques et maintien de leur cohérence.* Thèse de doctorat, Université Paris Nord.

[Kainz, 1995] Kainz, W. (1995). *Elements of Spatial Data Quality*, chapitre Logical Consistency. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Kavouras *et al.*, 1995] Kavouras, M., Paradissis, D., Ecker, R. et Jansa, J. (1995). *Geographic Information Systems : Materials for a Post Graduate Course vol. 1 : Spatial Information*, chapitre Data Sources for GIS. A. Frank (Ed.), Technical University Vienna.

[Kawell *et al.*, 1988] Kawell, L., Beckhart, S., Halvorsen, T., Ozzie, R. et Greif, I. (1988). Replicated document management in a group communication system. *In Conference on Computer Supported Cooperative Work (CSCW).*

[Kemme, 2000] Kemme, B. (2000). *Database Replication for Clusters of Workstations.* Thèse de doctorat, Institut fédéral technologique suisse de Zurich.

[Kermarrec *et al.*, 2001] Kermarrec, A., Rowstron, A., Shapiro, M. et Druschel, P. (2001). The icecube approach to the reconciliation of diverging replicas. *In ACM symp. on Principles of Distributed Computing.*

[Kilpelainen, 2000] Kilpelainen, T. (2000). Maintenance of multiple representation databases for topographic data. *The Cartographic Journal*, 37(2):101–107.

[Kim *et al.*, 1993] Kim, W., Choi, I., Gala, S. et Scheevel (1993). On resolving schematic heterogeneity in multidatabase systems. *Distributed and Parallel Databases*, 1(3):251–279.

[Kistler et Satyanarayanan, 1992] Kistler, J. et Satyanarayanan, M. (1992). Disconnected operation in the coda file system. *ACM Trans. Comput. Syst.*, 10(5):3–25.

[Kronenberg *et al.*, 1986] Kronenberg, N., Levy, H. et Strecker, W. (1986). Vaxclusters : A closely-coupled distributed system. *In ACM Trans. on Comp. Sys. (TOCS)*, pages 130–146.

[Kumar et Satyanarayanan, 1993] Kumar, P. et Satyanarayanan, M. (1993). Logbased directory resolution in the coda file system. *In 2nd International Confernce on Parallel and Distributed Information Systems (PDIS).*

[Larson *et al.*, 1989] Larson, J., Navathe, S. et Elmasri, R. (1989). A theory of attribute equivalence in databases with application to schema integration. *IEEE Transaction on Software Engineering*, 15(4):449–463.

[Laurini, 1996] Laurini, R. (1996). Raccordement géométrique de bases de données géographiques fédérées. *Ingénierie des systèmes d'informations*, 4(3):361–388.

[Laurini, 1999] Laurini, R. (1999). Spatial Multi-Database Topological Continuity and Indexing : a Step Towards Seamless GIS Data Interoperability. *International Journal of Geographic Information Science*, 12(4):373–402.

[Laurini et Raffort, 1994] Laurini, R. et Raffort, F. M. (1994). Topological reorganization of inconsistent geographical databases : A step toward their certification. *Computer and Graphics*, 18(6):803–813.

[le Roux, 2003] le Roux, P. (2003). Versioning, lineage, timestamps and temporal databases. Tehnical paper, Intergraph.

[Leblanc et Villot, 2003] Leblanc, N. et Villot, E. (2003). Environnement on line, volet dynamique et cohérence :proposition d'étude et démonstration. Rapport technique, EADS DS.

[Leclercq, 2000] Leclercq, E. (2000). *Interopérabilité sémantique des systèmes d'information géographique : une approche basée sur la médiation de contexte.* Thèse de doctorat, Université de Dijon.

[Leclercq *et al.*, 1998] Leclercq, E., Benslimane, D. et Yetongnon, K. (1998). Isis : une architecture multi-agents pour l'interopérabilité des sig. *In Colloque national SMAGET "Modélisation et systèmes multi-agents pour la gestion de l'environnement et des territoires.*

[Lecordix *et al.*, 2005] Lecordix, F., Jahard, Y., Lemarié, C. et Hauboin, E. (2005). The end of carto 2001 project : Top100 based on bdcarto database. *In 8th ICA WORKSHOP on Generalisation and Multiple Representation.*

[Lemarié, 1996] Lemarié, C. (1996). Etat de l'art sur l'appariement. Rapport technique, IGN, service de la recherche. Rapport technique DT/9600022/SRAP.

[LePape, 2005] LePape, C. (2005). *Contrôle de Qualité des Données Répliquées dans un Cluster.* Thèse de doctorat, Université de Paris 6.

[Libourel, 2003] Libourel, T. (2003). Autour de la conception de systèmes complexes. modélisation, évolution, infrastructures. Mémoire d'HDR.

[Lidl *et al.*, 1994] Lidl, K., Osborne, J. et Andmalcolm, J. (1994). Drinking from the firehose : Multicast usenet news. *In USENIX Winter Technical Conference.*

[Longley *et al.*, 2005] Longley, P. A., Goodchild, M. F., Maguire, D. J. et Rhind, D. W. (2005). *Geographic Information Systems and Science.* John Wiley and Sons. ISBN : 978-0-470-87001-3.

[Luzet, 1998] Luzet, C. (1998). Megrin's gddd, moving to distributed metadata. *In EOGEO98.*

[Maguire *et al.*, 1992] Maguire, D., Stickler, G. et Browning, G. (1992). Handling complex objects in georelational gis. *In Spatial Data Handling.*

[Masser, 1998] Masser, I. (1998). The first generation of national geographic information strategies. *In Third Global Spatial Data Infrastructure Conference.*

[Matos. *et al.*, 1997] Matos., J., Bento, J., Dionisio, A., Goncalves, A., Salgneiro, J., Martius, N. et Regateiro, F. (1997). An operational methodology for quality analysis and update of digital cartography. *In Joint European Conference and exhibition on Geographical Information.*

[Medeiros et Jomier, 1994] Medeiros, C. B. et Jomier, G. (1994). Using versions in GIS. *In Database and Expert Systems Application*, pages 465–474.

[METAFOR, 2005] METAFOR (2005). *Gamme de produits CARGENE. Format de fichiers de métadonnées.* République Française, Ministère de la défense, IGN/DT.TN/03.054.

[Moellering, 1987] Moellering, H. (1987). A draft proposed standard for digital cartographic data, national committee for digital cartographic standards. Rapport technique, American Congress on Surveying and Mapping.

[Monnet, 2006] Monnet, S. (2006). *Gestion des données dans les grilles de calcul : support pour la tolérance aux fautes et la cohérence des données.* Thèse de doctorat, Université Rennes 1.

[Moore, 1995] Moore, K. (1995). The lotus notes storage system. *In International Conference on Management of Data (SIGMOD).*

[Mummert *et al.*, 1995] Mummert, L., Ebling, M. et Satyanarayanan, M. (1995). Exploiting weak connectivity for mobile file access. *In 15th Symposium on Operating Systems Principles (SOSP).*

[Mustière *et al.*, 2004] Mustière, S., Sheeren, D. et Gesbert, N. (2004). Unifications des bases de données géographiques : Recherches au laboratoire cogit de l'ign. *Revue Géomatique Expert*, (32/33):50–54.

[MySQL, 2003] MySQL (2003).

[Nebert, 2004] Nebert, D. (2004). *eveloping Spatial Data Infrastructures : The SDI Cookbook.*

[Nogueras-Iso *et al.*, 2005] Nogueras-Iso, J., Zararaga-Soria, F. et Muro-Medrano, P. (2005). *Geographic Information Metadata for Spatial Data Infrastrucutres.* Resources, Interoperability and Information Retrieval, Springer editions. ISBN :3-540-24464-6.

[OQLF, 2004] OQLF (2004). *Le grand dictionnaire terminologique.* Office Québécois de la Langue Française : http ://www.granddictionnaire.ca.

[Oracle, 1996] Oracle (1996). *Oracle7 Server Distributed Systems Manual, volume 2.* Oracle Corporation.

[Oracle®, 2003] Oracle® (2003). Oracle® database advanced replication. Rapport technique, Oracle Corporation. Part No. B10732-01.

[Oster, 2005] Oster, G. (2005). *Replication optimiste et cohérence des données dans les environnements collaboratifs repartis.* Thèse de doctorat, Université Nancy 1.

[Otto *et al.*, 2004] Otto, H.-U., Capra, L., Löwenau, J., Sabel, H., Bracht, A., Angenvoort, J., Beuk, L., Bruns, K., Aleksiæ, M., Zeidler, W. et Fischer, C. (2004). Specification of actualisation strategies, map components version control and interfaces. Rapport technique, Tele Atlas. Project Name : Actual and dynamic MAP for transport telematics applications.

[Pacitti *et al.*, 1999] Pacitti, E., Minet, P. et Simon, E. (1999). Maintaining replica consistency in lazy master replicated databases. *In 1Journées Bases de Données Avancées (BDA).*

[Pacitti *et al.*, 2001] Pacitti, E., Minet, P. et Simon, E. (2001). Replica consistency in lazy master replicated databases. *Distributed and Parallel Databases*, 9(3):237–267.

[Pacitti et Valduriez, 1998] Pacitti, E. et Valduriez, P. (1998). Replicated databases : Concepts, architectures and techniques. *Network and Information Systems Journal*, (1):4–5.

[Peerbocus *et al.*, 2002] Peerbocus, A., Jomier, G. et Badard, T. (2002). A Methodolgy for Updating Geographic Databases using Map Version. *In Symposium on Geospatial Theory, Processing and Applications*.

[Petersen *et al.*, 1997] Petersen, K., Spreitzer, M., Terry, D., Theimer, M. et Demers, A. (1997). Flexible update propagation for weakly consistent replication. *In Symposium on Operating Systems Principles*.

[Peuquet, 1981] Peuquet, D. (1981). An examination of techniques for reformatting digital cartographic data. *Cartographica*, 18:21–48.

[Pierkot et Mustiere, 2007] Pierkot, C. et Mustiere, S. (2007). Gestion des mises à jour concurrentes dans des jeux de données géographiques répartis. *In Colloque international de géomatique et d'analyse spatiale, SAGEO*.

[Pierkot *et al.*, 2006] Pierkot, C., Mustiere, S., Ruas, A. et Hameurlain, H. (2006). Using metadata to help the integration of several multi-sources set of updates. *In 9th GSDI Conference, Santiago, Chile*.

[Pierkot *et al.*, 2005] Pierkot, C., Mustiere, S., Ruas, A., Hameurlain, H. et Raynal, L. (2005). Modelling heteregeneous and distributed spatial datasets in an update context. *In 22th International Cartographic Conference, ICA Publications*.

[Pierkot et Raynal, 2004a] Pierkot, C. et Raynal, L. (2004a). Environnement on line, volet dynamique et cohérence : Dossier de définition du programme d'études complémentaires. Rapport technique, EADS DS.

[Pierkot et Raynal, 2004b] Pierkot, C. et Raynal, L. (2004b). Environnement on line, volet dynamique et cohérence : Dossier de justification des choix effectués. Rapport technique, EADS DS.

[Pouliot *et al.*, 2001] Pouliot, J., Bédard, Y., Larrivée, S. et Nadeau, M. (2001). Projet m@jic : Problématique de mises à jour de données géospatiales. Rapport technique, Centre de recherche en géomatique, Université Laval, Québec. méthode et état d'avancement.

[Preguiça *et al.*, 2001] Preguiça, N., Martins, J. L., Domingos, H. et Duarte, S. (2001). Data management support for asynchronous groupware. *In ACM Conference on Computer supported cooperative work*.

[Puppo et Dettori, 1995] Puppo, E. et Dettori, G. (1995). Towards a formal model for multiresolution spatial maps. *In SSD'95*.

[Puricelli, 2000] Puricelli, A. (2000). *Réingénierie et Contrôle Qualité des Données en vue d'une Migration Technologique*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon.

[Querzola et Billout, 1995] Querzola, J. et Billout, M. (1995). *Guide de la cartographie informatisée*. EURO-Vista.

[Rajabifard et Williamson, 2001] Rajabifard, A. et Williamson, I. (2001). Spatial data infrastructures : Concept, sdi hierarchy and future directions. *In GEOMATICS'80 Conference*.

[Ramamritham et Chrysanthis, 1996] Ramamritham, K. et Chrysanthis, P. K. (1996). A taxonomy of correctness criteria in database applications. *VLDB*, 5(1):85–97.

[Ramirez, 1997] Ramirez, R. (1997). Development of a common framework to express raster and vector datasets. *In Auto Carto 13, ACSM/ASPRS*.

[Ratner, 1998] Ratner, D. (1998). *Roam : A scalable replication system for mobile and distributed computing*. Thèse de doctorat, University of California, Los Angeles.

[Raynal, 2005] Raynal, L. (2005). Environnement on line, volet dynamique et cohérence. etude 1 : Détermination et utilisation des identifiants pour les bases vecteurs. Rapport technique, EADS DS.

[Raynal et Ruffier, 2005] Raynal, L. et Ruffier, I. (2005). Environnement on line, volet dynamique et cohérence. etude 2 : Analyse des limites du mécanisme d'intégration par appariement local. Rapport technique, EADS DS.

[Reiher *et al.*, 1994] Reiher, P., Heidemann, J., Ratner, D., Skinner, G. et Popek, G. (1994). Resolving file conflicts in the ficus file system. *In USENIX Summer Technical Conference*.

[ReV !Gis, 2004] ReV !Gis (2004). Revision of the uncertain geographic information. Rapport technique, MIT Labotary for Computer Sciences and RSA data Security. Projet n°IST-1999-14189, www.lsis.org/REVIGIS/Full/index.html.

[Rhind, 2001] Rhind, D. (2001). Global and national geographic information policies, practice and education in a g-business world, in gi in europe : Integrative, interoperable, interactive. *In 4th AGILE conference*.

[Rivest, 1992] Rivest, R. (1992). RFC1321 : The MD5 Message Digest Algorithm. Rapport technique, MIT Labotary for Computer Sciences and RSA data Security.

[Rodriguez, 2000] Rodriguez, M. (2000). *Assessing Semantic Similarity among Entity Classes*. Thèse de doctorat, University of Maine.

[Rouet, 1991] Rouet, P. (1991). *Les données dans les systèmes d'information géographique*. Hermès, Paris.

[Saito et Shapiro, 2005] Saito et Shapiro, M. (2005). Optimisitic replication. *ACM Computing Surveys*, 37(1):42–81.

[Saito *et al.*, 1998] Saito, Y., Mogul, J. et Verghese, B. (1998). A usenet performance study. Rapport technique.

[Salgé, 1995] Salgé, F. (1995). *Elements of Spatial Data Quality*, chapitre Semantic Accuracy. S.C.Guptill et J.L.Morisson. Oxford : Elsevier.

[Scholl *et al.*, 1996] Scholl, M., Voisard, A., Peloux, J. P., Raynal, L. et Rigaux, P. (1996). *SGBD Géographiques : spécificités*. International Thomson Publishing France.

[Servigne *et al.*, 2005] Servigne, S., Lesage, N. et Libourel, T. (2005). *Composantes qualité et métadonnées*, chapitre 12. Qualité de l'information géographique, Traités IGAT, Hermès Sciences, Lavoisier. ISBN 2-7462-1097-5.

[Seshadri et Garrett, 2000] Seshadri, P. et Garrett, P. (2000). Sqlserver for windows ce - a database engine for mobile and embedded platforms. *In Conference on Data Engineering (ICDE)*.

[Sheeren, 2005] Sheeren, D. (2005). *Méthodologie d'évaluation de la cohérence inter-représentations pour l'intégration de bases de données spatiales. Une approche combinant l'utilisation de métadonnées et l'apprentissage automatique.* Thèse de doctorat, Université Paris 6.

[Shepherd, 1992] Shepherd, I. (1992). *Geographic Information Systems*, chapitre Information Integration and GIS, pages 337–358. Longman Scientific and Technical.

[Sheth, 1999] Sheth, A. (1999). *Changing Focus on Interoperability in Information Systems : from Systems, Syntax, Structure to Semantics*, pages 5–29. Goodchild M. and al. editions, Kluwer Academic Publisher. Interoperating Geographic Information Systems.

[Solar et Doucet, 2002] Solar, G. V. et Doucet, A. (2002). Médiation de données : solutions et problèmes ouverts. *In 2eme assises nationales du GDR 13.*

[Spaccapietra et al., 1992] Spaccapietra, S. et al. (1992). Model independent assertions for integration of heterogeneous schemas. *Very Large DataBases Journal*, 1(1):81–126.

[Spaccapietra et C.Parent, 1996] Spaccapietra, S. et C.Parent (1996). Intégration de bases de données : panorama des problèmes et des approches. *Ingénierie des systèmes d'information*, 4(3):333–358.

[Spaccapietra et Parent, 1991] Spaccapietra, S. et Parent, C. (1991). Conflicts and correspondence assertions in interoperable databases. *In ACM SIGMOD RECORD.*

[Spencer et Lawrence, 1998] Spencer, H. et Lawrence, D. (1998). *Managing Usenet.* O'Reilly and Associates.

[Spéry et Libourel, 1998] Spéry, L. et Libourel, T. (1998). Vers une structuration des métadonnées. *In Journées Cassini.*

[Stephan *et al.*, 1993] Stephan, E., Vckovski, A. et Bucher, F. (1993). Virtual data set : An approach for the integration of incompatible data. *In Auto Carto 11, ASPRS/ACSM.*

[Tellez et Servigne, 1998] Tellez, B. et Servigne, S. (1998). Updating urban database with aerial photographs. *In* Ltd., E. S., éditeur : *Computation, Environment and Urban Systems*, volume 21/2, pages 133–145.

[Terry *et al.*, 1995] Terry, D., Theimer, M., Petersen, K., Demers, A., Spreitzer, M. et C.H.Hauser (1995). Managing update conflicts in bayou, a weakly connected replicated storage system. *In ACM Symposium on Operating Systems Principles.*

[Terry *et al.*, 2000] Terry, D., Theimer, M., Petersen, K. et Spreitzer, M. (2000). An examination of conflicts in a weakly-consistent, replicated application. Rapport technique, Personal communication.

[Thomas, 1979] Thomas, R. H. (1979). A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems*, 4(2):80–209.

[Timpf et Frank, 1995] Timpf, S. et Frank, A. (1995). A multi-scale dag for cartographic objects. *In Auto Carto 12*.

[Tsou, 2002] Tsou, M.-H. (2002). An operational metadata framework for searching, indexing, and retrieving distributed geographic information services on the internet. *In Second International Conference on Geographic Information Science*.

[Ubeda, 1997] Ubeda, T. (1997). *Contrôle de la Qualité Spatiale des Bases de données Geographiques : Cohérence Topologique et Correction d'Erreurs*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon.

[Ubeda et Servigne, 1996] Ubeda, T. et Servigne, S. (1996). Geometric and topological consistensy of spatial data. *In 1st International Conference on Geocomputation*.

[Vangenot *et al.*, 2002] Vangenot, C., Parent, C. et Spaccapietra, S. (2002). Modeling and manipuling multiple representation of spatial data. *In 10th Spatial Data Handling Synopsium*.

[Vasseur, 2004] Vasseur, B. (2004). *Modélisation de l'information de qualité dans les applications Géographiques*. Thèse de doctorat, Université Aix-Marseille 1.

[Vasseur *et al.*, 2005] Vasseur, B., Jeansoulin, R. et Devillers, R. (2005). *Évaluation de la qualité externe de l'information géographique : une approche ontologique*. Qualité de l'information géographique, Traités IGAT, Hermès Sciences, Lavoisier. ISBN 2-7462-1097-5.

[Veregin, 1989] Veregin, H. (1989). *Accuracy of Spatial Databases*, chapitre Error Modeling for the Map Overlay Operation. Taylor et Francis.

[Vesperman, 2003] Vesperman, J. (2003). *Essential CVS*. O'Reilly and Associates.

[Vögele et Schlieder, 2002] Vögele, T. et Schlieder, C. (2002). The use of spatial metadata for information retrieval in peer-to-peer networks. *In 5th AGILE Conference on Geographic Information Science*.

[VMAP0, 1999] VMAP0 (1999). *Military specification for Vector MAP Level 0*. National Imagery and Mapping Agency's (NIMA).

[VMAP1, 1995] VMAP1 (1995). *Military specification for Vector MAP Level 1*. National Imagery and Mapping Agency's (NIMA).

[VMAP2, 1993] VMAP2 (1993). *Military Specification MIL-V-89032 Vector Smart Map (VMap) Level 2 (Draft)*. National Imagery and Mapping Agency's (NIMA).

[VMAPUrban, 2000] VMAPUrban (2000). *Military specification for Urban Vector Map*. National Imagery and Mapping Agency's (NIMA).

[VPF, 1998] VPF (1998). *General specification for Vector Product Format*. National Imagery and Mapping Agency's (NIMA).

[VRF, 2000] VRF (2000). *DIgital Geographic information Exchange STandard, Part 2 : Vector Relational Format*. Digital Geographic Information Working Group, members of OTAN. MIL-PRF-0089049(NIMA).

[W3C, 2006] W3C (2006). *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. W3C. http ://www.w3.org/XML/.

[Wang et Strong, 1996] Wang, R. W. et Strong, D. M. (1996). Beyond accuracy : what data quality means to data consumers. *Journal of Management Information systems*, 12(4):p5–34.

[Özsu et Valduriez, 1999] Özsu, T. et Valduriez, P. (1999). *Principles of Distributed Database Systems*. Prentice Hall.