

Université Pierre et Marie Curie – Paris VI
Mémoire d'habilitation à diriger des recherches

Bases de données : de l'objet à
l'interopérabilité

Bruno Defude

Soutenu le 24 février 2005 devant le jury composé de :

Mr Michel Adiba	Professeur, Université de Grenoble 1, Président
Mme Anne Doucet	Professeur, Université de Paris 6, Rapporteur
Mr Mokrane Bouzeghoub	Professeur, Université de Versailles Saint Quentin, Rapporteur
Mr Jean-Luc Hainaut	Professeur, Université de Namur, Rapporteur
Mr Guy Bernard	Professeur, Institut National des Télécommunications, Examineur
Mr Patrick Valduriez	Directeur de recherche, INRIA, Examineur

Travail effectué au sein du GET - Institut National des Télécommunications

Le travail présenté dans ce mémoire est le résultat d'un long cheminement à côté de beaucoup de personnes avec qui j'ai toujours eu beaucoup de plaisir à travailler et que je voudrais ici remercier. Je tiens à commencer par Yves Chiaramella, mon responsable de DEA puis de doctorat qui a cru en moi et m'a appris le métier de « chercheur ». J'ai ensuite toujours eu la chance d'avoir des responsables qui m'ont fait confiance et m'ont laissé aller de l'avant et je tiens pour cela à remercier principalement Yves, Michel Adiba ainsi que Guy Bernard.

La recherche c'est avant tout un travail en équipe. Sans pouvoir être exhaustif, je tiens à remercier les personnes avec qui j'ai travaillé dans les projets IOTA (Marie-France, Catherine, Patrick, Dalila, Philippe, ...), puis Aristote (Christophe, Christine, Hervé, Jean-Pierre, l'autre Christine et Christian, Samer, Mauricio, Pascal, ...) et enfin à l'INT (Michel, Jean-Michel, Claire, Amel, Samir, Guy, Sophie, Ana, Djamel, ...).

De nombreux étudiants, devenus pour certains des collègues, des amis, m'ont aidé à construire ma réflexion au fil des années et plus particulièrement Christophe, Claudia, Javam, Tarik, Anne, Nicolas et Freddy.

Une mention spéciale à Hervé, l'ami de tous les instants, toujours prêt à répondre quand on a besoin de lui, ou à refaire le monde du sport quand il est temps de faire une pause !

Beaucoup de personnes m'ont poussé à rédiger ce mémoire. Deux d'entre elles ont joué un rôle particulièrement important : Stanislaw Budkowski, pour m'avoir incité à prendre une période de congé d'études et m'avoir amicalement mis sous pression et Anne Doucet qui m'a permis de concrétiser ce projet de congé d'études au LIP6 et a accepté de rapporter ce travail.

Bien que le rôle de rapporteur soit difficile et consommateur de temps, Mokrane Bouzeghoub et Jean-Luc Hainaut ont accepté de jouer le jeu. Qu'ils en soient remerciés. Je suis particulièrement heureux d'avoir dans ce jury Michel Adiba et Patrick Valduriez qui sont deux chercheurs que j'estime profondément. De plus, Michel Adiba m'a fait l'honneur de présider ce jury.

Enfin, je tiens à remercier ma famille qui a, elle aussi, joué un rôle important dans mon métier. Mes parents tout d'abord, qui m'ont permis de poursuivre mes études aussi longuement, en ne comprenant pas toujours ce choix mais en me faisant confiance. Mes « femmes » ensuite, qui ont dû supporter au quotidien les humeurs, pas toujours agréables, de l'enseignant-chercheur qui doute.

TABLE DES MATIERES

1.	Introduction	7
1.1	Préambule	9
1.2	Introduction	9
2	Des SGBD à objets à l'interopérabilité.....	13
2.1	Les SGBD à objets.....	15
2.1.1	Modèle de données à objets	16
2.1.1.1	Contexte et introduction.....	16
2.1.1.2	Modèle de données Estrella	17
2.1.1.3	Intégration de documents dans un SGBD à objets	17
2.1.1.4	Bilan et comparaison avec les autres approches	19
2.1.2	Expression et validation de contraintes d'intégrité	19
2.1.2.1	Contexte et introduction.....	19
2.1.2.2	Comparaison avec les autres approches	21
2.1.2.3	Modèle de cohérence comportemental	22
2.1.2.4	Description des contraintes	23
2.1.2.5	Expression de transactions emboîtées	25
2.1.2.6	Bilan.....	26
2.2	Interopérabilité des bases de données	26
2.2.1	Interopérabilité relationnel-objet : le générateur d'applications BD Aristote ..	28
2.2.1.1	Introduction et contexte	28
2.2.1.2	Modèle et langage Aristote	29
2.2.1.3	Générateur d'applications Aristote.....	30
2.2.1.4	Bilan.....	31
2.2.2	Intégration de schéma de SIG à base de logique terminologique	31
2.2.2.1	Introduction et contexte	31
2.2.2.2	Le modèle GEOCOOPM	32
2.2.2.3	Conflits spatiaux.....	33
2.2.2.4	Le processus d'intégration	33
2.2.2.5	Bilan.....	37
3	Vers l'interopérabilité sémantique.....	41
3.1	Introduction	43
3.2	Modèle de description sémantique de ressources éducatives.....	43
3.2.1	Introduction et contexte	43
3.2.2	Architecture logique	44
3.2.3	Modèle de description des ressources	46
3.2.4	Sélection et Adaptation.....	50
3.2.4.1	Mode ressource	50
3.2.4.2	Mode concept	51
3.2.4.3	Mode requête.....	52
3.2.4.4	Règles d'adaptation	53
3.2.5	Implantation	54
3.2.5.1	Correspondance modèle de description avec RDF.....	54
3.2.5.2	SeRQL comme niveau d'inférence	55
3.2.6	Bilan	56
3.3	Pair-à-pair sémantique.....	57
3.3.1	Introduction.....	57
3.3.2	Organisation sémantique d'un réseau pair-à-pair	59
3.3.3	Bilan et comparaison avec les autres approches	60

4	Conclusion et perspectives	63
4.1	Bilan	65
4.2	Perspectives	66
4.2.1	Modèles et langages de description de ressources	66
4.2.2	Interopérabilité	67
4.2.3	Personnalisation	69
5	Bibliographie	71

1. Introduction

1.1 Préambule

Ce document présente le résultat des recherches que j'ai menées depuis la soutenance de ma thèse de doctorat d'informatique en juillet 1986. Cette thèse portait sur l'utilisation de techniques d'intelligence artificielle (systèmes experts, traitement de langage naturel principalement) pour améliorer la composante interrogation d'un système de recherche d'informations textuelles. Après cette thèse, j'ai orienté mes activités de recherche vers les systèmes de gestion de base de données (SGBD). J'ai commencé à travailler sur les modèles de données à objets et leur utilisation pour les applications géographiques en 1987 dans mon premier poste d'ingénieur de recherche à la société Matra DPMAC (Direction des Programmes et Moyens d'Aide au Commandement). J'ai continué sur cette thématique en l'étendant au problème d'interopérabilité entre bases objets et relationnelles dans le cadre du projet Aristote au Laboratoire de Génie Informatique de Grenoble où j'ai occupé un poste de Maître de Conférences en informatique entre octobre 1988 et octobre 1992. Ensuite, j'ai intégré l'Institut National des Télécommunications (INT), une des écoles du Groupe des Ecoles de Télécommunications (GET), comme Maître de Conférences. Là j'ai continué à travailler sur un projet commun avec Grenoble sur l'expression et la validation de contraintes d'intégrité dans les SGBD à objets. Puis, j'ai commencé à m'intéresser aux problèmes liés à la répartition et à l'interopérabilité. J'ai mené un projet sur l'interopérabilité des Systèmes d'Information Géographique (SIG) de 1996 à 1999 puis j'ai travaillé sur la problématique du test d'applications réparties de type CORBA. Depuis trois ans, je travaille sur une approche d'interopérabilité sémantique entre ressources pédagogiques (enseignement à distance) et plus récemment sur l'intégration de « sémantique » dans les systèmes de recherche à grande échelle fondés sur des architectures pair-à-pair (« *peer-to-peer* » ou P2P).

Je suis directeur d'études à l'INT depuis 1999 (grade « équivalent » à professeur mais avec un accent plus fort mis sur la réalisation de contrats et la responsabilité de grands domaines d'enseignement). Je suis responsable du groupe bases de données du département informatique de l'INT depuis janvier 2002 et à ce titre j'encadre les activités d'enseignement-recherche de deux puis maintenant trois autres enseignants-chercheurs ainsi que de plusieurs doctorants et stagiaires.

Pour terminer ce préambule, je voudrais noter que j'ai toujours eu la chance de travailler dans des institutions et avec des personnes qui m'ont toujours fait confiance et m'ont laissé une grande autonomie. J'ai également toujours essayé de mener mes activités de recherche en liaison avec des besoins applicatifs réels, ce qui s'est souvent matérialisé par des contrats de recherche et développement (DRET-DGA, Bull, EDF-DER, C.N.E.T. France Télécom, projet RNRT Castor, ...) ou des projets de recherche (AS CNRS Mobilité / Accès aux données, ACI GRID Padoue, ...).

1.2 Introduction

De nombreux courants ont marqué l'évolution de la recherche en bases de données ces quinze dernières années. Le domaine des bases de données est à la frontière de plusieurs autres domaines informatiques (langages de programmation, systèmes d'exploitation, répartition et parallélisme, conception de systèmes d'information) et est donc influencé par les évolutions de ces domaines. Je voudrais mettre l'accent sur les courants qui ont eu, à mon avis, le plus d'influence :

- l'approche à objet : venant du monde des langages de programmation (depuis Simula jusqu'à Java en passant par C++ et tant d'autres), elle a influencé pratiquement tous les domaines de l'informatique et donc les bases de données. En matière de bases de données à objets, après beaucoup de promesses initiales, le marché a tranché et les

- SGBD « purs objets » ont disparu. Il n'en reste pas moins qu'un certain nombre de leurs idées sont passées dans les produits commerciaux (les SGBD relationnels objets) et qu'une bonne partie du travail réalisé se réutilise aujourd'hui dans le contexte de XML par exemple (modèles de stockage, langage de requêtes) ;
- les architectures logicielles à base de composants : les SGBD commerciaux sont de plus en plus gros (ils se positionnent sur les 2 tiers serveurs) et restent très monolithiques. Pourtant, il y a un besoin pour certaines applications à fortes contraintes (systèmes embarqués par exemple) de n'utiliser qu'un sous-ensemble des fonctionnalités nécessaires (par exemple de la reprise sur panne sans contrôle de concurrence). Cela donne lieu à des travaux sur les SGBD à base de composants. Un SGBD est vu comme un ensemble inter-relié de composants que l'on peut choisir de déployer en tout ou partie. Cela permet éventuellement aussi de remplacer un composant par un autre (plus performant ou plus léger par exemple) ;
 - la répartition et le parallélisme : le modèle de programmation dominant n'est plus strictement séquentiel, mais on utilise de plus en plus des modèles d'exécution répartis à couplage faible (« répartition ») ou à couplage fort (parallélisme). En matière de bases de données cela a donné naissance aux BD réparties (qui permettent d'accéder à un ensemble de sources de manière uniforme) et aux BD parallèles (permettent d'exécuter une requête complexe sur un ensemble de nœuds en partitionnant les données et/ou la requête ou bien de supporter un grand nombre de requêtes simples et simultanées) [Oszu 00]. Les BD parallèles ont suivi l'évolution des architectures parallèles (multiprocesseurs à mémoire partagée, multiprocesseurs à mémoire privée, grappes de PC). Les produits commerciaux ont suivi cette évolution et proposent tous un support minimal de la répartition et du parallélisme (version parallèle d'Oracle par exemple). L'évolution est encore loin d'être terminée car l'écart entre la recherche et les produits commerciaux reste important et de plus de nouveaux problèmes arrivent comme la prise en compte de la mobilité et le passage à l'échelle en matière de répartition (début du *Peer to Peer computing* ou P2P) ;
 - le web : il est évident que la technologie informatique qui a le plus d'impact ces dernières années est la technologie du web. Le couplage d'un langage de définition de document hypermédia avec un protocole permettant d'accéder de manière distante à ces documents a révolutionné l'utilisation de l'informatique, y compris pour le grand public. Ce développement s'est fait initialement complètement en dehors du contexte bases de données, mais très vite est apparu le besoin de diffuser les données contenues dans les SGBD sur le web (web dynamique). De nombreux travaux et produits ont alors proposé des couplages plus au moins réussis entre web et BD permettant la production déclarative de sites web à partir de requêtes BD ainsi que des techniques de caches pour optimiser les accès,... Ces deux dernières années, le courant s'est accentué avec l'avènement du successeur du langage HTML, à savoir XML. Le modèle de données sous-jacent à XML est très proche des modèles de données semi-structurés et tous les travaux réalisés dans ce cadre ont pu être adaptés au contexte XML. XQuery, le nouveau langage de requêtes pour documents XML est très influencé par la communauté des chercheurs en BD ;
 - l'évolution technologique générale caractérisée d'une part par la montée en puissance rapide des serveurs (puissance des processeurs, taille des mémoires vives et de masse) et d'autre part par les progrès des réseaux change fondamentalement notre façon de travailler. Il faut par exemple, intégrer le fait que l'accès au réseau (au moins local) soit beaucoup plus rapide que l'accès au disque (à la latence près), ce qui remet complètement en question les algorithmes d'optimisation de requêtes réparties proposés dans les années 1980 ! Parallèlement à cette évolution des serveurs, de

nouveaux dispositifs matériels (comme les cartes à puces ou les assistants personnels numériques) sont apparus, destinés à la mobilité et qui ont des caractéristiques techniques très spécifiques (bonne puissance des processeurs mais mémoire relativement petite et surtout peu performante en écriture, et aussi faible autonomie en énergie). Progressivement, la demande en applications sophistiquées augmente sur ces nouveaux dispositifs et le besoin de SGBD se fait alors sentir ;

Parallèlement à cette évolution technologique, les applications elles aussi ont changé :

- de plus en plus d'applications utilisent les SGBD et apportent leurs contraintes : de nombreux travaux ont porté sur la gestion des données multimédia, puis des données spatiales, temporelles, ...
- l'avènement des réseaux, l'omniprésence de l'informatique et les besoins de réactivité toujours croissants des organisations conduisent à aller vers de plus en plus d'intégration des applications. Cette intégration ne peut se faire sans prendre en compte les problèmes d'interopérabilité dû aux différentes formes d'hétérogénéité (plates-formes matérielles, logiciels hétérogènes et surtout hétérogénéité sémantique).

Ces 15 années de recherche se sont situées dans ce tourbillon d'évolution et nous avons essayé à notre mesure de contribuer à l'effort de recherche, sans trop se perdre !

Nos principales contributions se sont situées dans le domaine des SGBD à objets pendant les premières années, puis se sont plus orientées dans le domaine de la répartition et de l'interopérabilité à partir du moment où j'ai intégré l'INT.

Il faut noter aussi la richesse de ce domaine qui permet de travailler sur des sujets variés sans changer de problématique. Cela peut conduire à une certaine dispersion, mais c'est sans doute nécessaire si on veut appréhender l'ensemble du domaine.

	Modélisation	hétérogénéité	répartition	adaptation	applicatif
<u>Estrella</u>	objet				SIG, document
<u>Aristote générateur</u>	Objet Méta- modélisation	génération			
<u>GECO</u>	objet	génération			
<u>GEOCOOPM</u>	Logique terminologique	Syntaxique sémantique	médiateur		SIG
<u>ASMAD</u>			mobilité		
<u>Padoue</u>	Méta-données		Médiateur P2P	contexte	environnement
<u>SIMBAD</u>	Méta-données Ontologie workflow	sémantique		utilisateur	e-learning entreprises virtuelles
<u>RARE</u>	Méta-données		P2P	utilisateur	

Tableau 1-1 : positionnement des recherches

Mes recherches ont porté sur différentes fonctions de gestion de données (voir tableau 1-1), en partant assez souvent de l'étude des problèmes posés par un domaine particulier pour en déduire des solutions en matière de technologie bases de données.

Dans ce tableau, les projets que j'ai animés ont leur nom souligné. Le principal domaine abordé est la modélisation, activité au cœur des bases de données. Pratiquement tous les projets ont abordé ce domaine et notamment la définition de modèles de données en utilisant divers formalismes (objets, méta-données, logique terminologique). L'hétérogénéité est présente dans plusieurs projets. Tout d'abord à Grenoble dans le cadre du projet Aristote avec

l'interopérabilité relationnel-objet traitée par génération à partir d'un méta-modèle, puis à l'INT avec l'interopérabilité des SIG (projet GEOCOOPM) traitée via la définition d'un modèle canonique et plus récemment la définition de méta-données éducatives dans SIMBAD. Le troisième volet couvre les modèles de répartition avec la médiation dans GEOCOOPM et Padoue, la mobilité pour l'accès aux données dans l'action spécifique CNRS « Mobilité / Accès Aux données » (ASMAD) et enfin le P2P dans Padoue (travail initié lors de mon séjour d'études au LIP6) et RARE. Un dernier volet plus récent est celui de l'adaptation que j'aborde dans le cadre de Padoue et plus largement au sein du nouveau projet GET SIMBAD que j'ai lancé depuis janvier 2004.

Dans ce document, je ne vais pas retracer l'ensemble de ces activités (une présentation synthétique est faite dans le CV détaillé), mais je vais me focaliser sur les points qui me semblent les plus importants.

Dans la première partie, je résume mes travaux passés en les organisant autour de la thématique SGBD à objets (modèle de données et contraintes d'intégrité) puis de l'interopérabilité syntaxique (définition de modèles de données canoniques). Mes travaux récents, portant sur la dimension sémantique des bases de données, sont présentés dans la deuxième partie en commençant par la description et l'adaptation de ressources puis en abordant le pair-à-pair sémantique. En conclusion, je présente les projets de recherche sur lesquels je commence à travailler et qui me semblent suffisamment porteurs d'avenir.

2 Des SGBD à objets à l'interopérabilité

Dans cette partie nous présentons successivement nos travaux sur les SGBD à objets puis sur l'interopérabilité des bases de données.

2.1 Les SGBD à objets

Dans les années 1985-95, une grande partie de l'activité de recherche en bases de données s'est consacrée aux SGBD à objets. L'intérêt pour cette technologie était triple :

- continuer l'évolution des modèles de données sémantiques. En effet, le modèle relationnel fait montre de ses faiblesses pour supporter certaines applications et de nombreuses propositions sont faites ;
- rapprocher le monde des bases de données de celui des langages de programmation et de celui des systèmes d'exploitation. Historiquement, ces trois mondes ont évolué de manière séparée alors qu'ils partagent un certain nombre de fonctionnalités. Les objets ont servi de dénominateur commun et de nombreux travaux fructueux ont eu lieu à l'intersection de ces domaines ;
- exploiter les capacités d'extensibilité de ces systèmes. Les « nouvelles » applications ont fait apparaître le besoin d'extensibilité (support de nouveaux types, de nouvelles techniques d'accès, ...) qui s'adapte assez bien au contexte objet. Cette problématique se poursuit aujourd'hui avec les approches composants.

Cette vogue pour les SGBD à objets a été très nette dans le monde de la recherche mais quasi inexistante dans le monde commercial. Alors que la recherche sur les bases de données relationnelles s'est diffusée très rapidement dans le monde commercial, celle sur les objets a été un échec. Les raisons en sont sûrement multiples. Problème commercial d'abord, les grands du secteur relationnel ont « verrouillé » et les entreprises qui venaient d'investir massivement dans le relationnel n'ont pas voulu d'une nouvelle technologie. Problème fonctionnel ensuite. Hormis quelques niches applicatives, les avancées des SGBD à objets n'ont pas correspondu aux attentes des entreprises. Problème technique aussi, la maturité des SGBD à objets n'était pas forcément très bonne. Cet échec commercial ne doit cependant pas masquer le fait que de nombreux travaux sur les SGBD à objets ont trouvé leur suite dans les bases de données semi-structurées (modèle de données, langage de requêtes, gestion d'objets complexes) et notamment avec XML.

Sur le plan de la recherche, les SGBD à objets ont été marqués par deux grands projets. Le projet Orion à MCC Austin [Kim 90], précurseur du domaine (milieu des années 80), a ouvert la voie sur de nombreux points et ensuite (années 90). Le projet Altaïr, dirigé par F. Bancilhon, a regroupé des chercheurs de l'INRIA, du LRI et du GIP Altaïr et a produit le système O₂ [Bancilhon 92, Adiba 93] qui a ensuite été industrialisé par la start-up O₂Technology (disparue depuis après son rachat par une société américaine). OQL, le langage de requêtes standard pour les SGBD à objets, émane également d'O₂Technology. Altaïr étant un projet français, il a d'autant plus marqué la recherche française sur ce sujet, sachant, qu'en plus, le système O₂ a été largement diffusé dans la communauté et a servi de base logicielle à beaucoup d'autres projets de recherche. [CACM 91] fait une présentation synthétique des grands projets de SGBD à objets au début des années 1990.

Ma recherche sur les SGBD à objets s'est déroulée de 1987 à 1997 entre Matra DPMAC, le projet Aristote au LGI/IMAG et l'INT. Sur cette thématique, j'ai travaillé sur deux points principaux :

- la définition de modèle de données à objets. Lorsque ce travail a été effectué (87-88) il n'y avait pas encore de consensus autour d'un modèle de données à objets et comme beaucoup nous avons proposé un modèle. Nous avons aussi étudié comment utiliser la richesse des concepts objets et les mécanismes d'extensibilité d'un SGBD à objets pour proposer un couplage SGBD – systèmes documentaires;

- l'expression et la validation de contraintes d'intégrité (92-97) : une fois un modèle de données objet stabilisé, l'ensemble des fonctionnalités des SGBD relationnels a dû être revu dans le contexte des SGBD à objets. Nous nous sommes intéressés au problème de l'intégrité en essayant de l'intégrer au mieux aux concepts objets.

Nous présentons maintenant successivement ces travaux.

2.1.1 Modèle de données à objets

2.1.1.1 Contexte et introduction

Au milieu des années 80, beaucoup de travaux se sont attachés à améliorer la puissance d'expression du modèle relationnel de données et ses langages de requêtes associés. Les principaux résultats sont les modèles sémantiques (extensions du modèle Entité-Association, modèles issus de la représentation des connaissances, ...), les extensions au modèle relationnel (N1NF ou Non First Normal Form, modèles où l'hypothèse de la première forme normale est supprimée, c'est à dire où un domaine n'est plus forcément atomique). Les travaux sur les modèles de données à objets ont eu lieu dans ce contexte et peuvent se classer en deux grandes catégories :

- modèles objets « purs », c'est à dire qui font abstraction complète du modèle relationnel : on peut citer les modèles basés sur des langages de programmation (ObjectStore et C++, Orion et Lisp) ou se voulant indépendant d'un langage spécifique (le modèle O2). Cette approche se concrétisera par la production d'un modèle objet « standard », le modèle de l'ODMG (Object Database Management Group, regroupant la plupart des éditeurs de SGBD à objets) [Cattell 97] ;
- modèles objets « hybrides » (ensuite appelés objet-relationnel), c'est à dire les propositions cherchant à étendre le modèle relationnel pour supporter des concepts objets ou tout au moins de pouvoir faire cohabiter les concepts relationnels et les concepts objets. On peut citer notamment Postgres et plus généralement la vision objet de SQL3 qui se trouve implanté maintenant dans la plus grande partie des SGBD relationnels commercialisés (Oracle, DB2, ...).

Au moment où nous avons commencé nos travaux à Matra DPMAC (1987) dans le cadre d'un contrat exploratoire DGA-DRET (support des applications géographiques à l'aide d'un modèle de données objet), la communauté scientifique n'a pas encore dégagé de consensus sur le sujet. Plusieurs projets de recherche existaient déjà (Orion par exemple), mais le projet Altaïr débutait. Nous avons donc été amenés à faire notre propre proposition qui devait répondre aux critères suivants :

- support des données géographiques et des applications associées : Matra-DPMAC était une société qui vendait des applications à nature géographique dans des contextes civils et militaires (photo-interprétation, préparation de missions aériennes, aide au commandement, ...). Le but était de définir un système de stockage et de recherche de données géographiques qui facilite le développement de ces applications. A cette époque, le simple recours à un SGBD était novateur. Les applications utilisaient simplement des systèmes de fichiers et devaient reconstruire tout ou partie des fonctions classiques d'un SGBD en fonction des besoins (contrôle de concurrence, langage de requêtes, ...)
- extensibilité : Matra-DPMAC utilisait beaucoup de bibliothèques numériques (par exemple pour le traitement d'images) et il était hors de question de les réécrire pour les intégrer dans un nouveau système. Il était fondamental de prévoir des mécanismes d'extensibilité permettant d'intégrer des codes existants sans avoir à les réécrire ;
- intégration avec les systèmes relationnels. Les contraintes industrielles nous ont imposé une solution compatible avec la technologie relationnelle.

Notre proposition s'est donc située dans le courant des modèles objet-relationnels. Je présente maintenant les grandes lignes du modèle de données Estrella [Damier 89, Damier 91]. Je décris ensuite une expérience d'extensibilité avec Omega en y ajoutant un « type abstrait » document [Damier 88].

2.1.1.2 Modèle de données Estrella

Le modèle de données Estrella repose sur les concepts, classiques aujourd'hui, d'objet, de classe, de fonctions et d'héritage. Nous donnons juste ici quelques détails sur le concept de fonction qui est un peu original dans Estrella, ainsi que sur l'extensibilité.

Une fonction est une opération définie par sa signature (nom, arguments et types en entrée et en sortie) et qui possède un corps (externe au modèle de données). Nous avons fait le choix de ne pas les rattacher explicitement aux classes. Une fonction peut être qualifiée de multi-méthode permettant une sélection qui n'est pas réduite à un seul argument (le receveur). Estrella permet la définition de fonctions de même nom, sous réserve que les signatures soient différentes. Cela suppose donc à l'exécution un mécanisme de liaison dynamique qui, en fonction du type des paramètres effectifs, se lie à la « bonne » fonction. Ceci est classique dans les langages orienté-objet à la différence que la liaison ne se fait pas uniquement par rapport au type de l'objet receveur, mais par rapport à l'ensemble des arguments (c'est ce que l'on appelle des multi-méthodes [Chambers 99]). La sélection est faite en utilisant la relation d'ordre sur les signatures. Le problème est le coût de calcul associé (d'autant plus que ce calcul est fait lors de l'exécution). De nombreux travaux ont été faits pour essayer d'améliorer les performances de ce type de liaison multi-méthodes (voir [Chambers 99] pour une proposition et une présentation de l'état de l'art).

L'extensibilité se fait par la possibilité de créer de nouvelles classes dans le modèle, mais cela ne permet pas d'introduire de nouvelles méthodes d'accès ou de nouveaux supports physiques. Ceci est pourtant nécessaire si on veut introduire des abstractions comme point, segment, polygone dans un système d'information géographique. Nous avons donc ajouté un mécanisme d'extensibilité via les classes descripteurs. La classe abstraite DESCRIPTOR est prédéfinie dans le treillis de classe Estrella. L'administrateur peut ensuite créer des sous-classes de cette classe DESCRIPTOR. Pour ce faire, il doit implémenter un certain nombre de fonctions abstraites comme l'égalité, la création, la suppression, ...

2.1.1.3 Intégration de documents dans un SGBD à objets

Estrella se voulant extensible, nous avons voulu en faire la preuve en introduisant un type abstrait texte [Damier 88]. Le but était double. D'une part étudier comment les concepts objets peuvent servir à mieux modéliser des documents structurés et d'autre part étudier comment introduire une structure de données spécifique pour supporter de l'indexation textuelle (codage superposé).

Les systèmes documentaires et les SGBD se sont longtemps développés de manière parallèle sans jamais vraiment chercher à s'intégrer. A l'époque de ces travaux (1988), les deux communautés communiquaient assez peu. Venant à l'origine de la communauté document et travaillant dans la communauté bases de données, il m'était apparu naturel d'essayer de faire cohabiter ces deux mondes.

Notre proposition a porté sur les points suivants :

- ajout d'un type abstrait *Text* au moyen d'une classe descripteur spécifique ;
- utilisation des constructeurs de type tableau et ensemble pour définir la structure logique d'un document ;
- utilisation de la notion de trajet pour exprimer les requêtes navigationnelles dans le document ;

- ajout d'une structure de donnée spécifique ainsi que d'opérations associées pour prendre en compte la recherche sur le contenu.

Nous avons utilisé la signature de fichiers [Faloutsos 84] comme index. Le principe est d'utiliser des fonctions de hachage pour représenter le document à partir du hachage de son contenu. Des variations sont possibles selon l'unité utilisée pour le hachage et la manière d'assembler les hachages des différentes parties du document. Assez fréquemment, le hachage s'applique sur les mots avec un regroupement par phrase. L'assemblage peut se faire par concaténation ou par superposition. Tout le problème est de trouver l'équilibre entre la précision de la recherche et la taille de la signature. Plus la signature est compacte, plus la probabilité d'obtenir des erreurs de recherche (« false drop ») est grande. La concaténation est plus précise mais nécessite plus de place, alors que la superposition perd de l'information mais est plus compacte. De nombreuses variantes du codage superposé ont été proposées et sont évaluées dans [Lee 95].

Avec cette structure, le modèle booléen continue à s'appliquer, avec l'introduction de manière naturelle des informations sur la structuration du texte (comme la phrase), ce qui permet de prendre en compte des opérateurs comme l'adjacence de deux termes.

Le type de requêtes sur le contenu que l'on veut exprimer est le suivant :

```
SELECT low(d)
FROM document d
Where d.author="Adiba" and belongs(("integrity AND (data ADJ base%)", d.text)
```

Où `low(d)` désigne le niveau dans la structure logique qui est choisi (ici le plus petit) et l'opération `belongs` applique une équation booléenne de recherche sur le contenu du document. Cette équation comprend les opérateurs booléens AND, OR, EXCEPT ainsi que l'opérateur ADJ (adjacence de deux mots). Le % joue le rôle de joker et représente n'importe quelle chaîne de caractères (il est utilisable aussi bien en partie droite qu'en partie gauche).

Pour pouvoir résoudre cette équation, le système a dû préalablement indexer le document pour en obtenir une représentation sous forme de codage superposé.

Pour cela, le document est découpé en blocs logiques. Chaque bloc logique comprend un nombre fixe de phrases. Un bloc logique va être composé par le OU logique des signatures de tous les mots composant les phrases. Pour pouvoir gérer le % et le ADJ, une phrase est découpée en une séquence de triplets consécutifs avec recouvrement et un triplet « fictif » est ajouté entre les deux mots pour le ADJ. De cette manière, si un mot A est inclus dans un mot B, la signature de A sera incluse dans la signature de B.

La résolution d'une équation de recherche se fait de la manière suivante :

- 1- résolution de l'équation de recherche sur l'ensemble des signatures. Comme le filtrage sur les signatures n'est pas exact et que certaines parties de l'équation de recherche ne peuvent être résolues sur les signatures (EXCEPT notamment), il faut donc transformer l'équation de recherche sur les documents en une équation de recherche sur les signatures qui n'est pas équivalente mais qui va produire un sur-ensemble du résultat ;
- 2- filtrage du résultat de l'étape précédente par des techniques classiques de recherche de chaînes de caractères et les parties non traitées par l'étape précédente vont pouvoir l'être. L'interprétation ensembliste des opérations AND, OR et EXCEPT a été revue pour s'adapter à des documents structurés (donc des arbres). L'intérêt de l'étape précédente est quand même d'avoir considérablement réduit l'espace de recherche ;
- 3- filtrage final du résultat pour adapter la réponse au niveau souhaité dans la structure logique (par exemple `low(d)`).

L'intégration dans Estrella a été réalisée via une classe descripteur *Text* qui implante les fonctions de recherche et d'indexation. La structuration logique des documents utilise la

puissance de modélisation des modèles objets et peut être représentée par différentes classes et références entre elles. Une modélisation plus fine de cette structure logique nécessiterait d'étendre le modèle de données (voir [Woelk 86] comme exemple).

2.1.1.4 Bilan et comparaison avec les autres approches

Le modèle de données Estrella a été prototypé dans le système Omega qui utilise le SGBD Oracle. Fonctionnellement, Omega implante la plupart des concepts proposés dans Estrella. La couche de stockage est classique, les structures de données Estrella sont représentées dans un ensemble de relations d'Oracle et la correspondance suit les principes détaillés dans [Valduriez 87]. Le langage de requête SQLOmega est une extension de SQL permettant l'utilisation d'expressions de chemin (appelées trajets) et l'appel de fonctions. L'évaluation d'une requête SQLOmega se fait par réécriture en son équivalent en SQL Oracle. Cette réécriture n'est pas toujours simple et notamment il faut, pour les requêtes avec fonctions, introduire des relations intermédiaires. L'évaluation SQLOmega ne repose pas sur une algèbre pour objets (peu de travaux avaient été fait dans ce sens à l'époque) et utilise des techniques simples d'optimisation.

Une évaluation complète du prototype n'a pas été faite (notamment parce que le projet s'est arrêté chez Matra DPMAC faute de financement). Les quelques mesures effectuées ont montré que la correspondance objet –relationnel introduit un surcoût important dans l'évaluation des requêtes (il faut associer du cache de requêtes et/ou des techniques de regroupement de données). Le bilan de l'implantation est donc mitigé. Fonctionnellement, Omega a fait la preuve de faisabilité des idées introduites dans Estrella, par contre il n'a pas eu la maturité, ni les performances nécessaires pour être utilisé dans les applicatifs de Matra DPMAC (ce qui était l'objectif initial).

Le modèle de données Estrella peut être comparé à l'extension objet de SQL telle qu'elle est définie dans la norme SQL3 [Gardarin 99]. SQL3 étend la notion de domaine (ou plutôt implante vraiment la notion de domaine telle que définie initialement dans le modèle relationnel de données) via un langage de définition de domaine à la type abstrait (structure de données encapsulée par des opérations). On trouve également des types « objets » (avec OID) et « valeurs » (sans OID). SQL3 utilise trois constructeurs de type complexes (*set*, *multiset* permet d'avoir des doublons, *list*). Une relation est définie soit sur les types prédéfinis habituels, soit sur les types utilisateurs. Il est possible de définir une relation comme sous-relation d'une autre, elle hérite alors du schéma de celle-ci. Des fonctions peuvent être ajoutées sur les relations. En ce qui concerne le langage de manipulation, des extensions sont proposées qui permettent d'accéder aux types pointés, d'invoquer les fonctions, de manipuler les constructeurs complexes, ... Estrella peut donc être vu comme un précurseur de SQL3 et Omega être un des premiers systèmes relationnel-objet. La principale différence porte sur les fonctions qui sont externes aux relations en Estrella.

Ce projet nous a permis d'acquérir une bonne compétence dans le domaine des SGBD à objets et de la valoriser par plusieurs publications [Damier 88, Damier 91] ainsi que la thèse de C. Damier que j'ai encadrée pour Matra DPMAC [Damier 89].

2.1.2 Expression et validation de contraintes d'intégrité

2.1.2.1 Contexte et introduction

Dans les SGBD, les contraintes d'intégrité sont un ensemble de propriétés qui définissent les états valides de la base de données, c'est à dire les états qui satisfont aux contraintes du monde réel. Le but des contraintes d'intégrité est d'affiner la définition du schéma de la base de données afin d'augmenter la « qualité » des données stockées. Cette information supplémentaire peut aussi être utilisée pour faire de l'optimisation sémantique de requêtes. La

prise en compte des contraintes d'intégrité est étudiée depuis fort longtemps dans le contexte des bases de données relationnelles et déductives. [Nicolas 82, Greffen 93] en font un état de l'art complet respectivement pour les bases de données déductives et pour les SGBD relationnels.

Le problème est double : d'une part être capable d'exprimer le plus grand nombre possible de contraintes d'intégrité et d'autre part être capable de les vérifier efficacement. Bien entendu, ceci est antagoniste et il faut trouver le compromis entre expressivité et efficacité.

Les contraintes sont habituellement classifiées en contraintes statiques et contraintes dynamiques. Les contraintes statiques s'expriment à l'aide de formalismes issus de la logique du premier ordre et portent sur l'état courant de la base de données (par exemple si une personne est de sexe masculin, elle ne peut pas avoir un nom de jeune fille). Les contraintes dynamiques portent sur plusieurs états de la base de données (par exemple le salaire d'un employé ne peut diminuer) et peuvent s'exprimer à l'aide de logiques temporelles.

Un système de vérification de contraintes doit répondre aux questions suivantes : quand vérifier les contraintes ? comment vérifier de manière efficace ? et comment réparer ?

Le modèle classique de transactions plates définit la transaction comme unité de contrôle. Pendant l'exécution d'une transaction, la BD peut être incohérente mais elle ne peut l'être à la fin de celle-ci. Cela impose donc que les contraintes soient vérifiées au plus tard à la fin de chaque transaction. Avec des modèles de transactions plus sophistiqués ou plus simplement avec des transactions réparties, la question du « quand » doit être revisitée (nous verrons plus loin nos contributions sur ce sujet).

Il est difficile d'assurer les performances des mécanismes de détection de violation de contraintes. Potentiellement il faut vérifier à la fin de chaque transaction que toutes les contraintes sont vérifiées sur toutes les données. Pour simplifier, on suppose qu'une BD est initialement cohérente et qu'il suffit donc d'assurer que l'exécution d'une transaction ne viole aucune contrainte d'intégrité pour garantir que la BD reste cohérente. Pour ce faire, la vérification est restreinte aux seules données manipulées par la transaction et aux seules contraintes concernées par ces données.

Lorsqu'une contrainte est violée, il faut « réparer » la BD pour qu'elle retrouve un état cohérent. Si on est capable de connaître l'instruction responsable de la violation, il suffit de la compenser. Cependant, il n'est pas forcément facile de le faire, d'autant plus qu'il peut s'agir non pas d'une seule instruction mais d'une séquence. La manière la plus simple de réparer consiste donc à annuler/compenser l'ensemble de la transaction en faute. Bien entendu cela peut conduire à défaire beaucoup trop de choses, mais cela a l'avantage d'être très simple à mettre en œuvre. Nous présenterons également mes contributions sur ce sujet.

La prise en compte des contraintes d'intégrité dans les SGBD à objets pose des problèmes nouveaux.

En ce qui concerne l'expression des contraintes, il y a besoin de plus d'expressivité pour prendre en compte les nouveaux constructeurs introduits par les modèles à objets. Il est également possible d'aller plus loin en essayant de structurer la définition des contraintes de manière analogue à la structuration des schémas objets (hiérarchie de classes, mécanisme d'héritage, surcharge). Nous montrerons nos apports sur ce thème.

En ce qui concerne la vérification, dans les SGBD à objets il n'y a plus de séparation entre les données et le code des applications. Ceci permet de faire des manipulations conjointes données/code qui apportent de nouvelles informations et permettent des vérifications plus efficaces. De la même manière, les réparations peuvent être effectuées de manière plus précise. Enfin, dans les SGBD à objets, le modèle de transaction plat, inadapté dans ce contexte, est abandonné au profit de modèles plus riches comme les transactions emboîtées.

Nous présentons dans la suite notre contribution en la situant par rapport aux autres approches proposées pour les SGBD à objets. Nous décrivons tout d'abord notre modèle d'intégrité (appelé modèle comportemental), puis nous définissons notre langage d'expression de contraintes avant d'aborder les problèmes de vérification et de réparation dans le cadre de transactions emboîtées.

2.1.2.2 Comparaison avec les autres approches

Dans les SGBD à objets deux grandes voies ont été menées. La première essaye d'adapter au monde objet les solutions apportées dans le monde relationnel. L'autre essaye plutôt d'adapter des solutions venant du monde des langages de programmation. Dans le premier courant, se trouvent les solutions basées sur l'utilisation de règles actives (ou ECA, Événement Condition Action). Les contraintes d'intégrité sont déclarées au niveau du schéma et compilées en un certain nombre de règles actives qui vont effectuer la vérification et la réparation [Ceri 94]. Le problème majeur est que la plupart des modèles de règles actives souffrent d'une sémantique peu claire. De plus des propriétés élémentaires comme la terminaison ne sont pas assurées.

L'autre approche reprend des techniques utilisées dans le monde des langages de programmation. On peut citer des mécanismes d'exécution (exceptions, pré et post-conditions) utilisés soit pour spécifier des contrats (comme dans le langage Eiffel), soit pour simplifier le traitement des erreurs (exceptions). Un langage de spécification de contraintes OCL [OMG 03] a été défini pour UML. Il permet de décrire des contraintes sur les opérations de type invariant, pré-condition, post-condition ou commande gardée. Le langage est assez riche et permet de manipuler le receveur de l'opération, le résultat, des collections et les associations. Une sémantique de l'héritage a été définie sur les invariants et pré(post)-conditions inspirée du principe de substitution de Liskov (« là où une instance d'une classe est attendue, elle peut être remplacée par une instance de n'importe quelle sous-classe »). Les invariants sont hérités et ne peuvent être que renforcés par surcharge, de même pour les post-conditions alors que les pré-conditions ne peuvent être qu'affaiblies par surcharge.

Des techniques d'analyse statique du code ou d'interprétation abstraite ont également été proposées. L'intérêt de ces techniques d'analyse du code des transactions et des contraintes est de pouvoir définir à priori quelles sont les contraintes susceptibles d'être violées par une transaction donnée. Cela permet de restreindre considérablement les vérifications à faire. [Benzaken 95] et al. dans le cadre du langage de programmation de bases de données Themis ont proposé des techniques d'analyse statique du code des transactions. [Benzaken 98] a ensuite utilisé des techniques de transformateurs de prédicats (initialement introduits par Dijkstra) pour des contraintes statiques puis dynamiques [Benzaken 00]. Le principe est le suivant : pour une méthode m et une contrainte c , on définit les transformateurs :

- si c est une contrainte vérifiée par les entrées de m , un transformateur en avant va produire la post-condition $POST(m, c)$ qui est nécessairement vérifiée par les données de sortie de m ;
- si c est une contrainte devant être vérifiée par les sorties de m , un transformateur en arrière va produire la pré-condition $PRE(m, c)$ qui est une condition suffisante sur les entrées de m pour satisfaire c sur ses sorties.

Ensuite, on peut définir qu'une méthode m est saine relativement à une contrainte c si :

$$POST(m, c) \Rightarrow c \Rightarrow PRE(m, c)$$

La preuve des implications utilise la logique du premier ordre pour les contraintes statiques et la logique temporelle linéaire du premier ordre pour les contraintes dynamiques. Pour cette deuxième catégorie, la preuve ne peut pas toujours être apportée car cette logique est incomplète. Les auteurs proposent alors deux définitions de la validité : la validité (si la démonstration est faite) et une notion plus faible de k -validité qui démontre la formule pour des histoires d'une taille k bornée (pour des transactions de k mises à jour). Cette approche est

intéressante même si nous n'avons pas d'information sur le passage à l'échelle (combien de contraintes, quelle taille de code, ...).

Enfin une dernière approche au problème de la vérification consiste à définir le langage d'écriture des transactions de telle manière qu'il ne puisse pas violer de contraintes d'intégrité. Cette approche est étudiée dans le cadre des schémas transactionnels de Abiteboul et Vianu [Abiteboul 89] par [Bidoit 98]. Des résultats intéressants ont été produits pour des classes de contraintes comme les dépendances fonctionnelles ou l'intégrité référentielle. La généralisation paraît difficile, de plus cela oblige à restreindre les langages de programmation existants pour qu'ils rentrent dans les contraintes fixées. L'idée de contraindre plutôt l'écriture du code que les états de la base est quand même intéressante et peut être abordée par la prise en compte de contraintes d'ordonnement des opérations (telle opération doit s'exécuter avant telle autre par exemple).

L'interaction entre vérifications de contraintes et modèle de transaction a été peu étudiée. Il existe des travaux qui recouvrent partiellement cette problématique dans des approches type workflow comme le ConTract Model [Wachter 94] ou les transactions longues [Dayal 90]. A notre connaissance, hormis nos propositions, seuls [Doucet 99, Doucet 01] ont traité pleinement cette problématique dans le cadre de transactions emboîtées centralisées et réparties. Enfin, [Gupta 93] aborde le problème de la vérification efficace de contraintes exprimées sur des données réparties.

2.1.2.3 Modèle de cohérence comportemental

Notre approche part plutôt des langages de programmation en l'adaptant au contexte SGBD à objets. L'idée est de redéfinir le modèle de cohérence classique de bases de données en l'adaptant au contexte des SGBD à objets. Nous proposons de passer d'une notion de BD cohérente à une notion d'exécution cohérente. Une BD est dans un état e_i cohérent, si initialement e_0 était cohérente (même principe) et si la séquence d'opérations S_i qui a fait passer la BD de l'état e_0 à l'état e_i est cohérente. Nous passons donc d'une cohérence sur les états à une cohérence sur les séquences d'opérations (voir figure 2-1).

La cohérence sur les opérations est définie par un modèle de type logique de Hoare [Hoare 69]. Une pré et une post-condition sont associées à chaque opération. La pré-condition exprime les bonnes conditions d'évaluation de l'opération alors que la post-condition contraint les états finaux possibles de l'opération. L'exécution d'une opération est définie comme cohérente si ses pré et post-conditions sont vérifiées. Cette définition de la cohérence est très forte et doit être assouplie pour prendre en compte deux problèmes. Premièrement, certains comportements ne peuvent être caractérisés localement mais seulement globalement à la fin de la transaction. Ces comportements doivent être considérés comme suspects au niveau de l'opération et doivent être évalués globalement à la fin de la transaction quand toute l'information nécessaire sera disponible. En fonction du contexte, la décision sera prise d'accepter ou de refuser au niveau global le comportement suspect. Deuxièmement, un mécanisme de contrôle de l'intégrité doit être suffisamment souple pour prendre en compte des exceptions aux règles générales [Esculier 90]. Par conséquent, avant de rejeter un comportement, il nous faut vérifier que cela ne correspond pas à un cas exceptionnel.

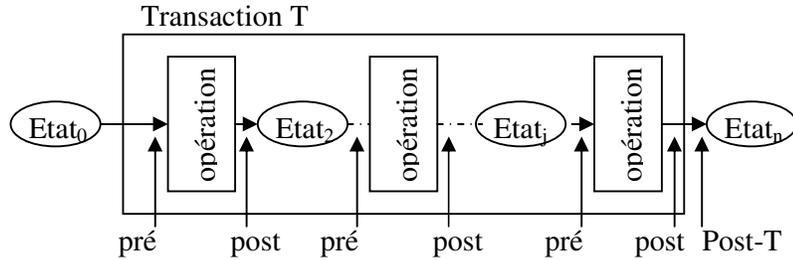


Figure 2-1 : modèle de cohérence comportementale

Pour assouplir notre mécanisme, nous associons aux pré(post)-conditions un mécanisme d'exceptions. Une pré(post)-condition violée déclenche une exception. Celle-ci peut être traitée immédiatement (cas d'un contrôle local) ou bien de manière différée en fin de transaction (cas d'un contrôle global). La partie traitement d'une exception (le corps) correspond aux exceptions à la règle générale exprimée par la pré(post)-condition. Ces exceptions à la règle sont exprimées sous la forme d'une suite de couples condition - action. Les actions possibles sont soit *continue* (cas d'une exception à la règle générale), soit *abort* (règle sans exception). Pour simplifier l'écriture du corps, seuls les couples correspondant à une exception à la règle générale sont décrits et la partie action étant implicite (*continue*) elle est omise. Si toutes les conditions du corps sont fausses, la règle est violée. Si une des conditions au moins est vraie, il s'agit d'une exception à la règle générale.

Une exécution d'opération *op* est cohérente si elle satisfait la pré-condition de *op* et la post-condition de *op*. L'exécution d'une transaction *T* est cohérente si l'exécution de toutes les opérations composant la transaction est cohérente et si la post-condition de *T* est satisfaite (la pré-condition d'une transaction est supposée toujours satisfaite). L'ensemble des exceptions différées déclenchées durant l'exécution d'une transaction correspond à la post-condition de celle-ci.

Le modèle d'exceptions est simple et présente l'avantage de garantir la terminaison. En effet, l'expression d'une condition (dans la partie *when* ou dans le corps de l'exception) ne peut faire appel à des méthodes et les seules actions possibles dans le corps d'une exception sont la validation si l'expression est évaluée à *true* et l'abandon sinon. Il ne peut donc pas y avoir de déclenchement d'exceptions en cascade. Il faut noter que la reprise du contrôle après le déclenchement d'une exception se fait dans le bloc déclenchant et non pas dans le bloc englobant comme dans les modèles d'exceptions classiquement rencontrés dans les langages de programmation. Ce mode de reprise est plus adapté à la vérification d'intégrité.

2.1.2.4 Description des contraintes

Nous étendons le langage de définition de schéma d'un SGBD à objets en associant à chaque opération définies sur une classe une séquence de pré-conditions et de post-conditions qui s'ajoute à la signature de l'opération. Une pré(post)-condition est composée d'une condition et d'une exception. La condition peut porter sur l'objet receveur, l'extension de la classe, les paramètres de l'opérations. L'exception a un contexte qui est constitué à partir de l'objet receveur et des paramètres. De plus, pour chaque classe nous définissons la signature (nom et contexte) de toutes les exceptions susceptibles d'être déclenchées par les opérations de cette classe.

```

classe C1 : Object {
  Op1(C2 c2)
    precondition (cond1) : Exception1(c2)

```

```

    postcondition (cond2) : Exception2(c2)
    immediate exception Exception1(C2 c2)
    deferred exception Exception2(C2 c2)
}

```

Les exceptions ont un type (immédiat ou différé) et un corps.

```

body exception Exception2(C2 c2) : C1
{
    cond1 : action1;
    ...
    condn : actionn ;
}

```

Les conditions ne peuvent déclencher des opérations avec effets de bord (mises à jour sur la base de données). Les seules actions possibles sont *continue* ou *stop*. De cette façon, le déclenchement en cascade des exceptions est impossible. Le corps de l'exception permet de décrire les cas exceptionnels par rapport à la règle générale. S'il n'y en a pas, le corps est réduit au couple (*true* : *stop*). Par contre, le corps ne peut être réduit au couple (*true* : *continue*) qui permettrait de « court-circuiter » le mécanisme de vérification.

Le corps d'une exception immédiate est évalué juste après le déclenchement de celle-ci, alors que le corps d'une exception différé est évalué à la fin de la transaction.

Les classes étant organisées selon une relation d'héritage, nous devons également donner une sémantique de l'héritage sur les pré et post-conditions ainsi que sur les exceptions.

```

classe C3 : C1 {
    Op1(C2 c2)
    precondition(cond3) : Exception3(c2)
    immediate exception Exception3 (C2 c2)
    deferred exception Exception2(C4 c2, entier e)
}

```

L'héritage sur les pré et post-conditions est défini par union, ce qui veut dire que l'on peut seulement ajouter de nouvelles contraintes sur une opération redéfinie dans une sous-classe. Cette sémantique semble assez naturelle et permet également d'assurer que, via le mécanisme d'héritage, la vérification de contraintes ne soit pas supprimée.

L'héritage sur les exceptions est également défini par union. Implicitement une classe hérite de toutes les exceptions définies dans ses super-classes et peut en ajouter de nouvelles. La surcharge est possible. Une exception définie dans une super-classe peut avoir sa signature surchargée de manière analogue à la surcharge d'opérations en ajoutant de nouveaux paramètres ou bien en typant un paramètre existant par une sous-classe. Une opération définie dans une classe C_i ne peut donc déclencher que des exceptions définies dans C_i ou bien dans les super-classes de C_i.

La surcharge du corps des exceptions est également possible. Il suffit de redéfinir la liste des nouveaux cas exceptionnels.

Une contrainte devant se définir au niveau d'une classe cela sous-entend qu'elle doit s'exprimer (au moins) par un quantificateur universel porté sur l'extension de cette classe. Les contraintes ne comportant pas de quantificateur universel sur une extension de classe ne peuvent donc pas s'exprimer dans notre modèle. Les termes qui peuvent être utilisés dans l'expression d'une contrainte sont un objet d'une classe (le receveur de l'opération sur laquelle est définie la condition), les objets référencés par celui-ci, les arguments éventuels de l'opération et leurs objets référencés et enfin les objets appartenant à une extension de classe. Cette définition implique que les contraintes dynamiques (c'est à dire celles s'exprimant sur plusieurs états de la base de données) ne peuvent être exprimées sauf dans les cas particuliers

où les arguments de l'opération contiennent les nouvelles valeurs (l'objet receveur représente alors l'ancienne valeur et les arguments la nouvelle valeur).

2.1.2.5 Expression de transactions emboîtées

Le modèle de transaction plat s'adapte mal au contexte d'exécution des SGBD à objets. En effet, le principe d'appels en cascade d'opérations inhérent à la programmation objet, structure le flot d'exécution sous forme d'un arbre. Les modèles transactionnels de type hiérarchique comme les transactions emboîtées ne souffrent pas de ce défaut. Nous avons donc intégré notre modèle de cohérence aux transactions emboîtées. Cette intégration permet d'affiner ce modèle dans deux directions. Tout d'abord cela va permettre de mieux définir la notion d'exception différée. En effet, il y a intérêt à différer le moins possible l'évaluation d'une contrainte (plus on attend, plus la transaction a réalisé des opérations et plus le coût de son abandon est élevé). Le modèle de transactions emboîtées offre des points d'observation beaucoup plus nombreux qui sont la fin de chacune des sous-transactions participantes. Il est alors possible de choisir au niveau de quelle fin de sous-transaction une exception différée est traitée et non pas forcément la fin de la transaction racine.

L'autre avantage du modèle de transactions emboîtées est l'expression de la gestion de la reprise après une erreur. Une erreur dans une sous-transaction entraîne l'abandon de celle-ci mais pas forcément l'abandon de sa transaction racine. La gestion de l'abandon d'une sous-transaction peut alors être exprimée comme le traitement d'une exception prédéfinie *annulation_sous_transaction*. Cela permet une description uniforme de la gestion de la cohérence et du contrôle de l'exécution de transactions. Il faut remarquer que notre proposition s'intéresse aux mécanismes d'exécution nécessaires mais n'est pas une proposition de langage d'expression fourni aux programmeurs.

En ce qui concerne l'interaction entre la validation des contraintes et le modèle de transactions emboîtées, notre idée est de permettre au programmeur de définir lui-même la portée d'une exception différée, sachant que par défaut c'est la validation de la transaction racine mais que le programmeur peut choisir la validation de n'importe quelle sous-transaction. De cette manière, le programmeur peut choisir le "meilleur" moment pour évaluer le traitement d'une exception. Bien évidemment le choix de la portée ne peut pas être fixé lors de la définition de l'exception, mais lors de la définition de la transaction. Une même exception différée peut être traitée à des points différents dans des transactions différentes. L'ordre de validation (*commit*) a comme paramètres l'ensemble des exceptions différées qui doivent être traitées par la transaction validante (on peut filtrer selon la transaction déclenchante, selon l'objet déclenchant, selon les noms d'exceptions). Pour la transaction racine, le *commit* même sans paramètre traite par définition l'ensemble des exceptions différées non encore prises en compte. L'intérêt de fixer le choix des exceptions au moment du *commit* plutôt que le définir au niveau de la transaction est de permettre pour une même exécution plusieurs choix possibles. Il faut noter cependant qu'il n'est pas forcément facile pour le programmeur de déterminer à quel niveau de sous-transaction il faut traiter une exception différée. Il serait intéressant de l'aider en proposant un outil lui permettant de bien apprécier le choix qu'il fait, ainsi que de lui montrer pour chaque transaction les exceptions à traiter potentiellement. Une erreur du programmeur dans son choix peut être lourde de conséquences, puisqu'une exception traitée dans une sous-transaction inadéquate peut déclencher l'annulation de celle-ci. Une analyse statique du code de la transaction peut également déterminer la sous-transaction la plus adéquate [Doucet 99].

Une sous-transaction peut être abandonnée alors que la transaction englobante peut choisir de continuer à s'exécuter. La question est alors de savoir si les exceptions différées déclenchées

par la sous-transaction abandonnée doivent être abandonnées à leur tour ou non. Le déclenchement d'une exception différée correspond à une présomption d'erreur provoquée par l'exécution d'une certaine opération. Si l'opération est annulée (puisque la sous-transaction abandonne) alors la présomption d'erreur n'a plus de raison d'être. Les exceptions différées déclenchées par une sous-transaction qui abandonne sont donc supprimées de la liste des exceptions différées sans être traitées.

2.1.2.6 Bilan

Ce travail reste à ma connaissance la seule tentative d'intégration des contraintes d'intégrité avec les concepts objets et a proposé un modèle de cohérence novateur. L'interaction entre le processus de vérification des contraintes et le modèle transactionnel a été un travail précurseur dans ce domaine. Un prototypage a été réalisé au-dessus du SGBD à objets Versant qui implante des transactions emboîtées. Pour ce faire nous avons défini des constructions syntaxiques spécifiques pour supporter les contraintes et définir le comportement des différentes sous-transactions en matière d'intégrité. Ces constructions sont ensuite transformées par pré-compilation en appels à des fonction du « *run-time* » de gestion d'intégrité que nous avons développé. Une validation quantitative a montré les avantages apportés par notre approche qui minimise fortement le nombre de lignes de code à écrire.

Il est intéressant de pouvoir comparer notre modèle de description de contraintes avec les approches plus classiques basées sur l'expression de contraintes globales au niveau du schéma décrites sous forme logique. Il est sûr que notre approche est moins déclarative mais mieux intégrée au modèle de programmation objet. Nous avons esquissé dans [Defude 97] des principes de traduction qui permettent de passer de l'expression déclarative d'une contrainte en son équivalent en termes de conditions et d'exceptions. La transformation ne peut être complètement automatisée car il reste à régler le problème du choix de la classe de rattachement pour les contraintes multi-classes et dans certains cas la traduction est impossible (ou plus exactement dépend de la modélisation objet effectuée).

Ce travail a été initié par la thèse de doctorat de H. Martin [Martin 91] avec qui j'ai commencé ce travail [Martin 93]. Après mon départ pour l'INT, j'ai continué à travailler avec H. Martin sur ce domaine [Defude 94, Defude 95, Defude 96]. La prise en compte des transactions emboîtées et un prototypage sur le SGBD Versant [Defude 97] ont été effectués dans le cadre d'un contrat avec le CNET et ont fait l'objet des mémoires d'ingénieur ou de DEA de A. Picault, T.L. Nguyen et L. Bousaiene.

2.2 Interopérabilité des bases de données

L'interopérabilité des bases de données est un sous-problème de l'interopérabilité des systèmes (ou applications). Ce problème est devenu aujourd'hui crucial car les entreprises (et donc leurs systèmes d'information) ont besoin de plus en plus de travailler ensemble. Cela peut être pour des relations clients-fournisseurs très intégrées (cas des équipementiers dans le secteur de l'automobile par exemple), pour coopérer au sein d'un même projet (conception d'un nouveau produit), dans des logiques de fusion-acquisition ou tout simplement dans des relations clients-fournisseurs (applications de commerce électronique). Ces dernières années ont vu apparaître dans le monde industriel un nouveau type d'outil d'interopérabilité appelé EAI (*Entreprise Application Integration*). Nous pouvons le définir comme un outil intégré supportant trois grandes fonctions : transport et connectivité, transformation/adaptation des flux de données, automatisation/orchestration des processus. En quelque sorte il s'agit du mariage des intergiciels (pour le transport et la connectivité), des outils ETL (*Extract, Transform, Load*) et des moteurs de workflows (pour la partie processus). C'est également la vision proposée par le *service-oriented computing* [Papazoglou 03], où la construction d'application se fait par couplage faible entre des services. Un service est défini comme une

fonction décrite de manière abstraite via une interface, indépendamment de toute technologie. Les services sont publiés via des annuaires et peuvent ensuite y être recherchés pour les réutiliser seuls ou après composition avec d'autres services. Le modèle des services web [W3C 04] est un exemple de cette vision service et peut également être défini comme un EAI pour le web [Morley 03].

Un intergiciel permet aux développeurs de s'affranchir des problèmes de communication et plus largement de l'hétérogénéité des couches logicielles sous-jacentes (système d'exploitation, réseau, SGBD, langages de programmation, intergiciels eux-mêmes). Après avoir focalisé sur la prise en compte du plus grand nombre possible de couches d'hétérogénéité (CORBA 2 peut être vu comme un aboutissement dans ce domaine), les intergiciels actuels essaient d'intégrer dans leur modèle des aspects non fonctionnels comme la gestion de transactions, le support de la persistance des données ou la sécurité. Cela a donné sur le plan industriel, plusieurs solutions aujourd'hui en compétition, souvent basées sur des modèles de composants pour faciliter la construction des applications : il s'agit du modèle J2EE (Java 2 Entreprise Edition) de Sun, du modèle CCM (Corba Component Model) de l'OMG et de la plate-forme .NET de Microsoft. Une quatrième solution semble émerger, le modèle des services Web, basé sur XML et surtout plus simple que ses concurrents (ce qui en fait à la fois sa force et sa faiblesse). Il semble maintenant qu'aucune solution ne va l'emporter sur les autres ce qui complexifie l'interopérabilité puisqu'il faut aussi gérer l'hétérogénéité des intergiciels.

Au niveau recherche, plusieurs courants se dégagent autour de la mobilité (qui amène une complexité aussi importante que la répartition) [Mascolo 02], des systèmes contraints ou embarqués (capteurs, cartes à puce, ...) ou temps réel [Schmidt 02]. Au niveau architectural une tendance forte est aux architectures réflexives capables de fournir des mécanismes d'adaptation et d'auto-configuration [Kon 02].

Pour permettre de s'abstraire des intergiciels sous-jacents, l'OMG a lancé en 2000 une nouvelle approche architecture appelée MDA (Model-Driven Architecture) [Soley 00] qui repose sur un principe de décomposition en trois niveaux : le modèle (par exemple un diagramme UML), le méta-modèle (par exemple UML) et un méta-méta-modèle (le MOF, Meta-Object Facility de l'OMG). L'idée sous-jacente est de pouvoir générer un modèle donné à partir d'un de ses méta-modèles (donc par exemple de générer un modèle J2EE ou bien un modèle CCM à partir d'un même méta-modèle). Cette approche a donné naissance à une nouvelle thématique de recherche appelée par certains « ingénierie des modèles » (voir par exemple [Bernstein 00], [Melnik 03] ou [Bezivin 03]).

A côté de ces thématiques centrées sur l'abstraction des différences « syntaxiques » entre les différentes couches logicielles formant un système, il existe une autre forme d'hétérogénéité (plus difficile mais au moins aussi fréquente) basée sur les différences sémantiques entre systèmes. Il s'agit ici de pouvoir faire interopérer des systèmes qui divergent selon les concepts manipulés, ce qui nécessite de pouvoir comparer, reconnaître les similitudes et les différences entre ces concepts. Il faut donc être capable de décrire leur sémantique pour éviter des algorithmes de comparaison primitifs basés sur la stricte égalité de nom. Dans le domaine des services web, des formalismes de description sémantique commencent à être proposés (par exemple OWL-S [OWL 04]).

Si on se restreint maintenant au sous-problème d'interopérabilité des bases de données, la thématique abordée traite à la fois les aspects syntaxiques et sémantiques. Sur le plan syntaxique, le problème est celui de l'accès uniforme (via un schéma global) à un ensemble de source de données hétérogènes. De nombreuses architectures ont été proposées de la fédération [Sheth 90] à la notion de médiateur [Vargas 02]. Les médiateurs définissent un schéma global et offre un langage unique d'accès aux données alors qu'un *wrapper* réalise l'adaptation entre le langage du médiateur et celui de la source. Il existe deux grandes

approches de médiation : dans l'approche LAV (*Local As View*), chaque source est une vue sur le schéma global alors que dans l'approche GAV (*Global As View*), c'est le schéma global qui est une vue sur les différentes sources. Les avantages – inconvénients sont symétriques. L'approche LAV simplifie l'ajout et le retrait de sources mais nécessite une évaluation complexe des requêtes alors que c'est le contraire dans l'approche GAV. Une approche mixte dite GLAV a été proposée. De nombreuses techniques d'optimisation de requêtes ont été proposées pour toutes ces approches mais il faut néanmoins toujours implanter un nouveau *wrapper* quand un nouveau type de source est ajouté.

Sur l'intégration sémantique différentes approches ont été proposées. Une première classe de systèmes s'appuie sur des outils issus de l'analyse automatique des langues et notamment de dictionnaires [Metais 97]. Une seconde classe utilise plutôt des outils issus de la représentation des connaissances et particulièrement les logiques de description qui offrent un bon compromis entre pouvoir d'expression et propriétés calculatoires [Borgida 02, Baader 02]. Enfin, le troisième courant est issu du « web sémantique » [Bernes-Lee 01] qui propose une pile de modèles sémantiques comprenant entre autre un niveau ontologique et un niveau d'inférence (les logiques de description ne sont d'ailleurs pas exclues de cette vision).

Nos travaux dans ce domaine ont commencé dans le cadre du projet Aristote sur la définition d'un modèle de données pivot entre le modèle relationnel et les modèles de données objet. Ce modèle, mis en œuvre via le langage PEPLM, pouvait être ensuite être implanté aussi bien sur des SGBD relationnels que des SGBD à objet. Une expérimentation a été faite dans ce sens en utilisant Oracle et O₂. Ce travail a plus porté sur la spécification « neutre » (indépendante d'une technologie) d'applications bases de données que sur l'interopérabilité proprement dit. Il pourrait être qualifié maintenant d'approche à la MDA [Soley 00].

Ensuite, j'ai travaillé sur l'interopérabilité entre SIG et plus précisément sur des algorithmes d'intégration de schémas de BD spatiales. L'intégration était faite en définissant les schémas à intégrer dans un modèle pivot basé sur une logique terminologique et en utilisant les primitives des logiques terminologiques pour construire l'intégration. Ce travail était plus orienté sur des aspects sémantiques de l'hétérogénéité et leur modélisation à l'aide de langages issus de la communauté intelligence artificielle.

Dans la suite, nous allons détailler successivement ces deux projets.

2.2.1 Interopérabilité relationnel-objet : le générateur d'applications BD Aristote

2.2.1.1 Introduction et contexte

Le projet Aristote était un projet de recherche commun entre le laboratoire de Génie Informatique et le centre de recherche Bull de Grenoble. Il s'est déroulé de 1989 à 1993. Ce projet fédérait des équipes de recherche en bases de données (universités et Bull), en conception de systèmes d'information et en recherche d'informations. La recherche dans Aristote était centrée sur un environnement de conception et de développement d'applications orientées objets, multimédia [Aristote 89, Adiba 92]. Aristote était né du constat suivant : l'approche objet avait « envahi » tous les domaines logiciels (conception, programmation, système, SGBD, IHM) et de nombreuses propositions avaient été faites dans ces différents domaines. Il était donc temps de chercher à unifier ces propositions et surtout à fournir aux concepteurs et développeurs des outils (si possible intégrés) les supportant. Aristote n'était donc pas un projet pour définir le N+1^{ème} modèle ou langage objet, mais plutôt un projet d'intégration et de définition d'un environnement intégré. Une autre dimension d'Aristote était son indépendance vis à vis des SGBD utilisés. C'est pourquoi nous nous sommes attachés à travailler sur la problématique d'interopérabilité relationnel – objet.

Le projet Aristote s'est structuré autour des thèmes suivants :

- génération d'applications, avec quatre grands objectifs : intégrer données et traitements dans un formalisme uniforme (approche langage de programmation pour bases de données), fournir un langage de programmation le plus déclaratif possible (intégrer les manipulations ensemblistes dans le langage par exemple), offrir des capacités de structuration des applications (améliorer la réutilisation en offrant des abstractions supérieures à l'objet ou la classe), pouvoir supporter des applications hétérogènes (c'est à dire intégrer des logiciels écrits dans des environnements différents) ;
- aide à la conception, avec trois grands objectifs : proposer une méthode générale et complète d'analyse et de conception de systèmes d'information couvrant tout le cycle de développement, augmenter la richesse d'expression des modèles objets (sémantique de la composition ou plus généralement des liens inter-objets), supporter les méthodes et langages proposés par des outils graphiques intégrés ;
- applications complexes : étudier des applications complexes (dossiers médicaux, génie logiciel) pour nourrir la réflexion des autres axes en leur proposant des cas concrets.

J'ai eu la chance d'arriver au LGI au moment où le projet Aristote était en gestation et j'ai pu contribuer à sa définition, puis ensuite j'ai assuré la responsabilité de l'axe « Générateur d'applications » en collaboration avec le centre de recherches Bull.

Dans la suite, je présente rapidement le modèle de données Aristote ainsi que les travaux sur l'aspect langage de programmation pour bases de données. Ensuite, je focalise sur l'architecture du générateur Aristote, et finalement je conclus sur les apports du projet.

2.2.1.2 Modèle et langage Aristote

Pour pouvoir travailler, nous avons eu besoin de définir un modèle de données pour Aristote. Les propositions existantes à l'époque (modèle de O₂, standard en gestation de l'O.D.M.G) ne nous ont pas complètement satisfaites. Notamment deux points nous ont semblé un peu faibles : la gestion des extensions et un manque de niveaux d'abstraction (entre application et classe). De plus, le générateur Aristote veut pouvoir s'utiliser aussi bien au-dessus d'un SGBD à objets que d'un SGBD relationnel. C'est pourquoi nous avons essayé d'intégrer dans un modèle objet une gestion plus propre et plus explicite des extensions pour mieux se rapprocher du modèle relationnel. La proposition que nous avons faite [Déchamboux 91] reprend les principes du modèle d'O₂ et y intégrant nos propositions sur ces deux points.

La figure 2-2 résume les concepts du modèle. Au plus bas niveau se trouvent les objets et les valeurs (nous reprenons la séparation faite par O₂). Un objet est typé par son A-type qui définit sa structure (à l'aide de types de base et de constructeurs de type) et son comportement (ensemble de fonctions et de procédures). Les types sont hiérarchisés par une relation d'héritage multiple avec une sémantique classique. Les A-schémas regroupent un ensemble de A-types et permettent de leur associer des A-opérations qui peuvent manipuler des objets de plusieurs A-types différents. Les A-schémas peuvent définir des vues sur les A-types qu'ils manipulent (sous-ensemble de la structure ou du comportement). Les A-schémas peuvent être hiérarchisés par une relation d'héritage simple. Une A-application enfin, définit des A-modules et utilise un ou plusieurs A-schémas en leur associant des extensions de types et en définissant une politique de persistance (persistance explicite via racine de persistance et propagation implicite comme dans O₂). Les A-modules manipulent les extensions à l'aide des opérations des A-schémas.

L'idée de ces niveaux de structuration est d'une part de bien séparer définition intensionnelle (A-type, A-schéma) de la définition extensionnelle (A-application). L'héritage sur les extensions est séparé de l'héritage sur les A-types (un langage ensembliste permet de décrire les relations entre extensions comme inclusion, partition, ...). Cela permet notamment de

mieux isoler les niveaux de réutilisation (une A-application est moins réutilisable qu'un A-schéma qui est moins réutilisable qu'un A-type).

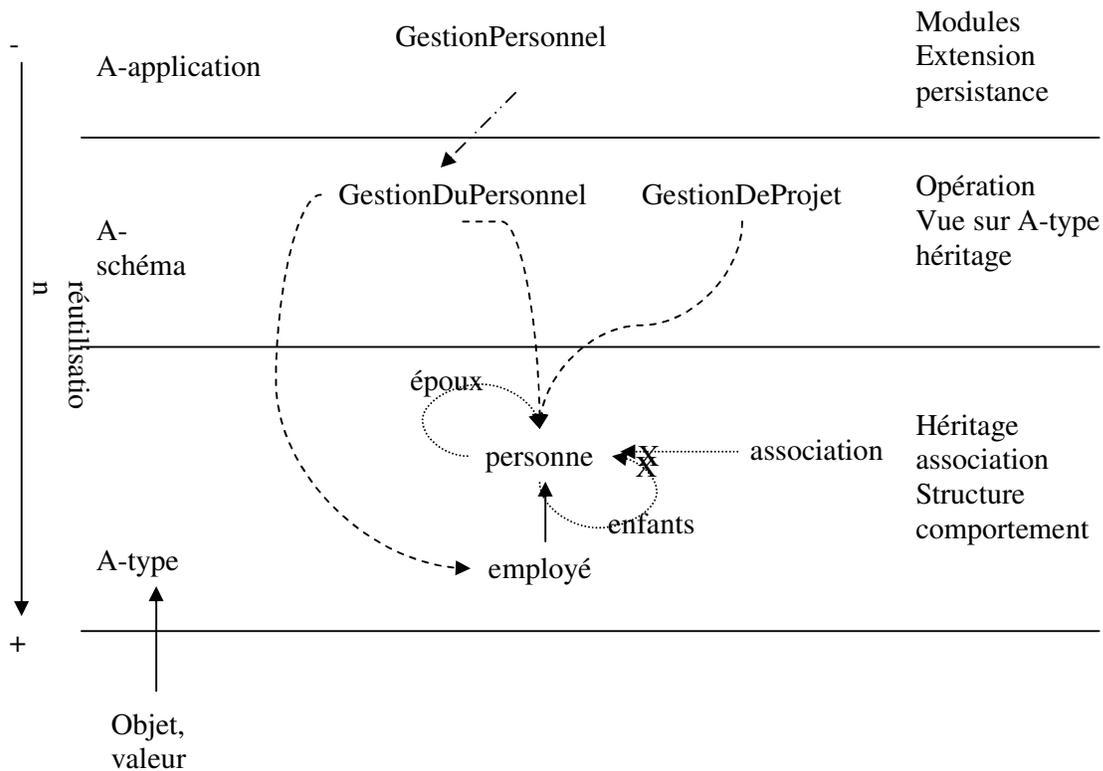


Figure 2-2 : concepts du modèle Aristote

Bien que nous visions un environnement multi-langages, le langage PEPLM (Persistent Programming Language for Object Manipulation) [Déchamboux 93] est le langage privilégié pour le modèle Aristote. Il inclut les concepts proposés dans ce modèle et ajoute des capacités déclaratives notamment des expressions ensemblistes.

Par exemple, ce morceau de code PEPLM, définit une variable *g*, objet de type *Gare*. Cette variable est affectée par le résultat d'une expression ensembliste `select(list-clause-domaine ; condition ; résultat)` qui est ici une simple sélection :

```
ref Gare g ;
g = select(g1 in GaresDeTriage; g1.nomVille=='Marseille'; g1);
```

D'autres opérations sont possibles comme les expressions de chemin, les opérations de restructuration (`nest`, `unnest`), ou des agrégats. Le principal avantage des expressions ensemblistes de PEPLM est leur intégration au langage. Partout où un objet (ou un ensemble d'objets) est attendu, on peut lui substituer une expression ensembliste.

2.2.1.3 Générateur d'applications Aristote

L'approche générateur permet :

- d'être relativement indépendant des systèmes sur lesquels le générateur s'appuie et donc d'être multi-cibles (génération d'applications sur plusieurs systèmes) ;
- de faire interopérer des applications générées sur des systèmes cibles différents ;
- de prototyper plus facilement que s'il faut développer un système complet.

L'inconvénient majeur est qu'elle n'offre pas la maîtrise des couches basses du système (transactions, contrôle de concurrence).

Les systèmes cibles sont les SGBD à objet (comme archétype nous prenons le système O₂), les systèmes relationnels étendus avec types abstraits de données (comme archétype nous prenons le système développé dans le cadre du projet européen EDS et son langage ESQLE [Gardarin 90]). Finalement, nous avons ajouté une génération moins complète avec un SGBD relationnel standard (comme exemple Oracle et son interface de programmation Pro*C).

2.2.1.4 Bilan

Ce projet m'a permis de participer à la définition d'un projet de recherche et à sa mise en œuvre dans un contexte multi-partenaires (un laboratoire universitaire et un laboratoire industriel). La définition du modèle de données et du langage associé PEPLM a fait l'objet de la thèse de doctorat de P. Déchamboux [Déchamboux 93]. Des extensions à PEPLM pour supporter le parallélisme ont été faites par J. Machado dans le cadre de son mémoire de DEA [Machado 92]. Un premier travail sur l'interopérabilité relationnel-objet a été fait par C. Roncancio dans le cadre de son DEA [Roncancio 91]. L'idée était de définir les règles de dérivation du modèle Aristote vers le modèle d'O₂ pour valider notre approche génération. Pour redéfinir le travail dans le vocabulaire plus moderne du modèle MDA, le modèle de données Aristote jouait le rôle de méta-modèle vers le niveau modèle instancié par le modèle d'O₂ et le modèle relationnel étendu. Peplom a été mis en œuvre de manière native, au-dessus de O₂ et également au-dessus du SGBD Oracle par C. Lacombe dans le cadre de son mémoire d'ingénieur CNAM [Lacombe 92].

A ma connaissance peu de travaux se sont intéressés à l'interopérabilité relationnel-objet comme nous l'avons fait. Il y a eu énormément de propositions pour permettre l'accès à des données relationnelles depuis une application objet (par exemple Persistence [Keller 93]) et quelques travaux sur l'accès à des données relationnelles depuis un SGBD à objets (par exemple [Amer Yahia 98]).

Bien que sa vision de l'interopérabilité y soit assez réductrice, ce projet m'a permis d'aborder ce problème que j'ai pu ensuite traiter sous d'autres aspects dans les projets suivants.

2.2.2 Intégration de schéma de SIG à base de logique terminologique

2.2.2.1 Introduction et contexte

Les Systèmes d'Information Géographiques (SIG) permettent l'intégration, la gestion, l'interrogation, l'analyse et la restitution des données géographiques. Ils sont utilisés maintenant à grande échelle et notamment dans toutes les collectivités territoriales, les entreprises de gestion de flux (téléphone, électricité, ...). Ils s'appuient d'une part sur des bases de données spatiales [Guting 94] pour stocker l'information et d'autre part sur un ensemble de transformations permettant de manipuler cette information. Un des problèmes importants à traiter est de fournir des outils permettant d'intégrer des informations provenant de différents SIG. De nombreuses applications nécessitent cette intégration, on peut citer par exemple la gestion de risques naturels où il faut être capable d'intégrer des informations provenant de la météorologie, des directions de l'équipement, de la sécurité civile, ... Ce problème d'intégration n'est pas propre aux SIG, mais l'aspect spatial rend le problème plus difficile (par exemple, on dispose souvent de bases spatiales représentant la même zone géographique mais avec à une échelle différente). Les problématiques recouvrent :

- l'interopérabilité technique des SIG : il faut être capable d'échanger des requêtes et des données entre les différents SIG. C'est un problème qui peut être réglé via la définition d'API génériques et le choix d'une abstraction de communication ;
- l'interopérabilité sémantique des SIG : il faut être capable de manipuler de manière cohérente les informations provenant des différents SIG. Par exemple, reconnaître que tel segment dans tel SIG correspondant à telle route dans un autre SIG.

Dans ce projet nous avons principalement travaillé sur l'interopérabilité sémantique. Nous avons commencé par définir une architecture répartie de type médiation qui suppose la définition d'un modèle de données canonique (voir section 2.2.2.2). Le processus d'intégration proprement dit (voir section 2.2.2.4) nécessite la connaissance des différentes formes de conflits spatiaux (voir section 2.2.2.3).

2.2.2.2 Le modèle GEOCOOPM

Le modèle GEOCOOPM (GEOgraphic COOperative Model) [Branki 97 et 98b] que nous avons défini relève des modèles sémantiques. Il s'inspire des travaux sur le modèle Candide [Sheth 93]. Il possède une sémantique assez riche et a les propriétés des logiques terminologiques communément appelés logiques de description [Brachman 91] (modèles provenant des travaux réalisés en intelligence artificielle dans le domaine des bases de connaissances [Brachman 77] de la famille KL-ONE [Brachman 85]). GEOCOOPM étend les logiques de description en supportant les transformations, plusieurs concepts universels (un concept universel pour chaque type de données) et la dimension spatiale.

Dans notre modèle GEOCOOPM, tout est représenté en termes de concepts et de rôles.

Les concepts permettent de représenter les différents types de données hétérogènes (données numériques, données spatiales, ...). Une donnée est assimilée à une relation unaire. Les rôles permettent quant à eux de modéliser les transformations car elles sont assimilées à des relations n-aires. Les rôles permettent, en outre, de représenter les attributs comme une relation binaire entre concepts. La subsomption et la classification donnent une organisation des données et faciliteront, par la suite, l'intégration de ces données.

Une présentation informelle du langage est la suivante :

Soient C_1, \dots, C_n des concepts, R un rôle (attribut) et k un entier naturel. Plusieurs concepts et rôles sont définis dans ce langage, notamment :

- les concepts universels pour représenter les divers types de données : *Anyspatial-data* pour les données spatiales, *Anynumerical-data* pour les données numériques, *Anymaplayer-data* pour les données cartographiques ;
- les concepts de base tels que : *String*, *Integer*, *Real* et *Geometry* (pour la représentation de la caractéristique spatiale) ;
- les concepts ensembles énumérés *Value-Domain* et sous-ensembles *aset* ;
- le concept $And(C_1, \dots, C_n)$ qui est interprété comme l'intersection des concepts C_1, \dots, C_n
- le concept $All(R, C)$ qui est l'ensemble des objets dont toutes les valeurs de l'attribut R sont dans le concept C ;
- les concepts $Atleast(k, R)$, $Atmost(k, R)$ et $Exactly(k, R)$ qui sont respectivement interprétés comme l'ensemble des objets dont l'attribut R a respectivement au moins, au plus et exactement k valeurs distinctes ;
- le concept $Not(C)$ qui représente l'ensemble des objets qui ne sont pas instances du concept C ;
- le rôle $In-data(C_1, \dots, C_{n-1})$ *And* $Out-data(C_n)$ qui représente l'ensemble des n-uplets (objets) de la relation n-aire qui est définie sur $C_1 \times \dots \times C_{n-1} \times C_n$ et qui permet la définition de transformations.

Ce langage a une syntaxe qui relève de la syntaxe des langages de clauses au sens de la programmation logique. Sa sémantique est complètement inspirée des logiques terminologiques et plus particulièrement du système terminologique Back [Peltason 91].

GEOCOOPM est caractérisé par deux éléments fondamentaux :

- sa syntaxe déclarative permettant de définir des concepts et des rôles en vrac dans la T-box sans souci de leur classification et cela en s'appuyant sur des constructeurs tels

que la conjonction, la négation de concepts, les restrictions de cardinalité du domaine d'arrivée de certains rôles ...

- sa sémantique formelle s'exprimant en termes d'interprétation sur des domaines et permettant d'associer à chaque concept un sous-ensemble du domaine et à chaque rôle une relation n-aire sur ce domaine. Par rapport à Back, nous avons introduit un concept prédéfini Geometry qui dénote des objets spatiaux à deux dimensions (points, lignes ou poly-lignes).

2.2.2.3 Conflits spatiaux

Pour pouvoir intégrer des données, il faut pouvoir détecter leurs similarités et hétérogénéités. Ces hétérogénéités sont appelées conflits lors de l'intégration de bases de données classiques. Divers types de conflits sont recensés : des conflits propres aux bases de données spatiales et des conflits plus fréquents dans les bases de données traditionnelles [Batini 86]. Devogele [Devogele 97] présente une classification des conflits en six classes regroupées selon des critères voisins des critères des bases de données classiques. Nous avons recensé globalement les mêmes conflits. Cependant, nous les classifions, différemment, en quatre catégories ou plus exactement sur quatre niveaux :

- les conflits de **modélisation** ou sous-jacents au modèle : Ce sont tous les conflits qui peuvent apparaître lors de la modélisation des données (différents modèles de représentation : Arc-Info, Smallworld, Geo-concept ...), dans la représentation de la topologie, dans le choix du format (raster ou vectoriel), ...
- les conflits de **schémas** : ce sont tous les conflits qui concernent plus à la structure d'un schéma et aux relations sémantiques entre classes de différents schémas. Ces conflits peuvent être des conflits de noms (deux mêmes classes définies différemment), des conflits de structure (par exemple, deux mêmes concepts peuvent être définis comme une classe dans une base de données et comme attribut dans une autre base), des conflits de fragmentation (une classe d'objets dans une base est représentée par plusieurs classes dans une autre base)...
- les conflits de **méta-données** : ce sont tous les conflits qui peuvent avoir lieu à un niveau plus bas que les conflits précédents. Ce sont des conflits qui sont propres aux SIG et qui découlent essentiellement de la présence de la dimension spatiale. Nous pouvons citer les conflits de méta-données géométriques (deux mêmes classes peuvent avoir différentes résolutions), les conflits de critères (une classe peut représenter les mêmes objets qu'une autre classe mais uniquement ceux ayant une certaine taille), les conflits de représentation de la troisième dimension (2,5D ou 3D), ...
- les conflits de **données** : ce sont des conflits qui surviennent au niveau le plus bas c'est-à-dire au niveau des instances. Ces conflits peuvent être des conflits de saisie (deux mêmes objets de deux classes différentes peuvent avoir des valeurs géométriques différentes dues à des erreurs de saisie), des conflits de limites (deux mêmes objets de deux classes différentes peuvent avoir des valeurs géométriques différentes dues aux caractères imprécis de certaines limites), des conflits de généralisation, ...

2.2.2.4 Le processus d'intégration

2.2.2.4.1 Introduction

Notre processus d'intégration comporte plusieurs étapes (voir figure 2-3) :

- une étape de traduction des différents schémas dans GEOCOOPM (pré-intégration). Cette étape de traduction n'a pas été traitée dans ce travail ;
- une étape manuelle de construction d'une hiérarchie d'attributs globale à partir des schémas traduits (comparaison et conformation de schémas). C'est à l'administrateur de la base ou expert du domaine d'analyser les attributs des différents schémas et d'en extraire les divers liens sémantiques ;
- une phase de construction automatique d'un graphe de concepts (schéma intégré partiel) grâce à la hiérarchie d'attributs, à la fonction de subsomption et aux règles manipulant les relations de la hiérarchie d'attributs (fusion). Le moteur terminologique enrichi de règles va déduire ce graphe automatiquement ;
- une dernière phase semi-automatique de réarrangement de schéma en utilisant les opérateurs de restructuration spatiale et de restructuration de schéma (restructuration). Cette étape est guidée soit par l'administrateur, soit par l'utilisateur final en fonction du but recherché.

Nous présentons ci-après plus en détails les différentes étapes.

2.2.2.4.2 Définition d'un schéma de données dans le modèle canonique

Nous devons traduire dans GEOCOOPM toutes les informations présentes dans un schéma de bases de données spatiales, c'est-à-dire les différents types de données (numériques, spatiales, cartographiques), les transformations ainsi que les méta-données associées.

Nous considérons deux grandes classes de méta-données :

1. les méta-données historiques : ce sont toutes les informations décrivant l'origine ou la source de la donnée spatiale. Ces méta-données peuvent être des informations spécifiant le mode de provenance de la donnée (mode raster, mode vectoriel), ou bien des informations décrivant l'origine de cette donnée (image radar, image satellite, carte papier, relevé de terrain, photographie aérienne) ;
2. les méta-données de description : ce sont toutes les informations donnant une description des éléments permettant de mieux prendre en compte la représentation spatiale des données, les critères de sélection de ces données ... Ces méta-données peuvent être des informations spécifiant la résolution de la donnée, sa précision, son exactitude, son orientation, son référentiel spatial ; des informations mentionnant la dimension de sa géométrie (point, ligne ou polyligne, polygone) ; des éléments décrivant les critères de sélection (exemple : on ne représente que les impasses de plus de 50 m), les critères de décomposition sur des attributs différents (exemple : routes segmentées en fonction du taux de trafic ou du nombre de voies) ou sur attributs identiques (exemple : routes représentées par des tronçons d'au moins 10 m ou d'au moins 100 m) ; ou une spécification de saisie décrivant la troisième dimension.

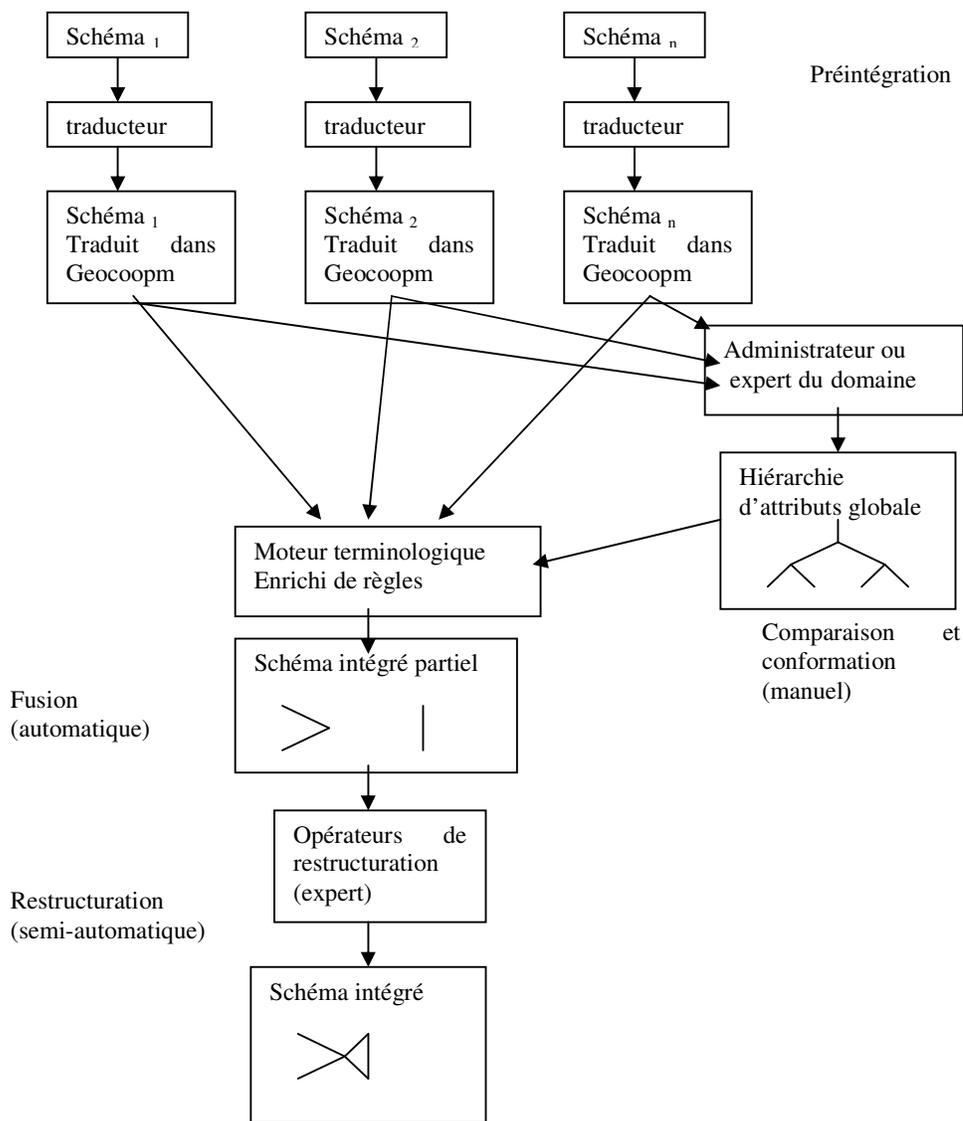


Figure 2-3 : processus d'intégration

Nous prenons en compte toutes ces méta-données. Parmi elles, la seconde classe est directement impliquée lors de l'intégration alors que la première classe permet juste de conserver des informations sur l'historique de la donnée. Toutes ces méta-données seront décrites pour chaque concept spatial. Elles sont décrites dans GEOCOOPM dans une méta-box (boite à part de la T-Box).

2.2.2.4.3 Construction d'une hiérarchie d'attributs globale

Une fois que tous les schémas sont traduits dans GEOCOOPM, une étape d'analyse de schémas est réalisée. Cette phase d'analyse est très importante car elle permet une première identification des éléments communs (comparaison d'attributs) entre les schémas et permet la construction d'un meilleur schéma intégré. Cette étape consiste à la génération d'une hiérarchie d'attributs construite manuellement par l'utilisateur (administrateur de la base ou expert du domaine) et déduite par comparaison d'attributs (rôles).

Différents travaux dans les bases de données classiques [Sheth 93] ont été consacrés à cette étape mais à notre connaissance il y a eu peu de travaux faits pour les bases de données spatiales. Les différentes recherches réalisées ont consisté à exhiber des relations d'équivalence, d'inclusion et de disjonction entre attributs. Les relations d'équivalence sont complètement déduites par le simple fait qu'une correspondance existe entre les domaines de ces attributs (ex : le nom d'une ville et le nom d'un fleuve correspondent). [Sheth 93] présente une construction d'une hiérarchie d'attributs basée sur une équivalence entre attributs plus fine où plusieurs regroupements sémantiques sont définis. Il est clair que l'interprétation sémantique donnée aux attributs est choisie par l'administrateur de la base ou l'expert du domaine en fonction des besoins de l'application.

Il s'agit donc d'identifier, dans un premier temps, les différentes relations entre attributs et de construire, dans un second temps, la hiérarchie. Plusieurs relations sont décrites dans la littérature mais elles concernent des relations spatiales directement entre objets et non entre attributs : relations métriques, topologiques, d'orientation, de composition ... Nous avons sélectionné quatre types de relations qui nous semblent les plus fréquentes et les plus intéressantes au niveau des attributs :

- des relations d'équivalence notées par le symbole \equiv : deux attributs (alphanumériques ou spatiaux) sont équivalents lorsque leurs domaines sémantiques (domaine de valeurs) sont identiques ;
- des relations d'inclusion notées par le symbole \subset : deux attributs sont reliés par une relation d'inclusion lorsque le domaine sémantique de l'un est inclus dans le domaine sémantique de l'autre ;
- des relations de composition notées par le symbole \in : deux attributs sont reliés par une relation de composition lorsque le domaine sémantique de l'un compose spatialement (appartient) le domaine sémantique de l'autre ;
- des relations de disjonction notées par le symbole \neq : deux attributs sont disjoints lorsque leurs deux domaines sémantiques sont disjoints.

La taxonomie des attributs fournie par l'administrateur couplée avec la fonction de subsomption va permettre une intégration plus fine des concepts.

Le moteur terminologique de GEOCOOPM utilise la fonction de subsomption classique des logiques de description enrichie de deux règles s'appuyant sur la hiérarchie d'attributs globale. La première règle permet d'inférer la subsomption de concepts à partir de l'inclusion d'attributs alors que la deuxième permet d'inférer des liens 'is-part-of' entre concepts à partir de la composition d'attributs.

2.2.2.4.4 Génération du graphe des concepts

En fait, c'est l'étape qui va générer un Schéma Intégré Partiel (SIP). Le SIP est automatiquement construit grâce au moteur terminologique. Ce SIP n'est rien d'autre que le graphe des concepts qui a été obtenu après application de la fonction de subsomption et des deux règles précédentes. Le graphe fait apparaître deux relations entre concepts : des relations de composition ('is-part-of') et des relations de subsomption ('is-a').

L'une des richesses des logiques de description est leur langage d'interrogation de la TBox. Ce langage permet de comparer tous les concepts entre eux selon différentes relations (subsomption, équivalence, recouvrement, disjonction et composition). La composition n'existe pas habituellement dans les logiques de description et nous l'avons introduit à partir de la règle générant des liens 'is-part-of' plus la transitivité. Ces comparaisons entre couples de concepts sont utilisées dans l'étape suivante.

2.2.2.4.5 Fusion et restructuration

Nous arrivons à l'ultime étape de l'intégration. Jusqu'à présent, les différents éléments que nous avons définis nous ont permis de faire une intégration des données spatiales qui pallie essentiellement les conflits de schémas (classification, composition, ...). Cependant, les conflits de données (conflits très fréquents dans les bases de données spatiales) dus aux hétérogénéités spatiales (données n'ayant pas le même résolution; ayant des types géométriques différents ...) ne sont pas encore résolus. Ces conflits ne peuvent être traités qu'en prenant en compte les méta-données (type géométrique du concept, résolution du concept, orientation, référentiel spatial ...). Afin de fusionner et de restructurer ces divers schémas, plusieurs opérateurs de restructuration sont nécessaires. Tous ces opérateurs n'existent pas dans les logiques terminologiques mais sont indispensables pour une bonne intégration des données. Ces opérateurs sont des fonctions à déclencher à chaque fois que l'on désire faire une intégration plus complète.

Nous pouvons les classer en deux catégories : les opérateurs de restructuration de schéma (création d'un nouveau concept par généralisation ou par spécialisation de deux concepts) et les opérateurs de restructuration spatiale qui permettent pour un concept donné défini avec certaines méta-données spatiales d'être redéfini avec des méta-données différentes. Nous avons défini des opérations permettant le changement de résolution, d'orientation, de référentiel spatial, de type géométrique, de précision. Bien évidemment ces transformations font passer à une représentation plus grossière. Nous avons également défini des fonctions de filtrage (on ne garde qu'un sous ensemble des instances d'un concept) et de décomposition.

Cette étape de fusion et restructuration est semi-automatique dans le sens où les opérateurs de restructuration existent mais ne sont appliqués qu'en fonction du but recherché par l'utilisateur. Cette étape est donc guidée par l'utilisateur.

Lorsque deux concepts n'ont qu'une seule méta-donnée qui est différente, alors ils vont être généralisés ou spécialisés après application d'une seule fonction de restructuration. Par contre, si les deux concepts sont hétérogènes sur plusieurs méta-données, plusieurs opérateurs de restructuration sont appliqués. Dès lors, une stratégie d'application de ces opérateurs s'impose. Certaines fonctions doivent être réalisées le plus tôt possible et d'autres le plus tard possible. Il est donc tout à fait naturel de n'appliquer les opérateurs qui dégradent la géométrie des objets (en l'occurrence, celles portant sur le type géométrique ou la résolution) qu'à la fin et d'appliquer ceux qui ne la dégradent pas (en l'occurrence, transformation d'orientation ou de référentiel spatial) au tout début. Les opérateurs de restructuration de schéma ne sont utilisés aussi qu'à la fin car ils ne sont mis en jeu qu'une fois que les concepts ont les mêmes méta-données géométriques.

2.2.2.5 Bilan

Notre approche permet de résoudre essentiellement les conflits de schémas et les conflits de méta-données :

- pour les conflits de schémas, les conflits de classification et de résolution sont résolus grâce à la définition de liens 'is-a' dans le graphe des concepts (schéma intégré). Les conflits de fragmentation sont traités par la définition de liens de composition entre concepts. Les conflits de noms et de structures sont facilement résolus par l'administrateur ou l'expert du domaine au moment de la construction de la hiérarchie d'attributs. Les conflits de données/méta-données et les conflits de stockage de l'information ne sont pas traités ;
- pour les conflits sur les méta-données, les conflits de méta-données géométriques, les conflits de dimension géométrique ainsi que les conflits de critères sont résolus grâce au stockage des méta-données, à la définition des liens méta et aux opérateurs de restructuration. Le conflit concernant la troisième dimension n'est pas traité car nous

ne travaillons qu'avec des données spatiales sur un plan. Le conflit de source de données est résolu par le traitement des conflits en aval (résolution ...). Une méta-donnée concernant la source est aussi conservée ;

L'originalité de notre approche apparaît, d'une part, par le fait qu'elle permet de trouver des solutions à plusieurs conflits à la fois (dans la littérature, les conflits sont, en général, résolus séparément vu la complexité du problème sauf dans [Devogele 97]) et d'autre part, par l'approche semi-automatique de son processus d'intégration guidée à la fois par l'administrateur et l'utilisateur (une première étape manuelle, une seconde étape automatique et une dernière étape semi-automatique).

Il est évident que le schéma intégré est construit de telle sorte que toutes les instances d'un même concept du schéma aient des méta-données identiques. Il reste tout de même, dans notre schéma intégré, le fait que deux concepts peuvent ne pas avoir des méta-données identiques.

L'hétérogénéité est donc partiellement résolue. Nous favorisons la multi-représentation des données. Tous les concepts initiaux sont gardés dans leur hétérogénéité alors que tous les nouveaux concepts que nous construisons doivent respecter une certaine homogénéité par rapport aux concepts préexistants.

En fait, ce choix de la multi-représentation a été fait afin d'une part de ne pas surcharger l'administrateur de la base par de multiples opérations de changement de géométrie des objets (les méta-données vont évidemment être modifiées) et, d'autre part, de ne pas dégrader la géométrie de ces objets au moment de l'intégration. Le principe est simple : nous ne dégradons la géométrie des objets d'un concept que si nous avons vraiment besoin d'un nouveau concept résultant de deux concepts hétérogènes.

La résolution de l'hétérogénéité des concepts existants est retardée et est traitée au moment de l'exécution d'une requête. Si une requête nécessite deux concepts hétérogènes, on homogénéise ces concepts au moment de son exécution par application des opérateurs de restructuration spatiale.

Ce travail a fait l'objet de la thèse de T. Branki [Branki 98a] et a été publié dans deux conférences internationales [Branki 97, Branki 98b].

Ce projet nous a permis de travailler sur un sujet assez peu abordé jusqu'alors, à savoir l'intégration de schémas de SIG hétérogènes. L'approche suivie a été de s'inspirer des travaux réalisés dans le cadre des bases de données classiques utilisant les logiques de description pour raisonner les schémas et de les adapter au contexte SIG. Le résultat est assez probant car une bonne partie des conflits spatiaux est ainsi traitée. Il n'en reste pas moins que ce résultat est obtenu en augmentant la puissance d'expression de la logique terminologique et donc en perdant peut être des propriétés calculatoires (nous n'avons pas travaillé sur ce point). Une partie également du processus reste manuel car l'étape importante de la définition de la hiérarchie d'attributs globale doit être faite par l'administrateur. Enfin, notre processus ne prend pas en compte les conflits d'instances et notamment tout le problème d'appariement de données spatiales (comment « recoller » deux objets spatiaux « voisins »). [Laurini 98] s'intéresse à ce problème et décrit de manière détaillée les hétérogénéités qui peuvent survenir entre données spatiales. Il présente plusieurs problèmes d'erreurs de mesures survenant dans les bases de données géographiques distribuées que ce soit lors de la fragmentation zonale ou dans la fragmentation des thèmes. Il propose une méthodologie d'intégration de fragments de zones voisines (conflits de limites, conflits de méta-données ...) en utilisant des tables d'adjacences et un algorithme de lien géométrique entre les zones à fusionner.

La définition d'un modèle de données canonique est traitée au niveau des standards comme OpenGIS [OpenGIS 04] ou par exemple par le modèle MADS [Spaccapietra 99]. Par rapport

à GEOCOOPM ces modèles ne mettent pas l'accent sur l'intégration et n'offrent pas de mécanisme classificateur puissant comme la subsomption.

En ce qui concerne la gestion des conflits au niveau schéma, [Devogele 98] propose de définir des correspondances entre schémas au moyen de règles ICA (Inter-schema Correspondence Assertions). Une règle ICA définit une correspondance entre deux éléments de schémas (exprimée par une opération ensembliste classique) en spécifiant également les instances associées (au moyen d'un prédicat de sélection) et la façon de passer d'une géométrie à une autre. L'intérêt de cette approche est qu'elle permet de décrire de manière homogène des informations sur les schémas, les instances et la géométrie. Par contre cela peut conduire à devoir multiplier le nombre de règles. Notre approche offre l'avantage de s'intéresser à une construction plus automatique du schéma global (la subsomption va être utilisée pour « apprendre » des correspondances). [Boucelma 02] reprend cette idée de règles ICA dans le contexte d'une architecture de médiation basée sur les standards OpenGIS. Les règles ICA sont ici utilisées lors de l'évaluation d'une requête globale pour réécrire la requête sur les sources. Notre approche est plutôt complémentaire puisque nous nous intéressons à la construction du schéma global, alors qu'ils s'intéressent plutôt à l'évaluation efficace de requêtes. [Cruz 02] utilise également une notion de correspondance (appelée *agreement*) décrite sous forme de *mapping* exprimé par un document XML. Le *mapping* est exprimé entre le schéma d'une source de données et le schéma global (exprimé sous forme d'ontologie). L'intégration est donc supposée résolue et ce travail ne s'intéresse qu'à la réécriture de requêtes.

[Leclercq 99] propose une architecture multi-agent pour construire le schéma global en interagissant entre les différentes sources. Le modèle canonique proposé est une extension spatiale du modèle ODMG. La sémantique des schémas est exprimée dans des contextes et des transformations permettent de passer d'un contexte à un autre. L'idée est assez proche des règles ICA, la principale nouveauté étant l'utilisation d'une architecture multi-agent.

Globalement la recherche sur l'interopérabilité des SIG suit celle sur les bases de données en général, avec le passage des architectures fédérées aux architectures de médiation. Les modèles de données suivent le même chemin avec l'utilisation d'ontologies à la place des schémas [Fonseca 02].

3 Vers l'interopérabilité sémantique

3.1 Introduction

Le projet GEOCOOPM a été un premier pas pour la prise en compte d'information sémantique dans le processus d'intégration. L'information sémantique y est décrite dans la hiérarchie d'attributs et les mécanismes de raisonnement de la logique terminologique utilisent cette hiérarchie lors du processus d'intégration de schémas. La sémantique n'est cependant pas centrale dans GEOCOOPM et nous avons essayé d'aller plus loin dans les projets qui ont suivis. Je présente dans cette partie deux approches sur lesquelles nous travaillons aujourd'hui. Tout d'abord nous utilisons la sémantique pour améliorer la description de ressources pédagogiques disponibles sur l'Internet et pour définir de la composition et de l'adaptation de ressources. La sémantique est ici décrite avec une approche web sémantique (ontologie de concepts, méta-données, « inférence ») et le système réalisé est d'ailleurs mis en œuvre avec des outils basés sur le formalisme RDF du W3C.

Le deuxième projet travaille sur la prise en compte de la sémantique dans les systèmes P2P. L'idée ici est d'améliorer le processus de recherche en organisant le réseau selon des critères sémantiques. Nous détaillons successivement nos contributions sur ces deux problématiques.

3.2 Modèle de description sémantique de ressources éducatives

3.2.1 Introduction et contexte

L'enseignement à distance a connu un grand développement ces dernières années et a donné une grande ampleur à une activité de recherche existante en informatique qui a d'abord travaillé sur l'EAO (Enseignement Assisté par Ordinateur), puis l'EIAO (Enseignement Intelligemment Assisté par Ordinateur) et ce qui est appelé souvent les « tuteurs intelligents » et enfin aujourd'hui les EIAH (Environnement Informatique pour les Apprentissages Humains). Le travail dans le domaine des EIAH est nécessairement pluridisciplinaire entre les informaticiens, les pédagogues, les spécialistes d'interaction, les didacticiens, ... Les travaux sur l'enseignement à distance peuvent se situer comme un courant plutôt informatique des EAIH centré sur les problèmes de production, de diffusion, de suivi, de collaboration et d'évaluation de contenus de nature pédagogiques dans un environnement informatique distribué (généralement le web). Ce domaine évacue beaucoup des problèmes fondamentaux des EAIH (interactions, modèles pédagogiques et didactiques) pour se centrer sur la seule dimension « transmission d'informations ». Dans ce contexte, le problème de l'enseignement à distance est très proche de celui de la construction de documents hypermédia adaptables et la spécificité apprentissage humain reste faible.

Ces fonctions sont généralement assurées par ce que l'on appelle un LMS (Learning Management System) ou plate-forme d'enseignement à distance. De multiples LMS ont déjà été développés (le site thot en recense plusieurs centaines) et supportent plus ou moins les fonctionnalités attendues. Cette floraison d'outils pose bien sûr des problèmes d'interopérabilité que les travaux de standardisation vont partiellement régler. Il existe notamment une proposition de standard appelée LOM (Learning Object Metadata) promue conjointement par l'IEEE et les organismes européens.

LOM définit un langage de méta-données pour décrire les objets pédagogiques. Cependant, ce standard définit un ensemble de méta-données très complexe et très long à décrire (neuf catégories et plusieurs dizaines d'attributs) et il ne va pas être facile de motiver les auteurs pour qu'ils décrivent leurs productions avec un tel langage. De plus, il est plutôt incomplet sur la partie description sémantique (ce que font les ressources) ainsi que sur les dépendances entre ressources (il propose une spécification a priori des dépendances, ce qui n'est pas valide

dans le cadre de ressources distribuées et présentes en grand nombre). En ce qui concerne la construction de ressources complexes, un autre standard semble se dégager, il s'agit de SCORM [ADLI 01] qui propose un langage de structuration de composants.

Le bilan de ces efforts de standardisation reste encore modeste. Si on peut penser que ces standards vont jouer un rôle intéressant dans le cadre d'échanges de ressources entre plateformes différentes de type LMS, leur manque de richesse descriptive va fortement les limiter dans la prise en compte des fonctions importantes que sont la réutilisation et l'adaptation aux apprenants. Pour aller plus loin, on peut même dire que ces standards vont surtout « rassurer » les utilisateurs de LMS parce qu'ils vont y voir un moyen de pérenniser leurs investissements.

D'autres outils sont également apparus que l'on peut définir comme des médiateurs (ou *brokers*) de ressources. Il s'agit de répertoires plus ou moins organisés de ressources, décrites à l'aide d'un jeu de méta-données (dans certains cas LOM) et classifiées via des hiérarchies plus ou moins élaborées. Ces médiateurs offrent deux fonctions principales, le dépôt et la recherche de composants. Par contre, ils offrent peu ou pas d'outils de réutilisation et de composition. Les organisations Ariadne (www.ariadne.org et [Ariadne 02]) ou educanext (www.educanext.org), toutes les deux créées à la suite de projets européens, offrent des outils de ce type.

Nous nous situons dans ce courant et nos travaux peuvent se définir comme des modèles et outils d'indexation, de recherche et d'adaptation de ressources basés sur des descriptions sémantiques. L'éducatif peut être vu comme un exemple d'application possible de notre démarche, même si le modèle de description choisi est lié au domaine éducatif. Nous donnerons dans la partie prospective de ce document les idées que nous avons pour généraliser les résultats obtenus.

Dans la suite de cette partie, nous présentons successivement l'architecture logique de notre système, puis nous détaillons le modèle de description des ressources et enfin nous terminons par la présentation des algorithmes de sélection et d'adaptation proposés.

3.2.2 Architecture logique

Notre système s'adresse à deux catégories d'utilisateurs les auteurs de ressources et les apprenants. Les fonctions proposées sont bien évidemment différentes :

- pour les auteurs : ajout, recherche d'une ressource, ajout d'une ressource par composition de ressources existantes ;
- pour les apprenants : différents modes pédagogiques sont proposés : mode cours (l'apprenant choisit un cours qui est ensuite « adapté » en fonction de son profil et de ses connaissances), mode concept (l'apprenant choisit un ou plusieurs concepts et il faut alors retrouver le cours correspondant le mieux à sa recherche et « l'adapter » ensuite), mode requête (l'objectif de l'apprenant est défini par une requête ne portant pas sur des concepts).

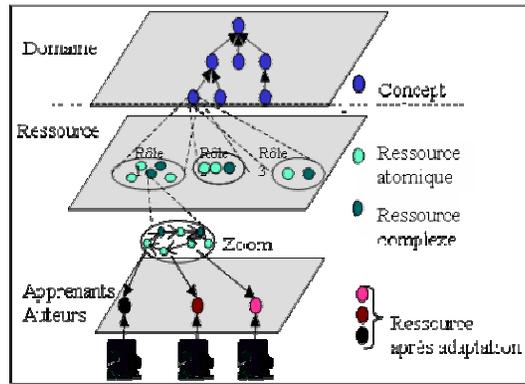


Figure 3-1 : les trois modèles de description

Pour réaliser ces fonctions, le système doit s'appuyer sur un certain nombre de modèles (voir figure 3-1) :

- modèle de domaine : il s'agit de décrire l'ensemble des concepts couverts par le système. Ces concepts sont décrits par un graphe où les nœuds sont des concepts et les arcs des relations sémantiques entre concepts (voir un exemple figure 3-2). Nous proposons d'utiliser la relation « est plus général » et son inverse « est plus spécifique » ainsi qu'un certain nombre de relations rhétoriques (« est antithèse de », « contraste avec », ...) extraites de [Mann 87]. Ce graphe peut être vu comme une ontologie simple du domaine couvert par les ressources. Ce graphe est construit à priori par l'administrateur du système. Il est évident que cette tâche n'est pas facile si le domaine couvert est grand. Il existe cependant déjà de telles ontologies (pas forcément avec des relations sémantiques complexes entre concepts) pour certains domaines (par exemple celle de l'ACM pour la définition des cursus en informatique). Ce modèle de domaine va nous servir de référentiel pour « indexer » sémantiquement, tant les apprenants que les ressources.
- modèle de l'apprenant : nous proposons de décrire un apprenant sous deux facettes. La première, que nous appelons ses préférences, décrit des informations factuelles (nom, adresse email, langues préférées, couleurs préférées, ...) et se modélise sous forme d'un ensemble de couples attribut-valeur. La deuxième, que nous appelons ses connaissances, décrit les concepts qu'il connaît qualifiés par un (des) rôle(s) (« introduction », « définition », « description », ...) et une pondération (évaluation du niveau de l'apprenant pour ce concept-rôle). Cette facette se modélise sous forme de relations vers le modèle de domaine. Le contenu de cette facette va évoluer de manière dynamique et automatique au fur et à mesure que l'apprenant va suivre des cours et donc acquérir de nouvelles connaissances.

Par exemple, l'apprenant U_{14} est décrit de la manière suivante :

$UM(U_{14}) = \langle U_{14}, \{ \langle \text{language}, \text{"français"} \rangle, \langle \text{media}, \text{"video"} \rangle, \langle \text{bgcolor}, \text{"white"} \rangle \}, \{ \langle \text{"relational algebra"}, \text{"definition"}, \text{"medium"} \rangle, \langle \text{"SQL"}, \text{"description"}, \text{"low"} \rangle, \langle \text{"SIG"}, \text{"-"}, \text{"very low"} \rangle \}$. Nous avons choisi de discrétiser les valeurs prises par la pondération du niveau de l'apprenant dans un concept-rôle. Cela nous semble plus facile à comprendre par les apprenants et suffisamment précis pour les manipulations que nous avons à faire.

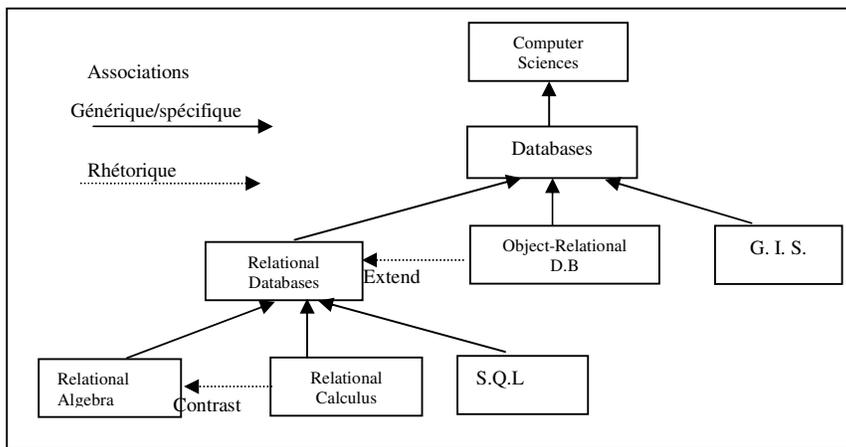


Figure 3-2 : exemple de modèle de domaine

- modèle de description de ressources : chaque ressource intégrée dans le système doit être décrite de manière complète pour que les auteurs et les apprenants puissent les retrouver pour les réutiliser ou les suivre. Le modèle de description de ressources est décrit en détails dans la section suivante.

3.2.3 Modèle de description des ressources

Pour pouvoir être retrouvée, réutilisée (seule ou par composition) une ressource pédagogique doit être décrite par un ensemble de méta-données (voir figure 3-3). Ces méta-données peuvent être classifiées en deux grandes catégories. La première décrit les caractéristiques générales de la ressource (auteurs, titre, langue, média, durée, ...) et se modélise par un ensemble de couples attribut – valeur. La deuxième décrit la sémantique de la ressource relativement au modèle de domaine. Cette sémantique est elle-même structurée en trois parties : les pré-requis (l'entrée de la ressource), le contenu et la fonction d'acquisition (ses sorties). Cette description est analogue à ce qui peut être utilisé pour décrire des programmes (pré-condition, invariant et post-condition).

Les pré-requis d'une ressource sont un ensemble de triplets $\langle \text{concept}, \text{rôle}, \text{niveau} \rangle$ analogues au modèle de l'apprenant. Le contenu est décrit par un ensemble de couples $\langle \text{concept}, \text{rôle} \rangle$ et enfin la fonction d'acquisition indique quels triplets $\langle \text{concept}, \text{rôle}, \text{niveau} \rangle$ vont être ajoutés au modèle de l'apprenant si une condition de validation est vérifiée. Ces triplets correspondent forcément à des concept-rôles présents dans le contenu de la ressource. La condition de validation peut être très différente selon la nature des ressources. Par exemple, pour une ressource statique (un site web par exemple) cela peut être simplement une évaluation du temps passé sur la ressource ou le pourcentage de pages visitées. Par contre pour une ressource dynamique, un résultat plus précis peut éventuellement être obtenu (par exemple le nombre de bonnes réponses à un QCM). Dans ce dernier cas, cela suppose également que l'on puisse accéder à ce résultat (cela dépend de l'interface offerte par la ressource et plus généralement de l'architecture de répartition choisie).

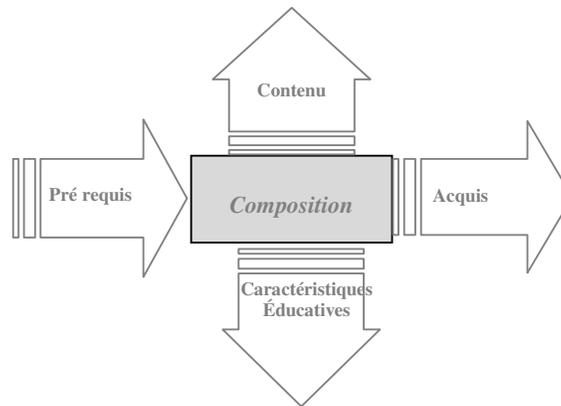


Figure 3-3 : modèle de description d'une ressource

La description sémantique des ressources va nous permettre de définir un certain nombre de propriétés intéressantes :

- satisfaction : un apprenant A satisfait une ressource R lorsque son modèle inclut les pré-requis de R ;
- maîtrise : un apprenant A maîtrise une ressource R lorsque son modèle inclut le contenu de R ;
- substitution : une ressource R_1 est substituable à une ressource R_2 lorsque les pré-requis de R_1 sont égaux aux pré-requis de R_2 (propriété commutative) ;
- équivalence de ressources : une ressource R_1 est équivalent à une ressource R_2 lorsque R_1 est substituable à R_2 et que le contenu de R_1 est égal au contenu de R_2 ;
- précedence faible : une ressource R_1 précède faiblement une ressource R_2 , si le contenu de R_1 est inclus dans les pré-requis de R_2 (symétriquement R_2 succède faiblement à R_1) ;
- précedence forte : une ressource R_1 précède fortement une ressource R_2 , si le contenu de R_1 est égal aux pré-requis de R_2 (symétriquement R_2 succède fortement à R_1).

Pour illustrer notre propos, prenons R_{10} ressource décrite par les caractéristiques générales suivantes :

$\{R_{10}, \{<language, \{“français”\}>, <media, \{“text”, “video”\}>, <author, \{“defude”, “duitama”\}>\}$

Le contenu CT de R_{10} est $CT = \{<R_{10}, “SQL”, \{“description”, “application”\}>\}$, c'est à dire R_{10} décrit et applique le concept *SQL*. Les pré-requis P, $P = \{<R_{10} “SQL”, “description”, “low”>, <R_{10} “relational algebra”, “definition”, “low”>\}$ définissent que R_{10} suppose que ses apprenants ont le niveau *low* pour le concept *SQL* et le rôle *description* ainsi que le concept *relational algebra* et le rôle *definition*. Si la ressource R_{10} est validée, le couple (“*SQL*”, “*application*”), qui appartient à CT, est ajouté au modèle de l'apprenant avec le niveau *medium*; i.e. <“*SQL*”, “*application*”, “*medium*”>.

Une ressource peut être atomique ou complexe. Une ressource complexe est construite par l'application (éventuellement récursive) d'opérateurs de composition sur des ressources (atomiques ou complexes). Ceci définit un graphe de composition. Ce graphe a obligatoirement une source (un nœud ressource) mais peut avoir plusieurs sorties. Nous avons choisi cinq opérateurs, trois opérateurs simples (SEQ pour la séquence, PAR pour le parallélisme, ALT pour l'alternative) et deux opérateurs plus complexes (AGG pour l'agrégation de deux ressources complexes et PROJ pour définir une ressource par projection d'une autre). L'opérateur AGG permet de composer une ressource complexe avec une autre

ressource (comme le graphe peut avoir plusieurs sorties, il faut spécifier quelle sortie doit être « raccrochée » à quelle entrée). L'opérateur PROJ permet de réutiliser une ressource complexe en la simplifiant.

Par exemple, R_{10} est une ressource complexe (voir son graphe de composition figure 3-4). Son graphe de composition est défini par $R_{10} = R_1 \text{ SEQ } (R_5 \text{ ALT}(R_2 \text{ SEQ } (R_3 \text{ PAR } R_4)))$

Les opérateurs de composition ont une sémantique bien définie qui permet de dériver automatiquement la sémantique d'une ressource complexe à partir de la sémantique de ses ressources composantes. Le tableau ci-dessous résume cette sémantique.

Expression	Prérequis	Contenu
$R_i \text{ SEQ } R_j$	$P(R_i) \wedge P(R_j) \wedge \neg C(R_i)$	$C(R_i) \wedge C(R_j)$
$R_i \text{ ALT } R_j$	$P(R_i) \vee P(R_j)$	$C(R_i) \wedge C(R_j)$
$R_i \text{ PAR } R_j$	$P(R_i) \wedge P(R_j)$	$C(R_i) \wedge C(R_j)$

Des problèmes de cohérence sémantique peuvent également être détectés sur le graphe de composition :

quelque soit (R_i, R_j) deux ressources du graphe telles qu'il existe (au moins) un chemin dans le graphe qui mène de R_i à R_j : $\text{Max}_{i, j} [\text{cardinal}(C(R_j) \cap P(R_i)) / \text{cardinal}(P(R_i))] = k$

k est compris entre 0 et 1. $k = 0$ indique que le graphe est fortement cohérent, alors que $k = 1$ indique que le graphe est fortement incohérent.

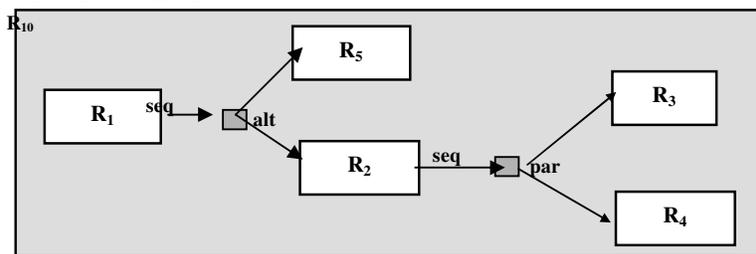


Figure 3-4 : exemple de graphe de composition

Lorsque l'on veut composer des ressources complexes, il faut pouvoir décrire précisément comment les deux graphes se fusionnent en un seul. Par exemple, si on considère le graphe de la ressource R_{10} et celui de la ressource R_{20} ($R_{20} = R_{12} \text{ SEQ } R_{13}$), l'expression $R_{30} = R_{10} \text{ SEQ } R_{20}$ définit le graphe de la figure 3-5. La fusion des deux graphes se fait en ajoutant un arc SEQ entre toutes les sorties de R_{10} et l'entrée de R_{20} .

Si on veut une autre sémantique, il faut utiliser l'opérateur AGG qui donne plus de flexibilité. Par exemple, l'expression $R_{10} \text{ AGG } R_{20}$ ($R_{10}.R_4 \text{ SEQ } R_{20}.R_{12}$) spécifie la fusion en ajoutant un opérateur-arc SEQ entre la sortie R_4 de R_{10} et l'entrée de R_{20} . L'ajout d'arc se fait forcément sur des nœuds feuilles du premier graphe (voir figure 3-6).

il y a toujours propagation de l'échec à la ressource englobante. La figure 3-7 décrit graphiquement un graphe de composition avec prise en compte des échecs.

3.2.4 Sélection et Adaptation

Le modèle de description de ressources choisi va nous permettre d'une part d'offrir des modes de sélection sophistiqués pour les ressources tant pour les apprenants que les auteurs et d'autre part d'adapter les ressources sélectionnées en fonction de l'apprenant. Les modes de sélection destinés aux apprenants vont correspondre à des modèles pédagogiques différents :

- mode ressource : l'apprenant sélectionne une ressource particulière (par navigation dans la base des ressources ou bien parce qu'un enseignant lui a fourni cette référence). Il s'agit du mode le plus classique qui correspond à la pratique habituelle d'enseignement où l'apprenant est passif par rapport au choix du contenu ;
- mode concept : l'apprenant définit le (ou les) concept(s) sur le(s)quel(s) il/elle veut travailler et ensuite il va y avoir un processus de sélection des ressources correspondantes. Il s'agit plutôt d'un mode actif ;
- mode requête : il s'agit du même principe que pour le mode concept, mais la requête porte sur la partie non sémantique de la description de ressources.

De manière plus générale, le processus peut se découper de la manière suivante :

- *sélectionne(requête R)* : renvoie les identificateurs de ressources correspondant à R ;
- *adapte(ensemble d'identificateurs de ressources I, apprenant A)* : choisit une ressource r dans I satisfaisant au mieux le modèle de A ;
- *présente(ressource r, apprenant A)* : construit la présentation de la ressource r pour l'apprenant A.

Ce processus s'enchaîne de manière séquentielle avec un bouclage possible de *adapte* vers *sélectionne* (voir ci-après).

L'apprenant va pouvoir interagir dans ce processus de deux manières. Tout d'abord, *sélectionne* et *adapte* sont définis en partie sous forme de règles. L'ensemble des règles par défaut peut être surchargé par l'apprenant pour mieux adapter le processus. Ensuite, l'apprenant peut choisir la stratégie qui lui convient le mieux pour résoudre sa requête (privilégier la précision, la vitesse, ...).

Nous allons maintenant préciser les différents modes pédagogiques.

3.2.4.1 Mode ressource

Dans ce mode, le processus *sélectionne* est vide puisque l'apprenant choisit lui-même une ressource. Le processus *adapte* se décompose comme suit (voir figure 3-8) :

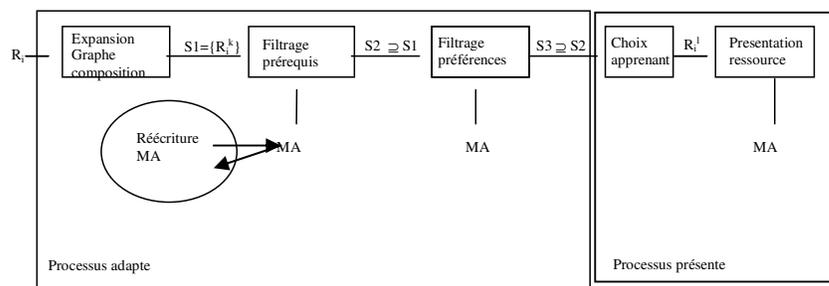


Figure 3-8 : adaptation en mode cours

- l'apprenant choisit une ressource R_i ;
- le système construit l'ensemble S_1 des graphes affichables de R_i . Un graphe affichable est un graphe sans opérateur ALT. Par exemple, le graphe de composition de $R_{10} = R_1$

SEQ (R₅ ALT(R₂ SEQ (R₃ PAR R₄))) peut s'expanser en {R₁ SEQ R₅, R₁ SEQ R₂ SEQ (R₃ PAR R₄)}. Bien entendu cette expansion ne produit plusieurs éléments que si le composant est complexe et si son graphe contient des ALT ;

- S₁ est ensuite filtré par les pré-requis, c'est à dire en calculant si le modèle de l'apprenant satisfait ou non les pré-requis de chaque graphe affichable. Cela construit S₂, sous-ensemble de S₁, ensemble des graphes affichables. Si S₂ est vide, deux stratégies sont possibles : le traitement s'arrête par un échec ou bien le processus de satisfaction est affaibli en appliquant des règles qui vont utiliser les associations entre concepts du modèle de domaine (voir partie sur les règles) ;
- S₂ est ensuite filtré selon les préférences de l'apprenant et cela construit S₃, sous-ensemble de S₂ ;
- Si S₃ est réduit à un seul graphe, il est présenté, s'il y en a plusieurs, l'apprenant est invité à en choisir un seul.

3.2.4.2 Mode concept

Dans ce mode, l'apprenant sélectionne des ressources en définissant une requête portant sur des concepts. Le processus est différent selon que la requête porte sur un concept ou sur plusieurs concepts. En effet, lorsqu'il y a plusieurs concepts, il se peut qu'aucune ressource ne corresponde à la requête mais que plusieurs ressources correspondent partiellement à celle-ci. En les composant, on peut alors construire dynamiquement une nouvelle ressource correspondant à la requête.

Une requête se définit de la manière suivante :

$$Q = (c_{1,1}, r_{1,1} \vee \dots \vee c_{1,k}, r_{1,k}) \wedge \dots \wedge (c_{n,1}, r_{n,1} \vee \dots \vee c_{n,l}, r_{n,l})$$

où c représente un concept, r un rôle, avec k, n et l ≥ 1

Lorsque n = 1, il s'agit du mode mono-concept et le processus s'effectue comme suit (voir figure 3-9) :

- la requête Q_i est résolue sur la base des ressources. La résolution peut se faire en deux modes : le mode exact (une ressource est sélectionnée si son contenu est égal à la requête) et le mode inclusif (une ressource est sélectionnée si son contenu contient la requête). Si le résultat est vide, la requête est réécrite, via des règles, pour essayer de produire une nouvelle requête fournissant un résultat non vide ;
- lorsque le résultat est non vide, on est ramené au mode ressource, c'est à dire que l'ensemble des ressources retournées est adapté ;
- une phase d'évaluation des résultats est ajoutée, exprimée par des règles, et qui porte essentiellement sur la taille de l'ensemble. Le résultat peut être soit trop petit, soit trop grand. Le cas échéant, la requête va être réécrite, via des règles, pour pouvoir obtenir un nouveau résultat plus conforme aux attentes.

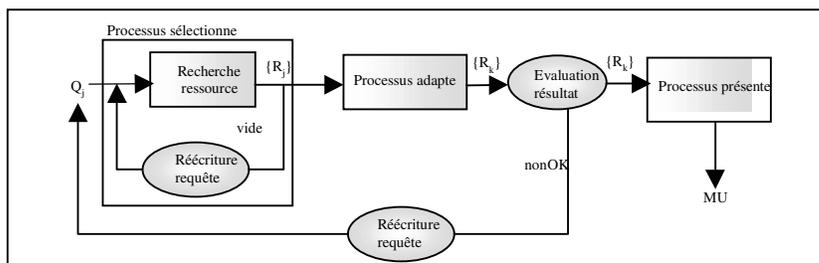


Figure 3-9 : adaptation en mode mono-concept

Lorsque n>1, il s'agit du mode multi-concept et le processus est changé comme suit (voir figure 3-10) :

- la requête est résolue sur la base des ressources. Si le résultat est non vide, le processus est identique au mode mono-concept ;
- si le résultat est vide, la requête doit être transformée. Le nombre de ressources à recomposer pour satisfaire la requête doit être minimisé. La requête est décomposée en n sous-requêtes mono-concepts qui sont ensuite résolues en mode inclusif sur la base. On calcule ensuite l'intersection entre le contenu de chaque ressource retournée et la requête et l'ensemble des ressources est trié selon la taille. Le choix des ressources peut se faire soit en cherchant à maximiser la taille des ressources sélectionnées, soit à minimiser le nombre de ressources;
- pour les requêtes transformées, il faut composer les résultats obtenus par les sous-requêtes pour satisfaire la requête initiale. Ce travail de composition va être fait en appliquant l'opérateur PAR entre les différentes ressources sélectionnées par la phase précédente. Les propriétés de précedence vont être utilisées pour calculer un pré-ordre entre les ressources et ainsi transformer des PAR en SEQ. Les ressources ainsi construites dynamiquement peuvent ensuite être sauvegardées pour proposition aux auteurs. Ceux-ci peuvent alors décider de les transformer en nouvelles ressources ou les supprimer ;
- le reste du processus est analogue au cas mono-concept.

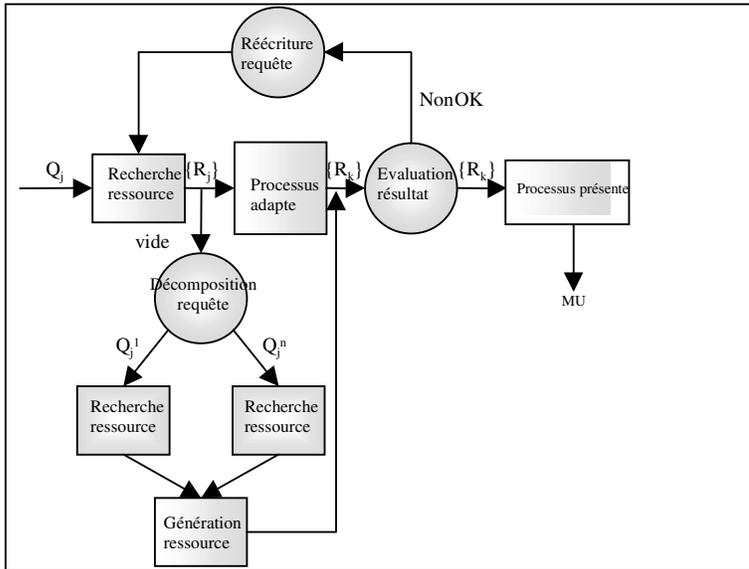


Figure 3-10 : adaptation en mode multi-concept

Par exemple, supposons que nous ayons à résoudre la requête suivante :

$Q_2 = (\langle \text{relational algebra} \rangle, \{ \text{definition} \}) \wedge (\langle \text{object-relational DB} \rangle, \{ \text{introduction} \})$

Supposons que le résultat de l'évaluation de Q_2 soit vide. On décompose alors Q_2 en $(Q_{21} = \langle \text{relational algebra} \rangle, \{ \text{definition} \}, Q_{22} = \langle \text{object-relational DB} \rangle, \{ \text{introduction} \})$

La résolution de Q_{21} donne $S_1 = \{ R_{10} \}$ et celle de Q_{22} donne $S_2 = \{ R_{23}, R_{45} \}$

La ressource générée dynamiquement est $R_{10} \text{ PAR } (R_{23} \text{ ALT } R_{45})$ car nous supposons ici qu'il n'y a pas d'ordre entre les ressources de S_1 et celles de S_2 .

3.2.4.3 Mode requête

Ce mode correspond au mode mono-concept, la seule différence est que la requête ne porte pas sur des concepts mais sur des méta-données classiques. Si la résolution de la requête n'est pas vide, le processus est celui du mode mono-requête. S'il n'y a pas de réponses, l'apprenant doit reformuler une nouvelle requête.

3.2.4.4 Règles d'adaptation

L'adaptation du système à l'apprenant se fait essentiellement par le biais de règles de type CA (Condition – Action) qui vont servir à réécrire le modèle de l'apprenant, réécrire une requête ou évaluer le résultat. De plus, un apprenant peut choisir de résoudre une requête en deux modes. Le mode « précision » où le système va minimiser la « dégradation » de la requête et le mode « approché » où le système peut appliquer des dégradations plus importantes.

Les règles sont utilisées :

- pour réécrire une requête en mode concept (cas où la requête retourne un ensemble vide de ressources) ou en mode ressource (cas où la ressource sélectionnée ne peut être adaptée pour l'apprenant);
- pour évaluer si le résultat construit en mode concept est conforme aux attentes de l'apprenant ;
- pour réécrire la requête lorsque le résultat n'est pas conforme aux attentes de l'apprenant.

Nous avons choisi une approche à base de règles car elle permet de simplifier l'entrée de nouveaux paramètres dans le système. Cette approche est utilisée dans le domaine des optimiseurs de requêtes par exemple, ou bien dans les hypermédia adaptatifs. Leur inconvénient majeur est le modèle d'exécution sous-jacent qui peut engendrer des problèmes de non-déterminisme ou de terminaison. Nous avons délibérément choisi un modèle d'exécution simple pour éviter ces problèmes.

Nous proposons de décrire le comportement du système par un ensemble de règles par défaut. A défaut d'autres entrées, ce sont ces règles qui s'appliquent. Les utilisateurs du système, l'administrateur ou même chaque apprenant, peuvent ajouter de nouvelles règles pour affiner le comportement du système. Se pose alors le problème du conflit entre ces différentes règles. Pour cela, nous avons défini une priorité entre les règles basée sur une estimation de la « précision » de la règle. Nous considérons qu'une règle est précise lorsque sa condition porte sur un grand nombre de conditions élémentaires, lorsqu'elle utilise des variables et non des constantes et lorsque les concepts manipulés sont spécifiques dans le modèle de domaine. Ceci est évalué par un entier compris entre 0 et 1 (0 règle imprécise et 1 règle très précise).

Nous illustrons le langage de règles par quelques exemples.

Règle 1.

```
FOR REWRITING QUERY
IF q-contain(:uq, :concept)
  AND equal(learner-type(:u1), "occasionnel")
  AND hierarchical-level (:concept) between 1 AND 3
USE
  ADD_RELATION("contrast", :uq);
  MODIFY_ROLES (
    "description" INSTEAD OF "definition";
    "example" INSTEAD OF "introduction", :uq );
  ADD_RELATION("narrower", :uq);
END;
```

Cette règle est utilisée pour réécrire une requête (clause FOR) lorsqu'une ressource sélectionnée ne peut être adaptée à un apprenant car aucun graphe d'affichage n'est satisfait par le modèle de l'apprenant. La condition filtre les apprenants « occasionnels » du système (estimés par une métrique) et le fait que la requête utilise au moins un concept de niveau hiérarchique compris entre 1 et 3 (le niveau hiérarchique d'un concept est la longueur du chemin qui mène de la racine du modèle de domaine au concept en utilisant les arcs de type généralise/spécialise ». Lorsque la condition est vérifiée, la règle propose trois modifications (dans l'ordre d'écriture). Ajouter une association « contraste » à la requête. En cas d'échec, modifier les rôles « définition » et « introduction » dans la requête. Enfin, ajouter la relation « narrower » à la requête.

Règle 2.

```
FOR REWRITING LEARNER MODEL
IF um-contain(:um, :concept)
  AND equal(learner-type(:u1), "occasionnel")
  AND hierarchical-level(:concept) between 1 AND 3
USE
  ADD_RELATION("contrast", :uq);
  ADD_RELATION("narrower", :concept);
END;
```

La règle 2 est utilisée pour réécrire le modèle de l'apprenant (clause FOR) pour dériver de nouvelles connaissances permettant la satisfaction de la requête courante par ce modèle. La partie condition est analogue à la règle 1 et l'action consiste en deux modifications de type ajout d'association sur la requête ou sur un concept.

Règle 3.

```
FOR FILTERING ANSWER
IF q-contain(:uq, :concept)
  AND equal(learner-type(:u1), "occasionnel")
  AND answer-size(:uq) > 100
  AND hierarchical-level(:concept) between 1 AND 3
USE
  ADD_ROLES({"definition", "application"}, :uq);
END;
```

La règle 3 porte sur l'évaluation du résultat (clause FOR). Sa partie condition est analogue aux règles précédentes en y ajoutant une condition sur la taille du résultat. Sa partie action ajoute des rôles dans la requête.

3.2.5 Implantation

Le modèle de description proposé a été traduit en RDF (Resource Description Framework du W3C) pour pouvoir être implanté sur un outil supportant RDF (en l'occurrence Sesame [Broekstra 01]). L'avantage du formalisme RDF est qu'il commence à être supporté par plusieurs outils et qu'il doit servir de base au « web sémantique » avec un certain nombre de langages d'ontologies et d'inférence au-dessus de lui.

Nous décrivons ici succinctement comment faire correspondre notre modèle de description de ressources avec RDF et RDFS (c'est également fait pour le modèle de domaine et le modèle de l'apprenant) et nous montrons également comment le langage de requête SeRQL nous sert de premier niveau de mécanisme d'inférence.

3.2.5.1 Correspondance modèle de description avec RDF

Le schéma RDF de la figure 3-11 décrit les ressources atomiques et composites. Les ressources composites contiennent un graphe de composition appelé *los:graph*. Les deux types de ressources ont un contenu appelé *los:contents* et des pré-requis appelés *los:prerequisite* qui sont des valeurs structurées comprenant *ums:role*, *dms:concept* et un *los:LOM-dataElement* (caractéristiques générales de la ressource décrites selon le standard LOM). RDF-S manque de pouvoir d'expression et ne peut exprimer des contraintes de cardinalités.

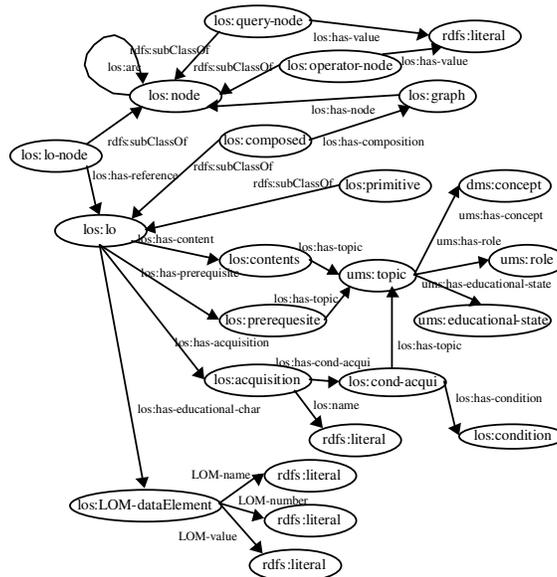


Figure 3-11 : schéma RDFS du modèle de ressource

Une instance de description de ressource se décrit de la manière suivante :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:los="http://bochica.udea.edu.co/~jfdutam/ecm/CM-schema#">
<los:primitive rdf:about="http://example.int.fr/CoursBdeD-INT/sql.html">
  <los:title xml:lang="FR">Le Langage SQL.</los:title>
  <los:typical-learning-time>180</los:typical-learning-time>
  <los:develops
    rdf:resource="http://example.int.fr/CM-
instance#contents-01"/>
  <los:develops
    rdf:resource="http://example.int.fr/CM-
instance#contents-06"/>
  <los:requires
    rdf:resource="http://example.int.fr/CM-
instance#prerequisite-01" />
</los:primitive>
  <los:prerequisite
    rdf:about="http://example.int.fr/CM-
instance#prerequisite-01">
    <los:has-role
      rdf:resource="http://example.int.fr/UM-
schema#definition"/>
    <los:has-concept
      rdf:resource="http://example.int-fr/DM-
instance#Relational-Algebra"/>
  </los:prerequisite>
</rdf:RDF>
```

Cette ressource atomique a un titre "Le langage SQL" et traite des concepts *contents-01* et *contents-02*; elle a de plus comme pré-requis *prerequisite-06*. Pour des raisons de simplicité, la définition de *content-01* et *content-06* est incomplète.

3.2.5.2 SeRQL comme niveau d'inférence

SeRQL peut être utilisé à différents niveaux dans notre système. Il est bien adapté pour rechercher ou naviguer dans le modèle de domaine (avec les expressions de chemins), le modèle de l'apprenant et le modèle de ressources. Nous donnons ici quelques exemples de l'utilisation de SeRQL pour naviguer dans les différents modèles ainsi que pour supporter les différents modes pédagogiques [Bouzeghoub 04].

La requête 1 ci-dessous filtre les apprenants qui ont une connaissance des concepts "S.Q.L" ou "Relational algebra". La clause FROM définit les expressions de chemins utilisés dans le reste de la requête (nous avons ici trois expressions de chemins avec la même tête "{ums:user} <ums:knows> {ums:topic}... <ums:has-topic> {topic}" et différentes queues). La clause WHERE filtre les graphes dénotés par la clause FROM et la clause SELECT extrait les nœuds intéressants.

Requête 1

```
SELECT User, Role, EducationalState
FROM {ums:user} <ums:knows> {ums:topic}<ums:has-topic> {ums:topic};
<ums:has-concept> {dms:Concept};
<ums:has-role> {ums:Role};
<ums:has-educational-state> {ums:EducationalState}
WHERE Concept="S.Q.L" or Concept="Relational algebra"
```

En mode concept, une fonction importante pour l'adaptation est de vérifier que le modèle de l'apprenant satisfait les pré-requis de la ressource. Cette propriété est calculée par les requêtes 2 et 3 (si le résultat de la requête 2 est inclus dans le résultat de la requête 3, la propriété est satisfaite).

Requête 2 recherche les pré-requis de la ressource "http://int-evry.fr/CM-exemple#DataBase".

```
SELECT Concept, Role, EducationalState
FROM {http://int-evry.fr/CM-exemple#DataBase} <los:has-prerequisite> {prerequisite}
<los:has-topic> {topic} <los:has-concept> {Concept};
<los:has-role> {Role};
<los:has-educational-state> {EducationalState}
```

Requête 3 recherche la connaissance de l'apprenant "http://int-evry.fr/U-example#Bob".

```
SELECT Concept, Role, EducationalState
FROM {http://int-evry.fr/U-example#Bob} <ums:knows> {topic} <ums:has-concept>
{Concept};
<ums:has-role> {Role};
<ums:has-educational-state> {EducationalState}
```

En mode concept, l'adaptation peut nécessiter de reformuler la requête en utilisant les relations du modèle de domaine. La requête 4 effectue une telle recherche.

Requête 4 : recherche les concepts du modèle de domaine en relation "narrower" par rapport concept "SQL".

```
SELECT Concept
FROM {<!http://int-evry.fr/DMI#SQL>}<dms:narrower> {Concept}
```

SeRQL offre une puissance d'expression assez satisfaisante pour exprimer nos requêtes. Néanmoins, nous aurions besoin d'opérations ensemblistes (par exemple pour évaluer les propriétés définies en 3.4.3) ainsi que de récursion (par exemple pour calculer les « voisins » d'un concept à n'importe quel niveau de profondeur).

3.2.6 Bilan

Ce projet d'équipe a permis à J.F Duitama de préparer sa thèse [Duitama 05] et de faire plusieurs publications [Bouzeghoub 03a, Duitama 03, Bouzeghoub 04]. L'idée retenue ici est d'ajouter de l'information sémantique pour pouvoir mieux manipuler les ressources que ce soit pour les rechercher, les réutiliser pour composition d'autres ressources ou les diffuser de manière personnalisée à des apprenants. Ce qui me paraît le plus novateur, c'est d'utiliser le même formalisme à la fois pour indexer/rechercher, composer et adapter des ressources pédagogiques. Le modèle retenu est simple mais semble suffisamment complet. Il repose cependant sur un modèle de domaine qui doit être fourni par un administrateur et qui dans l'absolu doit être assez riche (description de rôles et de nombreuses relations entre concepts). Nous supposons de même que les auteurs vont fournir une description sémantique de leurs ressources ce qui n'est pas évident. Pour les convaincre, il va falloir leur prouver que ce travail n'est pas inutile et qu'en retour ils vont bénéficier de capacités importantes de réutilisations d'autres ressources. Par contre, nous n'avons pas besoin, pour l'instant, de capacités inférentielles très importantes (un langage de requêtes nous suffit) ce qui permet une mise en œuvre aisée. De nombreuses pistes de recherche s'offrent à nous. Je ne décris ici que deux pistes à court terme, les autres sont évoquées dans la partie perspectives. Premièrement

il faut donner plus encore de flexibilité dans la définition d'une ressource complexe. En effet, plus il y a de flexibilité, plus les possibilités d'adaptation sont nombreuses. Pour l'instant, seul l'opérateur ALT introduit de la flexibilité. L'idée est de permettre de définir des ressources intentionnelles, c'est à dire une ressource pour laquelle un nœud (au moins) du graphe de composition est défini par une requête sur la base des ressources. Le système peut alors adapter la ressource en choisissant une ressource quelconque satisfaisant la requête. Certains problèmes doivent cependant être résolus : quand doit-on évaluer la requête, que se passe t'il, si la requête ne renvoie aucune réponse ou au contraire un très grand nombre de réponses, quelle est l'interaction avec le processus d'adaptation, ... ?

Deuxièmement, il faut prototyper et évaluer l'utilisation d'un tel environnement. Le prototypage est en cours mais doit être complété par la prise en compte (au moins minimale) de la présentation d'une ressource complexe. En effet, il va falloir construire une présentation la plus uniforme possible d'une ressource complexe. Ce n'est pas forcément évident puisqu'elle peut être composée de ressources développées indépendamment et donc avec des modèles de présentation différents. La partie test d'usage ne pourra être faite que si l'on dispose d'un ensemble suffisant de ressources. Une façon d'alimenter la base serait de développer un connecteur pour une base externe comme Ariadne ou Educanext. Le problème reste l'acquisition de la description sémantique des ressources dont la plus grande partie est manuelle. Il serait intéressant d'étudier l'apport d'algorithmes d'apprentissage pour calculer automatiquement (au moins partiellement) ces descriptions. Même si notre approche n'impose pas un niveau de granularité pour les ressources, il est bien évident que plus elles sont à grain fin plus la réutilisation va être importante. Une approche pourrait être de « découper » une ressource à grain trop gros et de la redéfinir comme une ressource complexe grâce à nos opérateurs.

Ce travail peut se rapprocher des travaux sur la standardisation des ressources éducatives, mais aussi des hypermédia adaptatifs ou des systèmes d'apprentissage utilisant le web. Les travaux sur la standardisation (LOM, SCORM) ne mettent pas assez en avant selon nous l'aspect sémantique des ressources éducatives. Nous nous contentons de suivre leurs recommandations sur les parties non sémantiques. Les hypermédia adaptatifs [Brusilovsky 96a] ont pour but de définir des modes de navigation personnalisés : ils reposent sur une architecture à trois niveaux similaire à la notre. Ils restent cependant généralistes et ne peuvent donc offrir des fonctions d'adaptation sophistiquées. Des systèmes d'apprentissage pour le web ont également été proposés [El Saddik 01, Brusilovsky 96b et 98] mais sont plus des systèmes auteurs mettant l'accent sur la construction de nouveaux contenus mais pas sur leur réutilisation. Les documents virtuels adaptatifs [Garlatti 04] sont très proches de nos travaux, même s'ils restent plus généralistes et n'incluent donc pas la dimension pédagogique.

3.3 Pair-à-pair sémantique

3.3.1 Introduction

Les systèmes pair-à-pair (*Peer-to-peer* ou P2P) sont devenus très populaires depuis l'avènement des systèmes de partage de fichiers (notamment musicaux) sur Internet (par exemple Napster, Kazaa ou edonkey). Ils reposent sur un principe d'égalité entre les nœuds et offrent une grande dynamique (les nœuds peuvent se connecter et se déconnecter à tout instant), et le passage à l'échelle (plusieurs milliers voire millions de nœuds). Pour ce faire, ils s'appuient sur une organisation la plus décentralisée possible du système (pas d'état global et auto-organisation). Le principe de résolution d'une requête est très simple : la requête est résolue localement au nœud et propagée sur des nœuds voisins de manière récursive. Ces systèmes peuvent se classer en trois grandes catégories.

Les systèmes « purs P2P » (comme Gnutella) forment la première catégorie. Chaque nœud est complètement autonome et la propagation des requêtes se fait par inondation du réseau (le nombre de voisins et la profondeur de la propagation sont fixés). Ces systèmes sont très simples, mais ils supposent une stricte égalité entre les pairs, une topologie du réseau sous-jacent de type small-world (intermédiaire entre un graphe connecté et un graphe aléatoire) et une distribution des requêtes non uniforme (un petit nombre de requêtes sont posées très fréquemment). Les deux dernières hypothèses sont vérifiées dans les systèmes d'échanges de fichiers (mais pas forcément dans toutes les applications potentielles), alors que la première hypothèse ne l'est pas. Le principe d'inondation est également fort consommateur en bande passante, de même que le protocole de construction des tables de routage qui est basé sur de la découverte dynamique. Pour pallier ces problèmes de performances, d'autres catégories de systèmes P2P ont été proposées. Les systèmes hiérarchiques, comme les *super-peers*, remettent en cause le principe d'égalité des nœuds. Les nœuds à faible bande passante sont rassemblés dans des groupes gérés en mode client-serveur classique par un représentant qui est un nœud à bonne bande passante, et seuls les représentants des groupes sont interconnectés en mode P2P. Ces systèmes offrent un bon compromis entre l'efficacité du client-serveur et le passage à l'échelle du P2P. Pour augmenter la tolérance aux fautes et éviter des points de défaillance uniques, on peut introduire des super-pairs redondants [Yang 03]. Les systèmes structurés (appelés aussi *Distributed Hash Tables* ou DHT) recherchent à la fois l'efficacité et la garantie de recherche. Pour cela ils organisent l'espace des nœuds ainsi que la répartition des ressources sur les nœuds selon une structure de données efficace (par exemple des arbres de recherche comme dans P-Grid [Aberer 02a] ou un anneau pour Chord [Stoica 01]). Le prix à payer est la perte d'autonomie des nœuds tant en matière de stockage que de routage (c'est l'infrastructure de DHT qui décide).

Ces différentes approches ne peuvent s'utiliser dans tous les contextes applicatifs ce qui fait qu'aucune solution ne se dégage aujourd'hui. On peut penser que les trois catégories vont continuer à se développer et il peut être intéressant de travailler sur des solutions hybrides ou de travailler sur des services indépendants de la couche de routage.

Le projet JXTA de Sun [JXTA 04] va en partie dans ce sens et offre une base logicielle minimale en open source pour développer plus rapidement des applications et systèmes P2P. Pour cela il fournit un certain nombre d'abstractions (pair, groupes de pairs, pipe pour communiquer entre pairs, protocoles de découverte, ...) et la possibilité d'ajouter de nouveaux services. JXTA est aujourd'hui utilisé par plusieurs projets de recherche comme Juxmem [Antoniu 03].

Une autre piste de recherche qui nous semble particulièrement prometteuse pour la gestion de données dans les systèmes P2P non structurés est la prise en compte d'informations sémantiques pour optimiser la fonction de recherche. Classiquement, la fonction de recherche dans ces systèmes est « aveugle », c'est à dire qu'elle ne prend pas en compte la sémantique ni des nœuds, ni des utilisateurs, ni des requêtes. La sémantique sur les nœuds peut permettre de caractériser le contenu des nœuds (en supposant que les nœuds couvrent une thématique plutôt spécialisée) et donc de favoriser le routage en comparant la requête avec le contenu des nœuds. C'est l'approche suivie par exemple dans [Crespo 02]. Une autre possibilité consiste à utiliser cette sémantique pour organiser le réseau en groupes de nœuds sémantiquement voisins. De cette façon, une fois qu'une requête a trouvé le bon groupe, sa résolution consiste à explorer tous les nœuds de ce groupe. L'apprentissage sur les requêtes consiste à garder pour chaque requête (ou « modèle » de requêtes) les nœuds pertinents dans le passé. Cette information est ensuite utilisée pour router une requête en la comparant aux modèles existants. Cette information peut également être couplée aux utilisateurs (ou aux groupes d'utilisateurs). On peut alors associer à chaque utilisateur, ses requêtes passées et les nœuds pertinents. Là

encore, la résolution d'une requête d'un utilisateur est faite par comparaison avec ses requêtes passées. L'agrégation de cette information au niveau de groupes d'utilisateurs (ou communautés) peut améliorer le processus en augmentant la connaissance.

Nous présentons ci-après nos premiers travaux sur l'organisation du réseau en groupes (ensemble de nœuds « sémantiquement » proches).

3.3.2 Organisation sémantique d'un réseau pair-à-pair

Ce travail se déroule dans le cadre du projet PADOUE qui est financé par l'ACI GRID du ministère de la recherche et fait l'objet de la thèse de doctorat de Nicolas Lumineau à Paris 6. PADOUE s'intéresse au partage de ressources dans le domaine de l'environnement (données, cartes, programmes). Ces ressources sont décrites par un ensemble de méta-données. J'ai commencé à travailler sur ce projet pendant mon séjour d'études au LIP6 au moment où se projet démarrait. L'unité thématique du projet (les sciences de l'environnement) et le caractère en général spécialisé des nœuds en fonction des organismes qui les gèrent, nous ont conduit à travailler sur une approche d'organisation du réseau en groupes sémantiquement liés. Le réseau sous-jacent dispose d'un assez grand nombre de nœuds géographiquement très dispersés, mais est relativement peu dynamique (les nœuds sont en général des serveurs connectés à Internet). Nous ne sommes pas dans un cadre typique P2P mais plutôt dans un cadre intermédiaire entre le P2P et les bases de données réparties.

L'idée est de construire un réseau de type super-pair mais en regroupant dans le même groupe les nœuds ayant un contenu similaire. L'intérêt est de pouvoir diminuer le nombre de sauts pour atteindre les nœuds pertinents à une requête. Nous proposons un service centralisé qui va gérer l'organisation du réseau. La structure du réseau est fixée par avance (nombre de groupes et structure de grille rectangulaire entre les groupes) et le service va dynamiquement allouer les nœuds sur cette structure statique.

Il faut tout d'abord caractériser le contenu des nœuds. Nous utilisons la même approche que [Crespo 02]. Un nœud est décrit par un vecteur thématique constitué par un ensemble de concepts issus du domaine de l'environnement. Cet ensemble de concepts de référence est partagé par tous les nœuds (initialement le nœud peut acquérir cet ensemble de référence en contactant le service). Un classifieur local au nœud doit ensuite pondérer ce vecteur, de manière à évaluer les thèmes couverts par ce nœud (voir figure 3.12). La somme des pondérations doit être égale à 1. Sous réserve de respecter cette propriété, n'importe quel classifieur peut être choisi. Le vecteur thématique est stocké par le nœud.

Thèmes	Climatologie	Hydrologie	Océanographie	Ecologie	...
Proportions	0,35	0,25	0,40	0	...

Figure 3.12 : exemple de vecteur thématique

Un groupe (ou cluster) va être défini comme un regroupement de nœuds partageant un contenu similaire. Le contenu d'un groupe va être estimé par un vecteur, identique aux vecteurs de nœuds, mais construit par agrégation des vecteurs de nœuds appartenant au groupe. Ce vecteur de groupe va être ensuite utilisé pour déterminer à quel groupe appartient un nœud entrant dans le système (le vecteur du nœud va être comparé aux différents vecteurs de groupe). La notion de groupe étant abstraite, le vecteur agrégé ne peut être stocké au niveau d'un nœud. Le stockage se fait donc au niveau du service, qui stocke pour chaque groupe son vecteur agrégé (initialisé par une valeur aléatoire si le groupe est vide) et un lien sur un point d'entrée (nœud du groupe qui va être en charge du processus d'insertion).

L'algorithme d'insertion d'un nœud N_i fonctionne en quatre étapes [Lumineau 04] :

- Entrée du nœud dans le réseau : N_i doit contacter le service et lui transmettre son vecteur thématique ;

- Sélection du groupe le plus proche : le service calcule G_j le groupe le plus proche du vecteur thématique de N_i ;
- Choix d'un point d'entrée : si G_j est non vide (cas 1), l'adresse IP du point d'entrée de G_j est transmise à N_i . Sinon (cas 2), on choisit G_k le groupe non vide le plus proche G_j dans la grille et l'adresse IP du point d'entrée de G_k est transmise à N_i . De plus, N_i devient le point d'entrée de G_j ;
- Insertion de N_i : N_i peut construire sa table de routage à partir de celle du point d'entrée et celui-ci va également modifier sa table pour prendre en compte N_i ;
- Mise à jour du réseau : le vecteur agrégé de G_j est modifié pour tenir compte de N_i . Les vecteurs agrégés des voisins dans la grille doivent également être mis à jour. Si le choix du point d'entrée correspond au cas 1, les points d'entrée correspondant aux voisins de G_j dans la grille doivent être notifiés de l'insertion de N_i (sinon la grille serait incohérente). Dans le cas 2, il faut vérifier que le point d'entrée n'a pas sa table de routage pleine, sinon il faut élire un nouveau nœud comme point d'entrée.

Le calcul du vecteur agrégé est fortement inspiré des cartes auto-organisatrices de Kohonen [Kohonen 97].

Une simulation de l'algorithme a été mise en place et montre l'efficacité de la recherche en nombre de sauts comparée à une approche d'inondation pure. L'expérimentation semble montrer également que le nombre « idéal » de groupes correspond au nombre d'entrées dans le vecteur thématique.

3.3.3 Bilan et comparaison avec les autres approches

Ce travail se situe dans une démarche hybride mixant une approche centralisée avec un système P2P. La gestion du réseau est centralisée avec une définition des groupes qui est statique et sur la base d'une connaissance (l'ensemble des thèmes) qui est également partagée. Seule l'allocation des nœuds aux groupes est dynamique et fonctionne selon un algorithme d'apprentissage auto-adaptatif. De nombreuses évolutions sont possibles, notamment celles visant à réduire l'aspect centralisé et à augmenter la dynamique. Pour éviter de fixer le nombre de groupes a priori, on doit pouvoir travailler sur une structure à plus de deux niveaux. Il suffit alors de fixer le nombre de groupes de premier niveau, mais lorsque le nombre de nœuds à l'intérieur d'un groupe devient trop grand, il faut alors définir un processus d'éclatement (et on rajoute un niveau). Il peut aussi être intéressant d'étudier comment les algorithmes d'apprentissage utilisés peuvent s'étendre au cas où le vecteur thématique n'est plus (complètement) homogène. La prise en compte de la dynamique, et de la tolérance aux fautes, passe par la redondance du service et des points d'entrée d'un groupe. Enfin, comme dans toutes les approches minimisant le nombre de sauts, il faut également prendre en compte la proximité géographique des nœuds (le coût d'accès à un nœud quelconque n'est pas uniforme).

La problématique de recherche en P2P sémantique commence à se développer aujourd'hui. [Crespo 02] propose une indexation du contenu des nœuds avec un vecteur thématique. Cela permet de classer les nœuds en fonction d'une requête donnée et donc de la propager aux nœuds les plus pertinents. La principale limite est le coût des mises à jour qui peut être très élevé (le nombre de nœuds auxquels envoyer les mises à jour est non borné). [Shen 04] propose une architecture de type super-pair basée sur des hiérarchies de résumés. L'intérêt majeur est que l'approche n'impose pas de référentiel commun d'indexation, chaque nœud choisit son vocabulaire d'indexation et des techniques de réduction du nombre de dimensions sont utilisées lors de la construction du résumé pour limiter l'explosion de la taille de celui-ci. [Nakauchi 04] propose un système de recherche à base de mots clés et de relations sémantiques entre ceux-ci. Chaque nœud dispose d'une base de connaissances locale constituée d'un ensemble de mots-clés et de relations entre ceux-ci. Les relations entre mots-

clés sont utilisées localement lors de l'interrogation pour élargir la requête. Les bases de connaissances sont gérées de manière décentralisée mais deux processus permettent de les faire évoluer. D'une part, les utilisateurs estiment les résultats par *relevance feedback* qui est utilisé pour mettre à jour les relations entre mots-clés sur les différents nœuds ayant participé à la résolution de la requête. D'autre part un processus de synchronisation des bases de connaissances, déclenché lorsque deux nœuds se rencontrent, est proposé. [Haase 04] se situe dans le même courant de gestion décentralisée des bases de connaissances. Il suppose une ontologie partagée entre toutes les sources et un mécanisme de publication des bases de connaissances locales (appelées expertises) proche de la synchronisation définie dans [Nakauchi 04]. Des mesures de similarité sont proposées pour comparer deux expertises (et donc permettre de classer les nœuds) ou pour comparer une requête avec une expertise (ce qui permet de classer les nœuds en fonction d'une requête).

pSearch [Tang 03] couple des techniques de recherche d'informations avec du P2P structuré. L'index construit par le processus de recherche d'informations est stocké dans une table de hachage distribuée (CAN en l'occurrence) de telle manière à ce que les documents proches sémantiquement soient également proches dans l'espace CAN.

Depuis quelques années, de nombreux projets de recherche se sont lancés pour étudier l'apport des systèmes P2P à la gestion de données ([SIGMOD 03] en présente un recueil). Les problématiques abordées en gestion de données P2P portent sur l'évaluation de requêtes (comment étendre les langages de requêtes utilisés dans les DHT et en faire une évaluation efficace, par exemple [Gupta 03] s'intéresse aux requêtes portant sur des intervalles), sur les architectures de données réparties [Aberer 02b], et sur la prise en compte de schémas [Nejdl 03, Tatarinov 03, Arenas 03, Ooi 03]. [Nejdl 03] utilise RDF pour décrire les schémas et propose des techniques efficaces d'évaluation de requêtes RDF réparties basées sur une architecture *super-peer*. [Tatarinov 03] s'intéresse à la correspondance entre schémas pour interroger des sources hétérogènes. Chaque pair peut exporter des données ou définir un *peer schema* (sa vision du reste du réseau). Des mappings définissent la correspondance entre deux ou plusieurs *peer schemas* et sont construits par une approche mixte GAV et LAV. Le schéma global est alors remplacé par un réseau de « *mappings* » entre schémas locaux qui permet de construire de nouveaux *mappings* par transitivité. [Aberer 02b] fait une proposition très proche également basée sur des *mappings* entre schémas locaux (exprimés par une requête). Lorsqu'une requête r_1 exprimée sur un schéma s_1 est propagée sur un nœud de schéma s_2 , un *mapping* s_1, s_2 (direct ou obtenu par transitivité) est appliqué sur r_1 et produit une nouvelle requête r_2 . Si r_2 est trop « éloigné » de r_1 , elle n'est pas évaluée. Dans [Arenas 03] l'approche correspondance entre schémas est étendue pour prendre en compte des correspondances entre données via des relations de correspondance maintenues à jour dynamiquement via des triggers. Dans le projet PeerDB [Ooi 03], les *mappings* sont construits dynamiquement en utilisant de l'information sémantique sur les schémas à intégrer (ensembles de mots-clés). Des techniques de recherche d'information sont utilisées pour comparer les relations et attributs selon ces mots-clés et ainsi propager les requêtes sur les pairs dont les schémas sont voisins.

4 Conclusion et perspectives

4.1 Bilan

Mes recherches m'ont amené à travailler sur des thématiques variées depuis les SGBD à objets à l'interopérabilité des systèmes. Ces recherches se sont concrétisées par plusieurs contrats de recherche et publications de bon niveau ainsi que par l'encadrement (ou co-encadrement) de trois doctorants et d'une dizaine de DEA. Mes principales contributions ont porté sur :

- les modèles de données à objet : le modèle Estrella se situe tout à fait dans la lignée de ce qui deviendra les extensions objet dans SQL3 et peut être vu comme un précurseur des approches relationnel-objet ;
- l'intégration données et document : le travail effectué dans le cadre de Estrella/Omega, à savoir la définition d'un type abstrait *Text* et son index spécialisé associé dans le cadre d'un modèle objet se situe bien à l'état de l'art de l'époque ;
- l'expression de contraintes dans les bases d'objet : la gestion de contraintes d'intégrité est une fonctionnalité classique pour les bases de données relationnelles mais qui manquait dans les SGBD à objets. Mon but ici n'a pas été seulement de transposer ce qui existait pour le relationnel mais de proposer un nouveau modèle d'expression qui soit homogène avec les concepts objets (héritage, surcharge, encapsulation) et qui s'adapte aux schémas objets (notion de cohérence comportementale). La réflexion sur le modèle d'exécution m'a amené à étudier de plus près l'interaction entre le modèle transactionnel sous-jacent et le mode de vérification des contraintes. Les résultats obtenus sur ces deux dimensions ont été résolument novateurs ;
- l'interopérabilité relationnel-objet : il s'agissait de définir un méta-modèle capable de dériver aussi bien un schéma objet à la O₂ qu'en schéma relationnel étendu à la ESQL. Il s'agissait d'une forme simple d'interopérabilité puisqu'on ne traitait pas les problèmes liés à l'exécution ;
- l'intégration de schémas de SIG à l'aide de logiques terminologiques : dans ce projet j'ai abordé les problèmes de l'interopérabilité sémantique, c'est à dire comment comparer des concepts (et dans notre cas spatiaux) venant de schémas hétérogènes. La logique terminologique, très utilisée dans le cadre des schémas de bases de données classiques, a montré ici aussi son intérêt ;
- la description de ressources éducatives à l'aide d'un modèle sémantique : ici nous avons essayé de déterminer quel niveau de sémantique est souhaitable pour pouvoir rechercher, composer, réutiliser et personnaliser des ressources éducatives. Le modèle proposé bien qu'assez simple, remplit bien tous ses objectifs ;
- l'adaptation aux apprenants des ressources éducatives : nous avons montré comment un modèle « sémantique » de description de ressources donne des possibilités intéressantes d'adaptation ;
- la prise en compte de connaissances pour structurer un réseau pair-à-pair.

Bien que ces recherches semblent assez disparates au premier abord, elles se situent pourtant dans la continuité et s'appuient le plus souvent sur la modélisation des données. Le travail sur les objets est le point de départ de mes recherches en bases de données, ce qui est assez normal vu l'époque. A partir de là, j'ai évolué vers la prise en compte de la répartition et notamment de la diversité des sources (interopérabilité). Pour aller plus loin dans ce domaine, j'ai progressivement intégré des approches sémantiques. Leur inconvénient majeur est que l'on sort alors des solutions génériques (utilisables pour tout type d'applications) pour aller vers des solutions spécifiques (dans notre cas aujourd'hui le e-learning). Il faudra sûrement faire des allers et retours du spécifique vers le générique pour essayer de généraliser des solutions au départ ad-hoc.

4.2 Perspectives

Avant de présenter plus précisément mes projets de recherche, je voudrais donner ma vision de l'évolution du domaine des bases de données. Depuis toujours les bases de données se situent au cœur des systèmes informatiques et donc sont liées de manière très forte avec d'autres technologies (systèmes d'exploitation, langages de programmation, recherche d'informations, intelligence artificielle, ...). Cette interaction avec d'autres domaines s'est encore accentuée ces dernières années et il y a maintenant des besoins forts de recherches communes avec le domaine des réseaux par exemple. Pour moi, les projets de recherche à mener dans le futur devront réunir des équipes avec des compétences complémentaires. Nous avons la chance à l'I.N.T de regrouper des compétences en systèmes répartis, réseaux et bases de données. J'ai déjà travaillé dans le passé sur des projets communs transversaux (projet RNRT CASTOR par exemple) et j'ai envie d'en lancer de nouveaux. Il y a quinze déjà sont apparues les « nouvelles applications » pour les bases de données. La C.A.O a été remplacé par la biologie ou l'astronomie, mais il y a toujours des problématiques nouvelles sur lesquelles travailler. L'environnement technologique change lui aussi : nous disposons de plus en plus de calculateurs légers (cartes à puce, capteurs en tout genre) interconnectés par des réseaux sans-fil qui amènent de nouveaux problèmes pour notre domaine (requêtes continues sur des flux de données par exemple). L'avènement de l'Internet a permis de construire des bases d'informations de taille considérable pour lesquelles il faut construire de nouvelles formes de SGBD répartis (gestion de l'hétérogénéité, prise en compte d'un grand nombre de sources, systèmes dynamiques, ...) offrant un accès personnalisé au plus grand nombre. La sécurité devient de plus en plus centrale et les risques de violation des libertés individuelles toujours plus grands. Dans ce contexte les besoins en outils d'apprentissage pour dériver de nouvelles informations, « résumer » des données trop volumineuses, analyser des usages, ... se font cruciaux mais doivent respecter la confidentialité des données privées. Que de défis à relever pour ces prochaines années !

Mes projets de recherche se positionnent principalement sur l'axe « grandes bases d'informations et accès personnalisé ». Ils se situent dans le cadre du projet GET SIMBAD démarré en janvier 2004 et qui regroupe quatre permanents (A. Bouzeghoub, C. Lecocq et S. Tata en plus de moi-même) et trois doctorants (I. Chebbi, J.F Duitama, N. M'Barek) et que je dirige. Les axes principaux du projet SIMBAD sont l'utilisation de la sémantique pour résoudre les problèmes d'interopérabilité et de personnalisation des systèmes d'information. Pour aborder ces problématiques complexes, nous voulons travailler en deux temps : étudier un domaine d'application particulier et essayer ensuite de généraliser les résultats obtenus. Les deux domaines d'applications privilégiés sont le e-learning et les entreprises virtuelles. Ce projet se structure autour de trois thématiques principales : modèles et langages de description de ressources, interopérabilité et enfin personnalisation. Sur chaque thème, nous partons d'un cas particulier que nous essayons ensuite de généraliser.

4.2.1 Modèles et langages de description de ressources

La modélisation de la sémantique retrouve un regain net d'intérêt avec l'avènement du « web sémantique » [Berners Lee 01]. La notion d'ontologie devient centrale et de nombreux travaux proposent différents formalismes et modèles de représentation (par exemple OIL, DAML et maintenant OWL du W3C). Les modèles de conception de bases de données et les ontologies offrent de nombreuses ressemblances [Cullot 03]. Les différences essentielles restent le manque de capacité de raisonnement des modèles bases de données (la notion de vue est par trop restrictive) et le caractère plus générique (partageables entre plusieurs applications) des ontologies [Spyns 02]. Nous allons tout à fait dans ce sens et pour nous un bon modèle d'ontologie est celui qui fournira les meilleures capacités inférentielles. Nous avons choisi d'utiliser RDF comme outil d'implantation de nos modèles en espérant que des

couches supérieures de raisonnement viendront s'ajouter aux langages de requêtes type SeRQL qui restent encore insuffisants.

Le projet SIMBAD s'intéresse principalement aux environnements de partage de ressources, qui peuvent être des données et/ou des programmes. Ces ressources sont décrites par un ensemble de méta-données (standardisées ou non) et sont gérées via des architectures réparties. Dans le modèle de description de ressources éducatives proposé, l'auteur a peu de contraintes lorsqu'il construit une ressource complexe. Il doit juste vérifier les propriétés de construction du graphe (un seul point d'entrée et pas de cycle) et de cohérence (un concept pré-requis d'une ressource R_1 ne peut être acquis par une ressource R_2 placée plus loin dans le graphe). Il est bien évident qu'il peut être intéressant de contraindre la construction de ressources, par exemple pour faire en sorte que toutes les ressources aient une structure comparable, ou bien pour éviter des ressources avec un contenu « non pédagogique » (avec trop peu ou trop de concepts abordés), ou bien pour imposer un « plan type ». Une notion de typage peut prendre en compte ces différents aspects. Ce typage pourrait prendre trois formes : un typage sur la structure (topologie) du graphe de composition (fixer un nombre maximum et/ou minimum de nœuds ressources, définir le degré maximum sortant des nœuds, limiter le nombre de nœuds ALT, ...), un typage sur la sémantique véhiculée par la ressource (définir un nombre maximum et/ou maximum de concepts traités ou bien imposer un certain nombre de rôles pour tout concept traité, ...) et enfin un typage mixte structure et sémantique (définir un type traitant d'un concept et avec pour ce concept une séquence composée d'un rôle *introduction* puis d'un rôle *présentation* et enfin d'un rôle *exemple*). Ce travail sur le typage pose deux grandes questions : quel(s) langage(s) d'expression du typage (et quel modèle de typage associé) et comment valider les types. En ce qui concerne la première question, il faut sûrement distinguer entre le langage offert à l'administrateur et le langage de représentation interne du type. La spécification du type pourrait se faire en définissant une sorte d'hypergraphe de composition où les nœuds et les arcs pourraient être des abstractions de graphes. De plus, chaque nœud ou arc pourrait être annoté par un ensemble de contraintes. Le choix d'un formalisme de représentation interne est plus délicat. Il est sûr que RDFS n'offre pas la puissance d'expression suffisante. Il faut donc se tourner vers des solutions venant soit du monde des langages de programmation (automates, lambda calcul, théorie des catégories), soit des graphes (grammaires de graphes).

Ce travail mené sur les ressources éducatives doit être étendu dans deux dimensions : d'une part la prise en compte de services et d'autre part la généralisation à d'autres domaines d'application. Sur la première dimension, dans la lignée de OWL-S [OWL 04] nous étudions des extensions sémantiques pour DAML (langage de description des services web). Pour la deuxième dimension, nous envisageons de mettre en évidence un cœur commun de concepts qui pourront être la base d'un méta-modèle générique qui serait ensuite « spécialisé » pour différents domaines d'applications. Pour cela, une piste à explorer est la description d'ontologies avec un niveau générique et des ensembles de contraintes permettant d'en donner plusieurs spécialisations (DOGMA [Jarrar 02] peut être un point de départ).

4.2.2 Interopérabilité

L'interopérabilité entre bases de ressources éducatives (et plus généralement entre bases de ressources décrites par des méta-données) est un sujet abordé dans le réseau d'excellence ProLearn. En fonction du nombre de bases à faire interopérer, le problème est différent. Si l'on suppose qu'il existe quelques bases gérant chacune un nombre important de ressources alors une solution à base de médiateur peut s'appliquer. Par exemple, on peut vouloir faire interopérer la base Ariadne et la base Educnext. Dans ce cas, il « suffit » de développer un

médiateur offrant un langage uniforme d'accès aux bases (et un formalisme de représentation des résultats) ainsi que des *wrappers* vers le langage des différentes bases. Une variante, plus simple, de cette solution est actuellement étudiée dans ProLearn. Il s'agit de définir une API standard d'interrogation des bases (appelée SQI [SQI 04]), supportée par chaque base. Cette API est indépendante du langage des bases, il s'agit juste d'offrir des abstractions de sessions et de requêtes : cela suppose donc que le client et la base distante se sont mis d'accord sur le langage d'accès à utiliser. Cette solution est peu satisfaisante pour un utilisateur final, puisqu'il doit maîtriser le langage des différentes bases mais elle peut être vue comme une couche basse pour un médiateur plus sophistiqué. Pour ProLearn nous allons développer une interface SQI pour nos bases de ressources SIMBAD.

Nous comptons généraliser ce travail en distinguant le type de couplage entre les sources. Nous distinguons les systèmes à couplage fort, généralement statiques (les nœuds se connaissent a priori) et pour lesquels un travail d'intégration de schéma doit être fait, et les systèmes à couplage faible, généralement dynamique (les nœuds commencent à se rechercher et ensuite ils coopèrent pour une durée assez courte) et pour lesquels il faut trouver un mode d'intégration plus simple. Un exemple typique du premier cas peut être une fédération de bases de ressources pédagogiques, alors que dans le second cas on trouve les entreprises virtuelles, regroupement d'entreprises établi pour la durée d'un projet spécifique. Dans SIMBAD, nous abordons le premier cas en étudiant les problèmes de correspondance (« mapping ») entre schémas (ou ontologies). Nous avons débuté en étudiant de manière manuelle les problèmes de « mappings » entre méta-données éducatives (le standard LOM, le modèle Ariadne et le modèle Educnext). Tout le travail effectué sur l'intégration automatique de schéma [Rahm 01] doit être repris et étendu pour prendre en compte l'intégration d'ontologies. La difficulté réside dans la prise en compte de représentations sémantiques plus complexes pour les ontologies que pour les schémas. [Doan 04] étudie les *mappings* 1-1 de deux taxonomies T_1 et T_2 en les basant sur la recherche pour chaque concept c_i de T_1 du « plus proche » concept dans T_2 . La distance peut être calculée par des fonctions de similarité fondées sur des distributions conjointes de probabilité. Ces distributions sont estimées par des fonctions d'apprentissage combinées via une stratégie d'apprentissage multiple. L'introduction d'heuristiques dans le système (par exemple le fait que deux concepts voisins dans une taxonomie doivent aussi l'être dans l'autre) est réalisée par l'utilisation d'une technique de *relaxation labelling*. Ces travaux, fort intéressants, montrent comment des techniques d'apprentissage symbolique peuvent être pertinentes dans ce contexte.

Nous voulons aussi étudier l'apport d'un autre courant prometteur, l'ingénierie des modèles [Bernstein 00, Melnik 03] qui met l'accent sur les modèles et leurs correspondances et propose de les manipuler comme des entités de première classe.

Le deuxième cas (couplage faible) est traité avec une vision plus service et workflow en étudiant comment décrire (de manière sémantique) et rechercher des services et ensuite construire des workflows inter-organisations. L'aspect dynamique implique qu'une intégration lourde ne peut être faite au préalable. Il faut que le « contrat » du service soit souple pour qu'il puisse s'adapter à un grand nombre de clients, tout en garantissant que l'interaction future ne causera pas de problèmes. Pour ce faire, nous étudions un modèle de contrat flexible qui autorise des divergences contrôlées. Ce travail fait l'objet de la thèse de I. Chebbi (co-encadrée avec S. Tata) qui a débuté en 2004.

Nous avons jusqu'à présent supposé que les sources étaient organisées via une architecture de médiation. Si nous voulons maintenant passer à l'échelle en nombre de sources (par exemple en faisant interopérer les bases individuelles de tous les professeurs d'un domaine particulier), la médiation ne peut plus s'appliquer. En effet, la complexité en nombre de langages va être

trop importante d'une part et d'autre part cela va être difficile de définir un schéma global. Dans ce contexte, nous pensons utiliser des solutions de type pair-à-pair. Plus généralement le pair-à-pair semble intéressant comme une alternative légère aux approches bases de données réparties classiques, dans les applications avec un grand nombre de sources et/ou à faible durée de vie. Le processus d'intégration de schéma, préalable à la médiation, est trop complexe pour être mis en place si la durée de la fédération des sources est faible. Pour cela de nombreux travaux restent à faire pour étudier comment deux sources peuvent dynamiquement « négocier » un « schéma commun » (*mappings* de schéma, comparaisons de schéma, négociation des systèmes multi-agents, ...).

4.2.3 Personnalisation

La personnalisation consiste à intégrer de l'information sur les utilisateurs (ou les groupes d'utilisateurs) pour améliorer le traitement fait par le système [Bouzeghoub 03b]. Traditionnellement dans les systèmes de gestion de données, cela consiste à réécrire les requêtes utilisateurs pour y injecter leur profil. Dans les systèmes de recherche d'information, l'utilisateur est mis dans la boucle de résolution de requête pour qu'il guide le processus, par exemple via du bouclage de pertinence ou « *relevance feedback* ». Dans SIMBAD, nous nous proposons d'introduire la notion de personnalisation plus au cœur du système et dans tous les processus.

Profils utilisateurs : Une première problématique porte sur la définition, la modification et l'acquisition de profils utilisateurs. Quelques standards commencent à être proposés (par exemple P3P du W3C mais qui s'intéresse plus à la protection des données privées qu'à leur représentation), mais les solutions retenues restent largement spécifiques et nous voulons travailler à leur généralisation. De plus, la notion de groupe (ou communauté) nous paraît fort importante pour compléter (voire remplacer lorsque l'information n'est pas disponible) le profil d'un utilisateur. Cela suppose de développer des techniques de constitution de groupes (statiques mais aussi dynamiques) ainsi qu'être capable d'agréger des profils. L'acquisition de ces profils peut être manuelle ou bien réalisée par des techniques d'apprentissage (nous ne traitons pas ce problème dans SIMBAD). Tout le travail sur le profil doit prendre en compte dès sa conception la protection des données privées. Dans un contexte de systèmes avec un très grand nombre d'utilisateurs, la gestion efficace de profils devient importante et doit être développée.

Processus de personnalisation : Dans un système de gestion de données mono-source, la personnalisation peut se faire au niveau de l'expression des requêtes, de l'annotation de requêtes et l'évaluation des requêtes. Pour l'expression des requêtes l'utilisateur doit pouvoir définir des « vues » sur le schéma ou l'ontologie, la liaison entre une vue et le schéma se faisant par un « mapping ». Il faut ici définir quels couples « vue, *mapping* » sont possibles. On peut distinguer deux types de vues. Le premier type recouvre les vues définies par simplification du schéma initial (supprimer des nœuds, supprimer des arcs, ...) et pour lesquelles les *mappings* vont être assez simples. [Villanova-Oliver 03] propose une telle approche dans un contexte de données XML mais avec des opérations assez simples. Dans le deuxième type les vues sont définies sans référence au schéma initial et donc les *mappings* vont être beaucoup plus complexes. Par exemple dans un contexte *e-learning* on peut imaginer un apprenant voulant interroger n'importe quelle base de ressources pédagogiques du domaine informatique via l'ontologie définie par l'ACM. Ce deuxième cas peut se ramener aux *mappings* de schéma/ontologie évoqués dans la partie interopérabilité. L'annotation de requêtes consiste à réécrire la requête utilisateur en y intégrant des informations de son profil (par exemple pour aider au filtrage ou introduire des clauses de classement des résultats) ou en définissant des critères de « qualité de service » (complétude souhaitée du résultat, qualité/pertinence souhaitée des réponses, temps de calcul, ...). Les systèmes actuels ne

prennent généralement en compte que ce niveau de personnalisation, nous proposons d'aller plus loin en intégrant des informations utilisateurs (sous forme de règles) dans le processus même de résolution des requêtes. Enfin, dans un système de données multi-source, la prise en compte de connaissances sur l'utilisateur (ou des groupes d'utilisateurs), par exemple à travers un processus d'apprentissage sur les requêtes passées, va permettre une meilleure sélection des sources sur lesquelles résoudre la requête. Nous travaillons dans ce sens en collaboration avec le LIP6, l'ENST Paris et l'ENST Bretagne dans le cadre du projet incitatif RARE qui vient de démarrer en janvier 2005.

5 Bibliographie

- [Aberer 02a] K. Aberer, M. Hauswirth, M. Puceva “Improving Data Access in P2P systems”, IEEE Internet Computing, 6(1), 2002
- [Aberer 02b] K. Aberer, P. Cudré-Mauroux, M. Hauswirth “A Framework for Semantic Gossiping”, ACM SIGMOD Record, 31(4), 2002
- [Abiteboul 89] S. Abiteboul, V. Vianu “A Transaction-Based Approach to Relational Database Specification”, Journal of the ACM, Vol36 N° 4, octobre 1989
- [Adiba 92] M. Adiba, C. Collet, P. Déchamboux, B. Defude “Integrated Tools for Object-Oriented Persistent Application Development”, Proc. Database and Expert System Applications – DEXA, Valence, Septembre 1992
- [Adiba 93] M. Adiba, C. Collet “Objets et bases de données : le SGBD O2”, Hermès, 1993
- [ADLI 01] ADVANCED DISTRIBUTED LEARNING INITIATIVE “Sharable Content Object Reference Model. The SCORM Content Aggregation Model”, Version 1.2, accessible depuis <<http://www.adlnet.org/>>, 2001
- [Amer Yahia 98] S. Amer Yahia, S. Cluet, C. Delobel “Bulk-Loading Technique for Object Databases and an Application to Relational Data”, Proc. VLDB Conf., 1998
- [Arenas 03] M. Arenas et al. “The Hyperion Project: From Data Integration to Data Coordination”, ACM SIGMOD Record, 32(3), 2003
- [Ariadne 02] ARIADNE FOUNDATION “ARIADNE Educational Metadata Recommendation V3.2”, accessible depuis http://www.ariadne-eu.org/en/publications/metadata/ams_v32.html, 2002
- [Aristote 89] Aristote “Environnement de conception et de développement d’applications orientées objets et multimedia”, RAP001, Rapport Aristote-LGI-Bull-IMAG, 1989
- [Baader 02] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider eds “Description Logic Handbook”, Cambridge University Press, 2002
- [Bancilhon 92] F. Bancilhon, C. Delobel, P. Kanellakis (Eds) “Building an Object-Oriented Database System : The Story Of O2”, Morgan Kaufmann, 1992
- [Batini 86] C. Batini, M. Lenzerini and S. B. Navathe “A comparative analysis of methodologies for database schema integration” ACM Computing Surveys, 18(4), pages 323-364, 1986
- [Benzaken 95] V. Benzaken, A. Doucet “Themis: a Database Programming Language Handling Integrity Constraints”, VLDB Journal, mai 1995
- [Benzaken 98] V. Benzaken, X. Schaeffer “Static Management of integrity in object-oriented database design and implementation”, Proc. EDBT Conference, LNCS, 1998

- [Benzaken 00] V. Benzaken, S. Cerrito, S. Praud “Static Verification of Dynamical Integrity Constraints: a Semantics Based Approach”, ISI, 2000
- [Bernes-Lee 01] T. Bernes-Lee, J. Hendler, O. Lassila “The Semantic Web”, Scientific American, 279, 2001
- [Bernstein 00] P. Bernstein, A. Levy, R. Pottinger “A Vision for Management of Complex Models”, ACM SIGMOD Record 29(4), 2000
- [Bezivin 03] J. Bezivin, E. Breton, G. Dupé, P. Valduriez “The ATL Transformation-based Model Management Framework”, Rapport interne IRIN 03-08, 2003
- [Bidoit 98] N. Bidoit, S. DeAmo “A First step towards implementing dynamic algebraic dependencies”, Theoretical Computer Science, 2(190) :115-149, janvier 1998
- [Borgida 02] A. Borgida, M. Lenzerini, R. Rosati “Description Logics for Databases”, in [Baader 02], 2002
- [Boucelma 02] O. Boucelma, M. ESSID, Z. Lacroix “A WFS-Based Mediation System for GIS Interoperability”, Proc. ACM GIS, Virginia, novembre 2002
- [Bouzeghoub 03a] A. Bouzeghoub, C. Carpentier, B. Defude “A Model of Pedagogical Components for the Generation of Adaptive Courses”, Proc. 4th International Conference on Information Technology Based Higher Education and Training, Maroc, juillet 2003
- [Bouzeghoub 03b] M. Bouzeghoub (coordinateur) “La personnalisation de l’information”, Rapport final AS CNRS-AS98/RTP 9, accessible depuis <http://prism.uvsq.fr/recherche/themes/sial/cnrs/Fichiers/Rapport/AS98-RapFinal.pdf>, 2003
- [Bouzeghoub 04] A. Bouzeghoub, B. Defude, S. Ammour, J.F. Duitama, C. Lecocq “A RDF Description Model for Manipulating Learning Objects”, Proc. IEEE ICALT Conf., août 2004
- [Brachman 77] R. J. Brachman “What's in concept: Structural foundations for semantic networks”, International Journal Man-Machine Studies, pages 127-152, 1977
- [Brachman 85] R. J. Brachman, J. G. Schmolze “An overview of the KL-ONE knowledge representation system”, Cognitive Science, 9(2), pages 171-216, 1985
- [Brachman 91] R. J. Brachman, D. L. Mc Guinness, P.F. Patel-Schneider, L. A. Resnick, A. Borgida “Living with CLASSIC: When and how to use a KL-ONE like language”, Principles of Semantic Networks: Explorations of the Representation of Knowledge, pages 401-456, 1991
- [Branki 97] T. Branki, B. Defude “A terminological Canonical Data Model for Cooperating Heterogeneous Geographical Information Systems”, Proc. DEXA Conference, Toulouse, LNCS 1308, septembre 1997
- [Branki 98a] T. Branki, B. Defude “Data and Metadata : Two-dimensional Integration of Heterogeneous Spatial Databases”, Proc. International Symposium for Spatial Data Handling (SDH98), Vancouver, juillet 1998

- [Branki 98b] T. Branki “Un processus d’intégration de bases de données spatiales hétérogènes par logiques de description”, Thèse de doctorat de l’université de Paris 13, septembre 1998
- [Broekstra 01] J. Broekstra, A. Kampman, F. Van Harmelen “Sesame: An Architecture for Storing and Querying RDF Data and Schema Information”, *Semantics for the WWW*, MIT Press, 2001, D. Fensel, J. Hendler, H. Lieberman and W. Wahlster.
- [Brusilovsky 96a] P. Brusilovski “Methods and Techniques of adaptive hypermedia”, *User Modeling and User Adapted Interaction*, 6 (2-3), 1996
- [Brusilovsky 96b] P. Brusilovsky “ELM-ART: An intelligent tutoring system on world wide web”, *Intelligent Tutoring Systems (Lectures Notes in Computer Science, Vol. 1086)*, 1996
- [Brusilovsky 98] P. Brusilovsky “Web-based education for all: A tool for developing adaptive courseware”, *Proc. International World Wide Web Conference*, 1998
- [CACM 91] CACM “Next Generation Database Systems”, *CACM* 34(10), octobre 1991
- [Cattell 97] R. Cattell et al. “Object Database Standard : ODMG 2.0”, Morgan Kaufmann pub., 1997
- [Ceri 94] S. Ceri et al. “Automatic generation of production rules for integrity maintenance”, *ACM Transactions on Database Systems*, Vol 19, N°3, pages 367-422, 1994
- [Chambers 99] C. Chambers, W. Chen “Efficient Multiple and Predicate Dispatching”, *Proc. OOPSLA’99*, Denver, 1999
- [Crespo 02] A. Crespo, H. Garcia-Molina “Routing Indices for Peer-to-Peer Systems”, *Proc. ICDCS 2002*
- [Cruz 02] I. Cruz, A. Rajendran, N. Wiegand “Handling Semantic Heterogeneities Using Declarative Agreements”, *Proc. ACM GIS*, 2002
- [Cullot 03] N. Cullot, C. Parent, S. Spaccapietra, C. Vangenot “Ontologies: A Contribution to the DL/DB Debate”, *Proc. First International Workshop on Semantic Web and Databases*, 2003
- [Damier 88] C. Damier, B. Defude “The Document Management Component of a Multimedia Data Model”, *Proc. 11th ACM-SIGIR Conference*, Grenoble, juin 1988
- [Damier 89] C. Damier “OMEGA : un SGBD multimédia orienté objet pour les applications géographiques”, thèse de doctorat informatique de l’I.N.P.G, juillet 1989
- [Damier 91] C. Damier, B. Defude “Le prototype OMEGA : un SGBD extensible pour les applications géographiques”, *Revue des Sciences de l’information géographique et de l’analyse spatiale, Hermès*, Vol 1, N°1, 1991

- [Dayal 90] U. Dayal, M. Hsu, R. Ladin “Organizing Long Running Activities with Triggers and Transactions”, Proc. of the ACM SIGMOD Conf., 1990
- [Déchamboux 91] P. Déchamboux, B. Defude “Modèle de base pour Aristote (version 2)”, RAP012, Rapport Aristote-LGI-Bull-IMAG, mai 1991
- [Déchamboux 93] P. Déchamboux “Gestion d’objets persistants : du langage de programmation au système”, Thèse de doctorat de l’Université Joseph Fourier Grenoble, juin 1993
- [Defude 94] B. Defude, H. Martin “From a Passive to an Active DBMS Supporting Exceptions”, Proc. of the DEXA Conf., Athens, LNCS 856, Springer Verlag, 1994
- [Defude 95] B. Defude, H. Martin, A. Picault “Un système de contrôle de l’activité de transactions basé sur des exceptions”, Proc. Journées BDA’95, Nancy, août 1995
- [Defude 96] B. Defude, H. Martin “Integrity Checking for Nested Transactions”, 7th International DEXA Workshop, Zurich, IEEE Press, septembre 1996
- [Defude 97] B. Defude “Un gestionnaire de cohérence pour le SGBD Versant”, Rapport INF/97/01, juin 1997
- [Devogele 97] T. Devogele. “Processus d’intégration et d’appariement de bases de données géographiques. Application à une base de données routières multi-échelles”, Thèse de Doctorat de l’Université de Versailles, décembre 1997
- [Devogele 98] T. Devogele, C. Parent, S. Spaccapietra “On spatial database integration”, International Journal of GIS, Vol 12, N°3, 1998
- [Doan 04] A. Doan, J. Madhavan, P. Domingos, A. Halevy “Ontology Matching: A Machine Learning Approach”, Handbook on Ontologies in Information Systems, S. Staab and R. Studer Eds, Springer Verlag, 2004
- [Doucet 99] A. Doucet, S. Gançarski, C. Léon, M. Rukoz “Nested Transactions with Integrity Constraints”, Proc. Transactions and Database Dynamics, LNCS 1773, Dagstuhl, 1999
- [Doucet 01] A. Doucet, S. Gançarski, C. Léon, M. Rukoz “Checking Integrity Constraints in Multidatabase Systems with Nested Transactions”, Proc. COOPIS’2001, Trento, 2001
- [Duitama 03] J.F. Duitama, B. Defude, A. Bouzeghoub, C. Carpentier “A Framework for the Generation of Adaptive Courses based on Semantic Metadata”, Proc. Workshop on Metadata and Adaptability in Web-based Information Systems in conjunction with OOIS’03 Conference, Genève, septembre 2003
- [Duitama 05] JF. Duitama “Un modèle de composants éducatifs pour des cours adaptatifs sur le Web”, thèse de doctorat informatique Université Evry Val d’Essonne– INT, soutenue le 25 mars 2005

- [El Saddik 01] S. El Saddik, S. Fischer, R. Steinmetz, R. “Reusability and Adaptability of Interactive Resources In Web-Based Educational”, ACM Journal of Educational Resources in Computing, 1(1), 2001
- [Esculier 90] C. Esculier, “Non monotonic knowledge evolution in VLKDBS”, Proc. VLDB Conf., Brisbane, août 1990
- [Faloutsos 84] C. Faloutsos, S. Christodoulakis “Signature Files: An Access Method for Documents and its Analytical performance evaluation”, ACM Trans. On Office Information Systems, 2(4), 1984
- [Fonseca 02] F. Fonseca, M. Egenhofer, P. Agouris “Using Ontologies for Integrated Geographic Information Systems”, Transactions in GIS, 6(3), 2002
- [Gardarin 90] G. Gardarin, P. Valduriez “ESQL: An Extended SQL with Object and Deductive Capabilities”, Proc. DEXA Conference, 1990
- [Gardarin 99] G. Gardarin “Bases de données objet et relationnel”, Eyrolles, 1999
- [Garlatti 04] S. Garlatti, S. Iksal “A Flexible Composition Engine for Adaptive Web Sites”, Proc. Adaptive Hypermedia and Adaptive Web-Based Systems, 2004
- [Greffen 93] P.W. Greffen, P.M Apers “Integrity Control in Relational Database Systems- An Overview”, Data and Knowledge Engineering, 10, pages 187-223, 1993
- [Gupta 93] A. Gupta, J. Widom “Local Verification of Global Integrity Constraints in Distributed Databases”, Proc. ACM SIGMOD, 1993
- [Gupta 03] A. Gupta, D. Agrawal, A. El Abbadi “Approximate Range Selection Queries in Peer-to-Peer Systems”, Proc. CIDR Conf., 2003
- [Guting 94] R. H. Guting “An Introduction to Spatial Database Systems”, VLDB Journal, Vol3, pp 357-399, 1994
- [Haase 04] P. Haase, R. Siebes “Peer Selection in Peer-to-Peer Networks with Semantic Topologies”, Proc. WWW Conf., 2004
- [Hoare 69] C.A.R. Hoare “An Axiomatic Approach to computer programming”, CACM pp 576-580, octobre 1969
- [Jarrar 02] M. Jarrar, R. Meersman “Formal Ontology Engineering in the DOGMA Approach”, Proc. CoopIS/DOA/ODBASE, LNCS 2519, 2002
- [JXTA 04] Site du projet JXTA, accessible depuis <http://www.jxta.org>, 2004
- [Keller 93] A.M Keller, R. Jensen, S. Agrawal “Persistence Software: Bridging Object-Oriented Programming and Relational Databases”, Proc. ACM SIGMOD Conf., 1993
- [Kim 90] W. Kim, J.F. Garza, N. Ballou, D. Woelk “Architecture of the ORION Next-Generation Database System”, IEEE TKDE 2(1):109-124, 1990

- [Kohonen 97] T. Kohonen “Exploration of very large databases by self-organizing maps”, Proc. International Conference on Neural Networks, 1997
- [Kon 02] F. Kon, F. Costa, G. Blair, R. Campbell “The case for Reflective Middleware”, Communication of the ACM, 45(6), 2002
- [Lacombe 92] C. Lacombe “Spécification du constructeur d’applications pour Aristote”, NOT017, Notice Aristote-LGI-Bull-IMAG, 1992
- [Laurini 98] R. Laurini “Spatial multidatabase topological continuity and indexing: a step towards seamless gis data interoperability”, International Journal of Geographical Information Systems, Special Issue: Interoperability in GIS, Volume 12, Number 4, pages 373-402, 1998
- [Leclercq 99] E. Leclercq, D. Benslimane, K. Yetongnon “ISIS : A Semantic Mediation Model and an Agent Based Architecture for GIS Interoperability”, Proc. IDEAS Conference, 1999
- [Lee 95] D.L Lee, Y.M Kim, G. Patel “Efficient Signature File Methods for Text Retrieval”, IEEE TDKE, 7(3), 1995
- [Lumineau 04] N. Lumineau, A. Doucet, B. Defude “Cluster Entries for Semantic Organization of Peer-to-Peer Network”, Proc. Int. Conf. On Semantics of a Networked World: Semantics for Grid Databases, poster session, juin 2004
- [Machado 92] J. Machado “Expression du parallélisme dans les applications de bases de données”, mémoire de DEA Informatique de l’INPG-UJF, juin 1992
- [Mann 87] W. Mann, S. Thomson, “Rhetorical Structure Theory: A theory of text organization”, Technical Report RS-87-190, Information Science Institute, 1987
- [Mascolo 02] C. Mascolo, L. Capra, W. Emmerich “Middleware for Mobile Computing (A Survey)”, Advanced Lectures in Networking, NLCS 2497, 2002
- [Martin 91] H. Martin “Contrôle de la cohérence dans les bases objets : une approche par le comportement”, Thèse de doctorat de l’Université Joseph Fourier, Grenoble, janvier 1991
- [Martin 93] H. Martin, B. Defude, M. Adiba “Consistency Checking in Object Oriented Databases: a Behavioral Approach”, Selected Papers, International Conference on Information and Knowledge Management- CIKM, Baltimore, USA, novembre 1992, Lecture Notes in Computer Science- LNCS 752, 1993
- [Melnik 03] S. Melnik, E. Rahm, P. Bernstein “RONDO: A Programming Platform for Generic Model Management”, Proc. ACM SIGMOD Conf, 2003
- [Metais 97] E. Metais, Z. Kedad, I. Comyn-Wattiau, M. Bouzeghoub “Using Linguistic Knowledge in View Integration : Toward a Third Generation of Tools”, Data and Knowledge Engineering, 23(1), 1997

[Morley 03] C. Morley, B. Defude, F. Butelle, D. Lang “Les outils du commerce électronique”, Annales des Télécommunications, Tome 58, N°1-2, Numéro spécial Commerce électronique le temps des évaluations, 2003

[Nakauchi 04] K. Nakauchi, Y. Ishikawa, H. Morikawa, T. Aoyama “Exploiting Semantics in Unstructured Peer-to-Peer Networks”, IEICE Transactions on Communications, E87-B(7), 2004

[Nejdl 03] W. Nejdl, W. Siberski, M. Sintek “Design Issues and Challenges for RDF and Schema-Based Peer-to-Peer Systems”, ACM SIGMOD Record, 32(3), 2003

[Nicolas 82] J.M Nicolas “Logic for Improving Integrity Checking in Relational Data Bases”, Acta Informatica, 18(3), pp227-253, 1982

[OMG 03] OMG “OCL 2.0 Revised submission”, version 1.6, janvier 2003

[Ooi 03] B. Ooi, Y. Shu, K.L Tan “Relational Data Sharing in Peer-Based Data Management Systems”, ACM SIGMOD Record, 32(3), 2003

[OpenGIS 04] The OpenGIS consortium, accessible depuis <http://www.opengis.org>, 2004

[Oszu 00] M. T Oszu, P. Valduriez “Principles of Distributed Database Systems”, 2nd edition, Prentice Hall, 2000

[OWL 04] The OWL Services Coalition “OWL-S : Semantic Markup for Web Services”, accessible depuis <http://www.daml.org/services/owl-s/1.0/owl-s.html>

[Papazoglou 03] M.P. Papazoglou “Service-oriented computing: Concepts, Characteristics and Directions”, Proc. WISE Int. Conf., 2003

[Peltason 91] C. Peltason “The back system- an overview” SIGART Bulletin, 2(3), pages 114-119, 1991

[Rahm 01] E. Rahm, P. Bernstein “On Matching Schemas Automatically”, VLDB Journal, 10(4), 2001

[Roncancio 91] C. Roncancio “Génération d’applications de bases de données selon l’approche objet”, DEA Informatique de l’INPG-UJF, juin 1991

[Schmidt 02] D. Schmidt “Adaptive and Reflective Middleware for Distributed Real-time and Embedded Systems”, Proc. Embedded Software Conf., 2002

[Shen 04] H. T Shen, B. Yu “Efficient Semantic-Based Content Search in P2P Network”, IEEE TKDE 16(7), 2004

[Sheth 90] A. P. Sheth, J. A. Larson “Federated database systems for managing distributed, heterogeneous and autonomous databases”, ACM Computing Surveys, Volume 22, Number 3, pages 183-236, 1990

- [Sheth 93] A. P. Sheth, S. K. Gala, S. B. Navathe “On automatic reasoning for schema integration”, International Journal of Intelligent and Cooperative Information Systems, 2(1), pages 23-50, 1993.
- [SIGMOD 03] K. Aberer ed. “Special Topic Section on Peer to Peer Data Management”, ACM SIGMOD Record, 32(3), 2003
- [Soley 00] R. Soley, OMG staff “Model-Driven Architecture”, accessible depuis <http://www.omg.org>, novembre 2000
- [Spaccapietra 99] S. Spaccapietra, C. Parent, E. Zimanyi “Spatio-temporal Conceptual Models : Data Structures + Space + Time”, Proc. ACM GIS, 1999
- [Spyns 02] P. Spyns, R. Meersman, M. Jarrar “Data Modelling versus Ontology Engineering”, ACM SIGMOD Record, 31(4), 2002
- [SQI 04] Simple Query Interface (SQI) for Learning Repositories, accessible depuis <http://nm.wu-wien.ac.at/e-learning/interoperability>, 2004
- [Stoica 01] I. Stoica et al. “Chord: a Scalable Peer-to-Peer Lookup Service for Internet Applications”, Proc. ACM SIGCOMM Conf., 2001
- [Tang 03] C. Tang, Z. Xu, S. Dwardakas “Peer-to-peer information retrieval using self-organizing semantic overlay networks”, Proc. ACM SIGCOMM, 2003
- [Tatarinov 03] I. Tatarinov et al. “The Piazza Peer Data Management Project”, ACM SIGMOD Record, 32(3), 2003
- [Yang 03] B. Yang, H. Garcia-Molina “Designing a Super-Peer Network”, Proc. ICDE Conference., 2003
- [Valduriez 87] P. Valduriez “Objets complexes dans les systèmes de bases de données relationnels”, TSI, 6(5), 1987
- [Vargas 02] G. Vargas Solar, A. Doucet “Médiation de données : solutions et problèmes ouverts”, Actes des 2^{ème} assises nationales du GDR I3, Nancy, décembre 2002
- [W3C 04] W3C Working Group “Web Services Architecture”, accessible depuis <http://www.w3.org/TR/ws-arch>, 2004
- [Villanova-Oliver 03] M. Villanova-Oliver, J. Gensel, H. Martin “A Progressive Access Approach for Web-based Information Systems”, Journal of Web Engineering, Rinton Press, 2(1-2), 2003
- [Wachter 94] H. Wachter, A. Reuter “The ConTract Model”, in Readings in Database Systems, 2nd edition, ed. M. Stonebraker, Morgan Kaufman, 1994
- [Woelk 86] D. Woelk, W. Kim, W. Luther “An Object-Oriented Approach to Multimedia Databases”, Proc. ACM SIGMOD Conf., 1986

Annexe

Liste des articles accompagnant le mémoire :

Partie 1 : SGBD à objets

- modèle de données

C. Damier, B. Defude “Le prototype OMEGA : un SGBD extensible pour les applications géographiques”, Revue des Sciences de l’information géographique et de l’analyse spatiale, Hermès, Vol 1, N°1, 1991

- gestion de documents dans un SGBD à objets

C. Damier, B. Defude “The Document Management Component of a Multimedia Data Model”, Proc. 11th ACM-SIGIR Conference, Grenoble, juin 1988

- expression et validation de contraintes d’intégrité

H. Martin, B. Defude, M. Adiba “Consistency Checking in Object Oriented Databases: a Behavioral Approach”, Selected Papers of CIKM Conference, Baltimore, USA, Novembre 1992, LNCS 752, 1993

B. Defude, H. Martin “From a Passive to an Active DBMS Supporting Exceptions“, Proc. DEXA Conference, Athènes, Septembre 1994, LNCS 856, Springer Verlag

B. Defude, H. Martin “Integrity Checking for Nested Transactions”, 7th International DEXA Workshop, Zurich, IEEE Press, Septembre 1996

Partie 2 : Interopérabilité

- interopérabilité relationnel et objet

M. Adiba, C. Collet, P. Déchamboux, B. Defude “Integrated Tools for Object-Oriented Persistent Application Development”, Proc. DEXA Conference, Valence, Septembre 1992

- modèle de données pivot pour intégration de données spatiales

T. Branki, B. Defude “A terminological Canonical Data Model for Cooperating Heterogeneous Geographical Information Systems”, Proc. DEXA Conference, Toulouse, LNCS 1308, septembre 1997

- un algorithme d’intégration de données spatiales

T. Branki, B. Defude «Data and Metadata : Two-dimensional Integration of Heterogeneous Spatial Databases», Proc. International Symposium for Spatial Data Handling (SDH98), Vancouver, juillet 1998

- Description de composants éducatifs selon un modèle sémantique

A. Bouzghoub, C. Carpentier, B. Defude “A Model of Pedagogical Components for the Generation of Adaptive Courses”, Proc. 4th International Conference on Information Technology Based Higher Education and Training, Maroc, Juillet 2003 et Moroccan Journal of Control Computer Science and Signal Processing, à paraître 2004

- Adaptabilité des composants éducatifs

J.F. Duitama, B. Defude, A. Bouzghoub, C. Carpentier “A Framework for the Generation of Adaptive Courses based on Semantic Metadata”, Proc. Workshop on Metadata and Adaptability in Web-based Information Systems in conjunction with OOIS’03 Conference, Genève, Septembre 2003; version longue à paraître dans Multimedia Tools and Applications, 25(3), 2005