Intégration de données hétérogènes, imprécises et incomplètes: Application dans le domaine du risque alimentaire

Patrice Buche

Université Paris Sud - Orsay

Document de synthèse présenté pour l'obtention de

l'Habilitation à Diriger des Recherches

Mention Informatique

par

Patrice BUCHE

http://metarisk.inapg.inra.fr/content/view/full/104

Intégration de données hétérogènes, imprécises et incomplètes: Application dans le domaine du risque alimentaire

Document provisoire, version du 5 avril 2007

Université Paris-Sud 11

Habilitation à diriger des recherches

Sciences Informatique

Soutenue le 7 Septembre 2007

Devant le jury composé de

- Patrick Bosc, professeur d'informatique à l'ENSSAT à Lannion.
- Thérèse Libourel, professeur d'informatique à l'université de Montpellier II.
- Janusz Kacprzyk, professeur au Systems Research Institute à Varsovie
- Henri Prade, directeur de recherche au CNRS à l'IRIT Toulouse.
- Chantal Reynaud, professeur d'informatique à l'université d'Orsay.
- Marie-Christine Rousset, professeur d'informatique à l'université de Grenoble.
- Laurent Rosso, directeur du laboratoire LERQAP (Laboratoire d'études et de recherches sur la qualité des aliments et sur les procédés agro-alimentaires) de l'AFSSA (Agence Française de Sécurité Sanitaire des Aliments).

Table des matières

Chapitre 1

Introduction

1.1 Problématique

Le besoin d'intégrer des données est essentiel dans notre société où l'information joue un rôle de plus en plus important. Des travaux précurseurs décrivaient dès le début des années 1930 des stations de travail permettant d'intégrer des données aujourd'hui qualifiées de multi-média (le sélecteur de microfilms d'Emmanuel Goldberg en 1931 et la station de travail intégrant des micro-films de Leonard Townsend en 1938). Le célèbre *memex* (pour memory extender) de Vannevar Bush (Bush 1945) était un appareil électronique relié à une bibliothèque capable d'afficher des livres et de projeter des films. Cet outil était aussi capable de créer automatiquement des références entre les différents média.

A partir du début des années 1970, le développement de la technologie des systèmes de gestion de bases de données a permis d'apporter des solutions concrètes aux besoins d'intégration de données. L'introduction du modèle relationnel (Codd 1970) et du système de gestion de données associé (Astrahan *et al.* 1976) révolutionnent la gestion des données jusqu'alors principalement réalisée sous la forme d'une collection de fichiers séquentiels et séquentiels indexés. Trente années d'innovations ont permis de rendre cette technologie extrêmement robuste et par voie de conséquence populaire, grâce à l'introduction (i) de techniques d'optimisation qui ont permis d'obtenir des temps d'exécution satisfaisants pour des requêtes complexes exécutées sur de gros volumes de données; (ii) de techniques de contrôle de concurrence d'accès aux données et de reprise sur panne qui ont permis d'obtenir des systèmes transactionnels fiables, (iii) d'architectures de bases de données extensibles permettant d'étendre les types de données atomiques afin de pouvoir intégrer dans les bases, des données complexes de type multi-média par exemple.

L'explosion, depuis une décennie, de l'Internet et du commerce électronique s'est accompagnée de l'explosion de la quantité et du type de données disponibles. Les applications informatiques d'aujourd'hui ont besoin de dialoguer avec des bases de données, des entrepôts de données, des systèmes de gestion de contenu, des moteurs de recherche, des serveurs d'application, des outils d'analyse et d'aide à la décision, des files de message et des systèmes de workflow. Ces

applications doivent extraire et combiner des données représentées dans des formats multiples et générées par des mécanismes d'acheminement variés. Par voie de conséquence, les concepteurs de la technologie informatique travaillent aujourd'hui à l'élaboration d'une plateforme qui offre une vue unifiée sur un ensemble de services et les données associées (voir Roth *et al.* 2002).

Cette plateforme repose sur une architecture en deux couches:

- La couche basse réalise l'intégration de données. Celle-ci permet de rendre accessible des données hétérogènes, internes ou externes à l'organisation, par une interface commune et un schéma d'intégration. L'objectif est que l'utilisateur ait l'impression d'interroger une base de données unique.
- La couche haute réalise l'intégration d'applications d'entreprise (EAI). Elle permet d'homogénéiser l'accès à un ensemble de fonctions proposées par des applications diverses, par exemple de type ERP (Enterprise Resource Planning) ou CRM (Customer Relations Management). Le modèle de programmation par flot (voir par exemple Leymann & Roller 2002) permet de réaliser ce type d'intégration intra ou inter-organisations. Les fonctions accèdent aux données via l'interface commune proposée par la couche intégration de données.

Dans ce mémoire, nous nous intéressons uniquement à la couche basse réalisant l'intégration de données. J'ai été confronté à la problématique d'intégration de données lorsque j'ai été sollicité en 1999 par le réseau Sym'Previus (http://www.symprevius.org/) pour concevoir une base de données dans le domaine du risque alimentaire. J'ai répondu à cette sollicitation car les données présentaient plusieurs caractéristiques, décrites dans la suite de cette introduction, qui nécessitaient un travail de modélisation important. Je vais maintenant présenter ce domaine d'application et y situer mon implication.

1.2 Domaine d'application

L'analyse du risque alimentaire est devenue une préoccupation stratégique pour les institutions nationales, européennes et internationales depuis la signature des accords SPS (Sanitaires et Phyto-Sanitaires) dans le cadre de l'OMC (Organisation Mondiale du Commerce) en 1994. Par ces accords, l'OMC reconnaît les standards de l'OMS (Organisation Mondiale de la Santé) en matière de qualité et de sécurité des aliments, ceux-ci étant élaborés dans le cadre du Codex Alimentarius depuis 1962. L'OMC reconnaît également que les pays ont le droit de définir leurs propres standards, qui peuvent être plus contraignants que ceux de l'OMS, en matière de qualité et de sécurité des aliments. Mais ce droit n'est reconnu que si ces standards ont pour objectif d'assurer la protection de la santé humaine. Pour cette raison, ils doivent reposer sur une argumentation scientifique crédible. On peut citer à titre d'exemple la réglementation européenne interdisant dans l'Union européenne la production et l'importation de viande aux hormones (essentiellement en provenance des Etats-Unis). En août 1997, cette réglementation a été jugée par l'OMC contraire aux dispositions des accord SPS, car les risques pour la santé humaine

n'ont pas été scientifiquement prouvés. On comprend donc l'importance de développer des méthodologies d'analyse du risque alimentaire basées sur une argumentation scientifique solide. Une étape importante dans l'élaboration de ces méthodologies est la constitution de systèmes efficaces de collecte et d'interrogation de données scientifiques.

En particulier, l'évaluation du risque chimique ou microbiologique, à laquelle nous nous sommes intéressés, ne peut se faire qu'à partir de données sur la présence de contaminants dans les aliments et le comportement de germes pathogènes dans ces mêmes aliments. La microbiologie prévisionnelle (voir McMeekin *et al.* 1993; McMeekin & Ross 2002; Nauta 2002; Coleman 2003) est un champ disciplinaire fondé sur les statistiques et la microbiologie, qui s'intéresse à la problématique de l'analyse et de la prévention du risque microbiologique. Les équipes travaillant sur cette problématique ont été amenées à construire des bases de données permettant une capitalisation de la connaissance en microbiologie prévisionnelle, telles que Pathogen Modeling Program, Seafood Spoilage Predictor, ComBase, Sym'Previus (voir par exemple les actes de la 4th International Conference of Predictive modelling in foods dans Impe *et al.* 2003). Les institutions internationales et les équipes de recherche travaillant sur l'évaluation du risque chimique ont développé une approche similaire. On peut notamment citer le programme GEMS/Food (http://www.who.int/foodsafety/chem/gems/en/) de l'OMS qui maintient une base de données internationale sur la contamination chimique des aliments depuis 1976 en collaboration avec un réseau de centres implantés dans 70 pays. Plus récemment, dans le cadre du projet européen SafeFoods (http://www.safefoods.nl/default.aspx), la constitution d'une base de données européenne sur les contaminants chimiques a également été entreprise.

Dans ce contexte, je me suis fortement impliqué dans la conception et la réalisation de deux systèmes de portée nationale et internationale:

- *le système du GIS*[1] *Sym'Previus*: de 1999 à 2005, j'ai coordonné la conception et la réalisation d'un système complet d'acquisition et d'interrogation de données expérimentales pour la microbiologie prévisionnelle. Ce système est opérationnel sur le site du GIS Sym'Previus, réseau national rassemblant de nombreux acteurs industriels, centres techniques et organismes de recherche dans le domaine de l'agro-alimentaire. Ce système interagit avec d'autres modules développés par des partenaires du GIS permettant, pour un microorganisme pathogène donné, d'une part de simuler sa croissance dans un aliment et d'autre part de déterminer les zones d'interface entre sa croissance et sa non-croissance.
- *le système de l'unité de recherche INRA Mét@risk*: depuis 2004, je coordonne la conception et la réalisation d'un système complet d'acquisition, d'interrogation et d'analyse de données permettant l'évaluation du niveau d'exposition d'une population cible à un contaminant chimique. Ce travail est effectué dans le cadre d'une coopération internationale, l'unité Mét@risk ayant le statut de centre collaboratif de l'OMS.

[1] Groupement d'Intérêt Scientifique

Nous avons conçu un système d'intégration de données à partir duquel nous avons implémenté ces deux systèmes. Trois caractéristiques importantes des données nous ont guidés dans la conception de ce système d'intégration de données.

1.3 Caractéristiques des données à intégrer

Les données sont hétérogènes: La nécessité d'intégrer en permanence dans notre système de nouvelles données biologiques de format variable rend difficile la conception d'un schéma unique de base de données, sauf à le faire évoluer en permanence. A titre d'exemple, un niveau de contamination d'un produit alimentaire pour un contaminant chimique peut s'exprimer comme une valeur atomique (cas du plomb ou du mercure), mais il peut également s'exprimer comme le résultat d'un calcul portant sur une liste de contaminants élémentaires (cas des dioxines et des PCB). Cette difficulté de structuration est rendue encore plus sensible pour plusieurs autres raisons:

- la nature du support: les publications scientifiques représentent l'une des sources principales de données à intégrer. S'agissant d'un document rédigé en langage naturel, même s'il existe des modèles à respecter (titre, résumé, introduction, matériel et méthodes, etc), la structuration de l'information est très variable d'un document à l'autre.
- la diversité des sources: en plus des publications scientifiques, dans les deux systèmes que nous avons réalisés, nous avons eu besoin d'intégrer des sources de données de nature différente générées par plusieurs partenaires (données industrielles, données institutionnelles, ...), mais aussi des données provenant du Web (rapports de projet, supports de cours, thèses, ...). La structuration de l'information y est également variable.
- l'évolution de la connaissance: la connaissance scientifique sur le comportement des contaminants dans les produits alimentaires (par exemple sur la croissance, la décroissance ou la survie pour un contaminant microbiologique) est en constante évolution. De nouveaux modèles de simulation de ce comportement sont régulièrement proposés pour mieux le comprendre. Ils intégrent de nouveaux paramètres qui doivent être pris en compte dans le système d'intégration de données.

Notre système d'intégration doit donc proposer des modèles de représentation de données permettant de gérer au mieux l'hétérogénéité de la structure des données. Une autre difficulté que nous avons rencontrée dans la conception de notre système d'intégration est l'hétérogénéité du vocabulaire utilisé dans les sources de données pour nommer des objets similaires (par exemple: "Rôti de bœuf" et "Carcasse de bœuf"). Il s'est avéré nécessaire de proposer une méthodologie permettant d'unifier le vocabulaire utilisé dans le système d'intégration de données. Ce problème est assez rarement évoqué dans la bibliographie en intégration de données, les auteurs se limitant à résoudre cette question au niveau de l'intégration des schémas des sources de données. Il est d'ailleurs évoqué dans la liste des sujets de recherche à développer (thème *fusion de données*) établie par le groupe international de chercheurs en bases de données réunis

à Lowell en 2003 (Abiteboul *et al.* 2005).

Les données sont imprécises: Les données biologiques peuvent présenter deux natures différentes d'imprécision :

- l'imprécision est due à la variabilité biologique. Lorsqu'une expérimentation est répétée, on obtient rarement la même valeur comme résultat de la mesure effectuée. Le résultat de l'expérimentation est donc par nature imprécis et représenté, par exemple, par une liste de valeurs obtenues pour chaque répétition, ou de manière plus synthétique par un intervalle délimité par les valeurs extrêmes.
- l'imprécision est due à la limite de résolution des capteurs utilisés. La mesure d'un contaminant chimique ou microbiologique est connue de manière précise si elle est supérieure au niveau de résolution du capteur utilisé. Dans le cas contraire, deux seuils sont classiquement utilisés: la limite de quantification (LOQ) et la limite de détection (LOD) avec $LOQ > LOD$. La mesure du contaminant peut donc être imprécise dans deux cas: (i) $LOD < mesure < LOQ$, le contaminant est présent à un niveau supérieur à LOD, mais il ne peut pas être quantifié, (ii) $mesure < LOD$ indique que la mesure de contamination est inférieure à LOD sans garantir qu'elle soit nulle.

Notre système d'intégration doit donc proposer des modèles permettant de représenter des données imprécises et des méthodes permettant de les interroger. Ce sujet fait également partie des questions de recherche en bases de données (thème *raisonnement sur des données incertaines ou imprécises*) listées dans Abiteboul *et al.* (2005).

Les données sont incomplètes: Le nombre de combinaisons de valeurs de paramètres pertinents pour décrire le comportement des contaminants dans les produits alimentaires est potentiellement infini. Par exemple, la vitesse de croissance d'un micro-organisme pathogène dans un produit alimentaire varie en fonction de la température de conservation mais également du pH du produit. Ces deux paramètres étant définis sur un domaine de valeurs à support continu, ils peuvent prendre eux-mêmes une infinité de valeurs possibles. Il est donc théoriquement nécessaire, pour un produit alimentaire donné et un contaminant donné, de gérer une infinité de résultats expérimentaux.

Dans la pratique, la quantité d'information que l'on peut stocker dans le système d'intégration est limitée par deux facteurs: le coût d'acquisition des données (coût de l'expérimentation et coût de stockage en base) et la confidentialité de l'information due à sa nature stratégique [2].

Notre système d'intégration doit donc proposer des méthodes permettant de pallier la rareté relative de l'information.

[2]Cette information est stratégique car elle est utilisée pour établir les standards internationaux en matière de sécurité des aliments évoqués dans la section 1.2.

1.4 Démarche de recherche

La problématique à laquelle je me suis intéressé dans mon travail de recherche en informatique est la prise en compte dans un système d'intégration de données des trois caractéristiques de données rencontrées dans le domaine du risque alimentaire: l'hétérogénéité, l'imprécision et l'incomplétude.

La première étape de mon travail a consisté à proposer un système d'intégration de données permettant de gérer l'hétérogénéité, l'imprécision et l'incomplétude des données internes à une organisation. J'ai imaginé pour cela, avec Ollivier Haemmerlé, alors, tout comme moi, Maître de Conférences à l'INA P-G, un système d'intégration de données qui permet d'interroger de manière unifiée, à partir de vues prédéfinies, à la fois des données structurées stockées dans une base de données relationnelle et des données faiblement structurées, stockées dans une base de graphes conceptuels. Cette proposition apporte *une solution au problème de l'hétérogénéité de structure des données internes*. Les données qui ne correspondent pas à la structure de la base relationnelle sont stockées dans la base de graphes conceptuels. Ce formalisme procure en effet une souplesse plus importante que le modèle relationnel pour représenter des données faiblement structurées (ce point sera plus amplement développé dans le chapitre 4). Par ailleurs, nous avons ressenti le besoin de définir la notion d'ontologie de notre système d'intégration de données pour permettre l'interrogation unifiée des deux bases. Cette ontologie est composée d'une part de la taxonomie des termes utilisés pour indexer les données stockées dans les deux bases et d'autre part des relations sémantiques définissant la nature des liens existant entre ces données. Nous avons co-encadré successivement deux stages de DEA, celui d'Hakima Kadri-Dahmani puis celui de Rallou Thomopoulos sur la conception de ce système d'intégration.

Puis, afin de proposer *une première solution au problème de l'incomplétude des données*, nous avons proposé d'introduire dans le système d'intégration, un mécanisme d'interrogation flexible, baptisé MIEL (pour Moteur d'Interrogation Elargie). Nous avons utilisé pour cela la théorie de la logique floue afin de modéliser les critères de sélection d'une requête sous forme de préférences priorisées représentées par des sous-ensembles flous. Cette modélisation permet de restituer à l'utilisateur l'information la plus pertinente par rapport à ses critères de sélection. Nous avons aussi proposé de *représenter des données imprécises dans les bases de notre système d'intégration* en nous appuyant sur la théorie des possibilités. Dans le cadre du co-encadrement de la thèse de Rallou Thomopoulos soutenue en 2003, nous nous sommes principalement intéressés à la représentation et à la comparaison, dans le modèle des graphes conceptuels, de données imprécises et de préférences exprimées dans des requêtes utilisateur en nous appuyant sur la théorie de la logique floue et la théorie des possibilités.

La deuxième étape de mon travail a consisté à proposer une deuxième solution au problème de l'incomplétude des données. A partir de Janvier 2003, dans le cadre du projet RNTL e.dot [3], en collaboration avec d'une part l'équipe GEMO [4] de l'INRIA et du LRI et d'autre part la

[3] http://gemo.futurs.inria.fr/gemo/edot
[4] http://gemo.futurs.inria.fr/gemo/

société Xyleme [5], nous avons étendu notre système d'intégration de données de telle manière qu'il soit possible de l'alimenter semi-automatiquement avec des données externes, extraites du Web. Cette extension a fait de nouveau ressortir le rôle central de l'ontologie du système d'intégration. Elle est un véritable pivot utilisé dans les étapes d'extraction des données du Web, d'annotation sémantique de ces données stockées dans une base XML et d'interrogation de ces données. En particulier, les termes de l'ontologie sont utilisés pour annoter les données externes de telle manière que le même vocabulaire soit utilisé pour décrire à la fois les données internes et externes. Cette annotation étant réalisée de manière automatique, elle peut contenir des erreurs. Nous la considérons dans ce mémoire comme une information imprécise, modélisée avec l'aide de la théorie de la logique floue et de la théorie des possibilités.

Dans le cadre du stage de DEA de Mounir Houhou, que j'ai co-encadré avec Ollivier Haemmerlé et Juliette Dibie-Barthélemy également Maître de Conférences à l'INA P-G, puis celui de François Rouillard que j'ai co-encadré avec Ollivier Haemmerlé, nous avons défini un format de représentation de données imprécises dans une base de données XML et proposé une extension de notre système d'intégration de données afin de pouvoir interroger une base de données XML floue. Ce système d'intégration, baptisé MIEL++, permet d'interroger simultanément et de manière transparente pour l'utilisateur, d'une part les deux bases locales, la base de données relationnelle et la base de graphes conceptuels, qui contiennent respectivement les données internes structurées et faiblement structurées, et d'autre part la base de données XML contenant les données externes provenant du Web.

Depuis Septembre 2004, je co-encadre avec Ollivier Haemmerlé et Juliette Dibie-Barthélemy, la thèse de Gaëlle Hignette. Cette thèse a pour objectif d'approfondir les travaux entrepris dans le cadre du projet e.dot. Il s'agit de proposer une méthodologie permettant d'une part l'annotation sémantique floue des données provenant du Web et d'autre part l'interrogation flexible de ces données annotées, ces deux opérations étant guidées par l'ontologie du système d'intégration. Les composants logiciel qui résulteront de ce travail seront intégrés à la plateforme du projet ANR/RNTL WebContent (http://www.webcontent.fr/), projet d'une durée de 3 ans de 2006 à 2009 auquel notre équipe de recherche participe.

1.5 Plan du mémoire

Le plan de ce mémoire s'articule autour de la description de notre système d'intégration de données en dégageant à chaque chapitre les aspects originaux.

Le chapitre 2 est un chapitre court consacré à la présentation globale de notre système d'intégration de données. Nous commençons ce chapitre par une présentation de l'état de l'art dans le domaine de l'intégration de données. Nous présentons ensuite l'architecture de notre système et son fonctionnement. Nous terminons ce chapitre en le positionnant par rapport à l'état de l'art.

[5] http://www.xyleme.com/

Le chapitre 3 est consacré à la présentation du langage d'interrogation flexible de notre système d'intégration de données. L'une des originalités de ce langage, dans le contexte des systèmes d'intégration de données, est qu'il autorise l'utilisateur à représenter ses critères de sélection sous la forme de préférences priorisées modélisées par des sous-ensembles flous. Cela permet au système de restituer des réponses qui sont ordonnées par leur degré de pertinence aux critères de sélection. Nous commençons ce chapitre en présentant un état de l'art dans le domaine de l'interrogation flexible de bases de données. Nous considérons en effet que les techniques d'interrogation flexible représentent une solution pertinente à l'incomplétude d'une base de données dans la mesure où elles permettent de pallier au manque de données en proposant à l'utilisateur, en complément des réponses exactes, des réponses qui sont proches sémantiquement. Nous présentons ensuite la théorie des sous-ensembles flous et son utilisation dans le domaine de l'interrogation des bases de données. Puis, nous introduisons la notion de sous-ensemble flou défini sur un domaine de valeurs organisé en taxonomie qui représente une autre originalité de notre système d'intégration. Enfin, nous définissons le langage d'interrogation flexible de notre système d'intégration.

Le chapitre 4 est consacré à la présentation des trois modèles de données utilisés pour représenter les données internes et externes de notre système d'intégration. Dans ce chapitre, nous mettons l'accent sur la représentation de données imprécises dans notre système d'intégration qui en constitue également une originalité. Nous commençons ce chapitre en présentant l'état de l'art concernant la représentation de données imprécises dans le domaine des bases de données. Nous présentons ensuite la théorie des possibilités que nous avons utilisée pour représenter des données imprécises dans nos trois sous-systèmes. Puis, nous développons les aspects les plus originaux de notre travail: (i) nous proposons une extension du modèle des graphes conceptuels et du modèle de données XML pour représenter des données imprécises, (ii) l'extension du modèle de données XML est utilisée pour effectuer des annotations sémantiques floues associées aux données du Web.

Le chapitre 5 est consacré à la présentation des adapteurs qui transforment une requête adressée à notre système d'intégration de données en une requête exécutable par chacun de ses trois sous-systèmes. Ce chapitre est divisé en deux parties qui correspondent aux aspects les plus originaux de notre travail concernant les adapteurs, à savoir l'adapteur permettant d'interroger la base de graphes conceptuels flous et celui permettant d'interroger la base de données XML floue.

Chapitre 2

Architecture du système d'intégration

2.1 Introduction

Ce chapitre court est consacré à la présentation globale de notre système d'intégration de données. Cette partie de mon travail a été réalisée, en ce qui concerne la conception du système d'intégration de données internes, dans le cadre de la thèse de Rallou Thomopoulos (Thomopoulos 2003), que j'ai co-encadrée avec Ollivier Haemmerlé. L'extension du système d'intégration à la prise en compte de données externes provenant du Web a été réalisée en collaboration avec Ollivier Haemmerlé et Juliette Dibie-Barthélemy. Le travail présenté dans ce chapitre a donné lieu aux publications suivantes:

- 1ère Journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA'05: Buche *et al.* (2005*b*),
- International Conference on Conceptual Structure, ICCS'2006: Buche *et al.* (2006*c*),
- Revue Fuzzy Sets and Systems: Buche *et al.* (2006*b*).

Nous commençons ce chapitre par une présentation de l'état de l'art dans le domaine de l'intégration de données. Nous présentons ensuite l'architecture de notre système et son fonctionnement. Nous terminons ce chapitre en positionnant notre système par rapport à l'état de l'art.

2.2 Etat de l'art en intégration de données

Depuis le début des années 1990, la communauté de recherche en informatique (notamment en bases de données) et l'industrie ont proposé deux approches complémentaires pour concevoir la couche d'intégration de données présentée dans l'introduction de ce mémoire: l'approche distribuée à base de médiateurs (Wiederhold 1992; Garcia-Molina *et al.* 1995; Levy *et al.* 1996*b*; Ullman 2000) et l'approche centralisée par la construction d'entrepôts de données (Widom 1995; Chaudhuri & Dayal 1997; Wu & Buchmann 1997; Vassiliadis 2000).

Dans l'approche *médiateur*, l'intégration des données s'effectue en deux étapes:

- Le système d'intégration récupère la requête de l'utilisateur, détermine l'ensemble des sources de données susceptibles de répondre à cette requête et génère l'ensemble des sous-requêtes appropriées qui sont soumises aux sources de données sélectionnées.
- Le système récupère ensuite les résultats transmis par les sources de données, effectue les transformations, filtrages et fusions nécessaires afin de pouvoir retourner la réponse finale à l'utilisateur.

On parle d'une approche *médiateur* en référence à Wiederhold (1992) qui a été le premier à proposer ce terme pour décrire la décomposition d'une requête en sous-requêtes et la combinaison des résultats obtenus par l'exécution de ces sous-requêtes. L'autre caractéristique importante de cette approche est que l'information n'est extraite de la source de données que lorsqu'une requête est posée par l'utilisateur. On peut alors parler d'une approche de l'intégration de données *à la demande*.

L'alternative naturelle à cette approche est l'intégration de données dite *par avance* ou encore *par construction d'entrepôt de données*. Dans l'approche *entrepôt de données*, l'intégration se fait également en deux étapes:

- Les informations pertinentes sont extraites par avance de chaque source de données. Elles subissent les transformations et filtrages nécessaires, avant d'être fusionnées et stockées dans un entrepôt de données centralisé.
- La requête de l'utilisateur est alors posée directement à l'entrepôt de données sans accès aux sources de données d'origine.

L'approche médiateur est bien adaptée dans les cas suivants: lorsque l'information change rapidement, lorsque les requêtes des utilisateurs correspondent à des besoins difficiles à prévoir, et lorsque, à la fois le nombre de sources et le nombre de données concernées sont très élevés. Par contre, cette approche peut se révéler inefficace lorsque le temps d'accès aux sources de données est long, lorsque l'accès est coûteux ou indisponible ou lorsque le temps de traitement des requêtes générées par le médiateur est long.

L'approche entrepôt de données est bien adaptée pour le traitement de requêtes prédéfinies, dont le temps de réponse doit être rapide (ceci est garanti par le fait que les données sont locales). Cette approche est également bien adaptée lorsque les utilisateurs veulent accéder à une information qui a été annotée ou synthétisée. Elle est indispensable si les utilisateurs veulent accéder à un historique de l'information qui n'est pas maintenu dans les sources de données (par exemple: un entrepôt de données d'achat cumulées sur plusieurs années constitué à partir de sources de données qui gèrent uniquement des données d'achat sur l'année en cours). Par contre, cette approche est mal adaptée si l'utilisateur veut accéder aux informations les plus récentes publiées dans un ensemble de sources de données qui sont fréquemment mises à jour. Il y a en effet toujours un décalage temporel incompressible entre la mise à jour de la source de données et l'intégration de cette mise à jour dans l'entrepôt.

Les deux approches sont donc toutes les deux pertinentes pour traiter le problème de l'intégration de données. Le choix de l'approche à adopter dépend des particularités du problème

d'intégration à traiter.

La communauté de recherche en informatique a été tout particulièrement active dans les années 1990 pour traiter un problème commun à ces deux approches (voir Halevy 2001 pour une synthèse) qui peut être résumé par la question suivante: comment répondre à une requête en utilisant des vues[1] ?

Dans l'approche entrepôt de données, cette question a été abordée dans l'optique d'optimiser des requêtes incluant notamment des opérations d'agrégation, par l'utilisation de vues matérialisées. A la différence d'une vue abstraite, relation virtuelle dont le contenu n'est jamais effectivement calculé, une vue matérialisée est une relation définie à partir de relations élémentaires dont le contenu est calculé et stocké en base. Si, dans une vue matérialisée, une partie des calculs nécessaires pour exécuter une requête a déjà été réalisée, alors il peut être intéressant d'utiliser cette vue pour optimiser l'exécution de la requête.

Dans le cadre de l'approche médiateur, la question de l'utilisation de vues pour répondre à une requête a été abordée avec pour objectif de transformer une requête utilisateur exprimée en termes de vues d'un schéma global d'intégration sous la forme d'une requête exprimée en termes de vues sur les sources de données. Considérons l'exemple suivant dans lequel le schéma global d'intégration est constitué de 3 vues:

- Film(Titre, Année, Réalisateur) qui contient une liste de titres de films avec son année de sortie et le nom de son réalisateur,
- Européen(Réalisateur) qui contient une liste de noms de réalisateurs européens,
- Revue(Titre, Critique) qui contient une liste de critiques d'oeuvres variées (film, livre, musique, etc).

Dans ce système d'intégration, deux sources de données sont accessibles par les vues V_1 et V_2 :

- V_1(Titre, Année, Réalisateur) qui rassemble des films réalisés par des réalisateurs européens depuis 1960,
- V_2(Titre, Critique), contenant des critiques de films depuis 1990.

Dans l'approche médiateur, la requête utilisateur Q est exprimée dans les termes du schéma global d'intégration. Par exemple, la requête *Liste des critiques de films sortis en 1998* s'exprime, dans un formalisme proche du calcul relationnel (défini dans Ullman 1988), en termes de vues du schéma global d'intégration de la manière suivante:

$$Q = \{C, T | Film(T, 1998, R) \wedge Revue(T, C)\}.$$

Le problème de la transformation de la requête en terme de vues sur les sources de données a été traité de deux façons différentes:

- si les vues sur les sources de données sont définies comme une conjonction de vues du schéma global, on parle d'une approche centrée sur les sources (ou LAV pour Local-As-View),

[1] relations définies par combinaison de relations élémentaires

– si les vues du schéma global sont définies comme une conjonction de vues sur les sources, on parle d'une approche centrée sur le schéma global (ou GAV pour Global-As-View).

Dans l'approche LAV, les deux vues V_1 et V_2 de l'exemple précédent sont définies de la manière suivante:

– $V_1 \subseteq \{T, A, R | Film(T, A, R) \wedge Europeen(R) \wedge A \succeq 1960\}$,
– $V_2 \subseteq \{T, C | Film(T, A, R) \wedge Revue(T, C) \wedge A \succeq 1990\}$.

Dans l'approche GAV, les vues du schéma d'intégration sont définies à partir des vues sur les sources de données de la manière suivante:

– $Film(T, A, R) \supseteq \{T, A, R | V_1(T, A, R)\}$,
– $Europeen(R) \supseteq \{R | V_1(T, A, R)\}$,
– $Revue(T, C) \supseteq \{T, C | V_2(T, C)\}$.

Ullman (2000) effectue une comparaison détaillée des deux systèmes pionniers représentant les deux approches: Information Manifold (Levy *et al.* 1995, 1996*b*,*a*) architecture de type LAV et Tsimmis (Garcia-Molina *et al.* 1995; Papakonstantinou *et al.* 1995*a*,*b*), architecture de type GAV. Dans l'approche LAV, la requête utilisateur Q est réécrite sous la forme d'une requête conjonctive Q' portant sur les vues correspondant aux sources de données (voir Halevy 2001 pour une synthèse sur les algorithmes de réécriture). La solution finale pour Q est l'union des résultats obtenus pour toutes les requêtes Q', réécritures possibles de Q. Dans l'approche GAV, la requête utilisateur Q est simplement transformée en remplaçant les noms des vues du schéma global par leur définition en terme de sources de données (voir Ullman 2000).

Quelle que soit l'approche utilisée, GAV ou LAV, la requête de l'exemple précédent, exprimée en terme des vues du schéma global d'intégration, est transformée pour donner la requête suivante, exprimée en terme des vues sur les sources de données:

$$Q' = \{T, C | V_1(T, 1998, R) \wedge V_2(T, C)\}.$$

2.3 Présentation de l'architecture de notre système d'intégration de données

Notre système d'intégration de données permet d'interroger simultanément trois bases de données dans lesquelles sont stockées les données internes structurées (base relationnelle) et faiblement structurées (base de graphes conceptuels) et les données externes provenant du Web (base XML). La figure 2.1 présente l'architecture globale du système MIEL++. Notre système d'intégration s'apparente à un système d'intégration de données de type médiateur dans la mesure où les données internes et externes sont stockées dans des sources de données différentes. Le schéma global d'intégration se compose d'une part d'une liste de relations sémantiques interprétées comme des vues pré-définies dans lesquelles l'utilisateur formule ses requêtes et d'autre part d'une taxonomie de termes qu'il peut utiliser pour spécifier ses critères de sélection. Comme nous l'avons indiqué dans l'introduction du mémoire, ce schéma global est appelé

ontologie du système d'intégration.

L'utilisateur du système MIEL++ dispose d'une interface graphique qui lui permet d'exprimer une requête dans le langage de requête MIEL. Comme l'utilisateur-cible du système est un non-informaticien, nous avons choisi de définir un langage de requête simple. L'utilisateur doit sélectionner une vue dans laquelle il spécifie, parmi l'ensemble des attributs disponibles, une liste d'attributs de sélection en leur associant une valeur et une liste d'attributs de projection. Une caractéristique importante du système d'interrogation est que les attributs à valeur symbolique (par exemple les noms de produits alimentaires, les noms de contaminants) sont définis sur un domaine de valeurs organisé en taxonomie. En effet, de nombreux systèmes de classification sont utilisés en biologie pour structurer l'information et tout particulièrement dans le domaine de l'alimentaire (voir Ireland & Moller 2000 pour une synthèse). L'interface graphique du système MIEL++ permet de visualiser cette taxonomie pour aider l'utilisateur à choisir les valeurs qui l'intéressent. La requête MIEL, une fois saisie par l'utilisateur, est transmise par le médiateur à chacun des trois sous-systèmes: le sous-système relationnel, le sous-système graphes conceptuels et le sous-système XML. Chaque sous-système transforme la requête MIEL en une requête exécutable sur sa propre collection de données. Le médiateur se met alors en attente de la réponse qui est une juxtaposition des réponses renvoyées par les trois sous-systèmes.

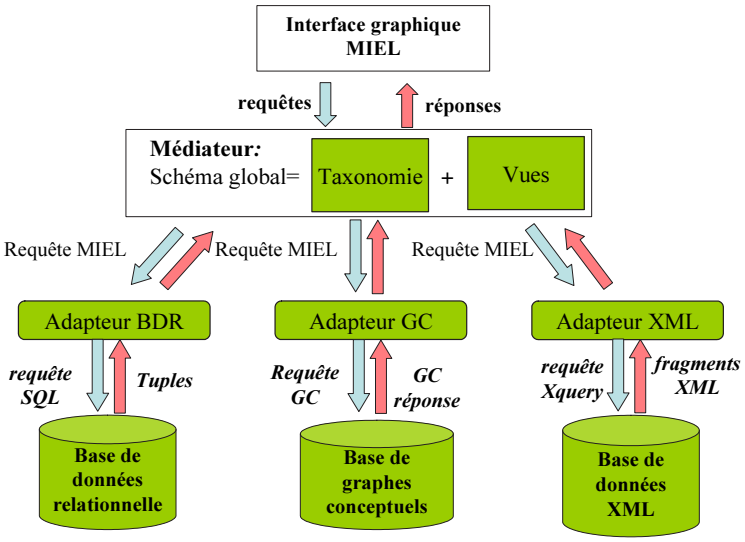


Fig. 2.1 – Architecture générale du système d'intégration de données MIEL++

2.4 Positionnement de notre système d'intégration de données par rapport à l'état de l'art

Notre système est une version simplifiée d'un système de médiation classique. En effet, alors que dans un système de médiation classique, une requête peut être exprimée comme une conjonction de vues du schéma global d'intégration, nous avons interdit cette possibilité dans le système MIEL. Nous avons fait ce choix afin de faciliter la prise en main du système par l'utilisateur final non-informaticien. Afin de simplifier au maximum l'expression de la requête, l'utilisateur a uniquement besoin de sélectionner une vue et d'exprimer ses critères de sélection dans cette vue.

Par ailleurs, notre système de médiation est une version simplifiée d'un médiateur de type GAV. En effet, premièrement, alors que dans un système de médiation classique de type GAV, un médiateur peut faire appel à un autre médiateur, dans notre système, il n' y a qu'un seul niveau de médiation qui distribue la requête aux trois sous-systèmes. Deuxièmement, à une vue du schéma global correspond, au plus une vue dans chacun des trois sous-systèmes. Ces choix ont été faits dans un souci d'efficacité: une requête MIEL correspond dans chacun des trois sous-systèmes à au plus une seule requête qui est générée et exécutée par l'adapteur du sous-système. Troisièmement, il n'est pas possible de définir une vue du schéma global comme une conjonction de vues sur des sous-systèmes différents. Nous n'avons pas permis les conjonctions de vues sur les sources de données car les utilisateurs n'apportent pas le même niveau de confiance aux données stockées dans les trois sous-systèmes. En ce qui concerne les données internes, le sous-système relationnel contient les données les plus fiables, alors que le sous-système graphes conceptuels contient des données intéressantes, mais marginales et par conséquent moins pertinentes que celles du sous-système relationnel. Les données du sous-système XML contiennent des données externes extraites du Web auxquelles les utilisateurs n'accordent pas la même confiance qu'aux données internes. Les données issues des trois sous-systèmes ne peuvent donc pas être combinées et sont restituées séparément à l'utilisateur.

2.5 Conclusion du chapitre

Par rapport aux systèmes d'intégration de données de type médiateur, nous avons fait le choix d'une architecture simplifiée essentiellement pour des raisons d'efficacité (en terme de temps d'exécution) et de simplicité d'utilisation pour des non-informaticiens. Notre système d'intégration de données est écrit en langage Java. Une requête MIEL est exécutée dans une architecture multi-tiers de type J2EE:

- l'interface graphique du client MIEL est générée par une servlet/JSP (Java Server Page) sur le serveur JSP et s'affiche dans un navigateur Internet;
- le médiateur communique avec le serveur JSP par le protocole RMI (Remote Method Invocation). Il dialogue avec le client MIEL, transmet la requête du client aux trois adap-

teurs, puis juxtapose les réponses renvoyées par les sous-systèmes afin de les présenter à l'utilisateur;
- chaque adapteur traduit la requête MIEL dans un format de requête exécutable dans son sous-système;
- chaque sous-système (RDB, GC, XML) exécute la requête adaptée.

Le sous-système relationnel de notre système d'intégration de données est opérationnel et mis en production sur le site du GIS Sym'Previus depuis 2005. Les deux autres sous-systèmes existent pour l'instant à l'état de prototype. La priorité est pour l'instant mise sur le sous-système XML qui sera intégré au système du GIS Sym'Previus dans le cadre du projet Web-Content (2006-09).

Nous envisageons dans l'avenir d'améliorer cette architecture notamment afin de limiter la réplication d'une partie des connaissances représentées dans l'ontologie du système d'intégration à l'intérieur des sous-systèmes. En effet, dans la version actuelle, la taxonomie faisant partie de l'ontologie est répliquée dans la base de données relationnelle pour permettre les contrôles d'intégrité référentielle. Elle est également répliquée dans le support terminologique de la base de graphes conceptuels pour permettre le bon fonctionnement de ce sous-système. Elle est enfin répliquée sous la forme d'un fichier XML dans le sous-système XML. Notre objectif est donc de modifier l'architecture du système d'intégration afin de permettre l'accès des trois sous-systèmes à une ontologie partagée.

Nous envisageons également d'ouvrir notre architecture afin d'y intégrer d'autres sources de données construites autour d'ontologies différentes de celle de notre système d'intégration de données. Nous nous intéresserons pour cela à la problématique de la mise en correspondance d'ontologies.

Chapitre 3

Interrogation flexible dans le système d'intégration de données

3.1 Introduction

Ce chapitre est consacré à la présentation du langage d'interrogation flexible de notre système d'intégration de données. Cette partie de mon travail a été réalisée dans le cadre du projet Sym'Previus et de la thèse de Rallou Thomopoulos (Thomopoulos 2003), que j'ai co-encadrée avec Ollivier Haemmerlé. Elle a donné lieu aux publications suivantes:

- International Conference on Conceptual Structure, ICCS'2003: Thomopoulos *et al.* (2003*b*),
- Conférence nationale Extraction et Gestion des Connaissances, EGC'04: Thomopoulos *et al.* (2004),
- Revue IEEE Transactions on Fuzzy Systems: Buche *et al.* (2005*a*),
- Revue Fuzzy Sets and Systems: Buche *et al.* (2006*b*)[1],
- Revue IEEE Transactions on Knowledge and Data Engineering: Thomopoulos *et al.* (2006).

Nous commençons ce chapitre en présentant un état de l'art dans le domaine de l'interrogation flexible de bases de données. Nous considérons en effet que les techniques d'interrogation flexible représentent une solution pertinente à l'incomplétude d'une base de données dans la mesure où elles permettent de pallier au manque de données en proposant à l'utilisateur, en complément des réponses exactes, des réponses qui sont proches sémantiquement. Puis, afin d'introduire les concepts nécessaires à la compréhension de notre exposé, nous faisons dans la section 3.3, une introduction élémentaire à la théorie des sous-ensembles flous. Nous rappelons dans la section 3.4 les concepts empruntés à la théorie des sous-ensembles flous mis en œuvre dans le contexte des bases de données. Dans la section 3.5, nous étudions la notion de sous-ensemble flou défini sur un domaine de valeurs hiérarchisé. Nous présentons dans la section 3.6 le langage de requête de notre système d'intégration de données. La section 3.7 décrit brève-

[1] déjà citée dans le chapitre 2

ment l'interface graphique qui permet de définir des requêtes MIEL et présente des résultats expérimentaux concernant la mise en œuvre de la notion de sous-ensemble flou défini sur un domaine de valeurs hiérarchisé dans notre système d'intégration de données.

3.2 Interrogation de bases de données incomplètes

L'hypothèse implicite effectuée généralement dans les systèmes de base de données est l'hypothèse du monde clos (Closed World Assumption). On considère en effet que si une information n'est pas présente dans la base, c'est que cette information est fausse. Par exemple, supposons qu'une base de données relationnelle contienne les informations suivantes: le lait entier est contaminé par Listeria et Salmonella, de plus le lait écrémé est contaminé par Escherichia Coli. La requête "quels sont les contaminants qui ne sont pas présents dans le lait entier ?" renvoie Escherichia Coli. Cette hypothèse est gênante car souvent dans les applications réelles, il est impossible de rassembler dans la base de données toute l'information disponible sur le domaine concerné. Nous l'avons d'ailleurs illustré dans l'introduction en présentant les caractéristiques des données dans le domaine du risque alimentaire. Il est donc important de pouvoir "relâcher" l'hypothèse du monde clos afin de pouvoir considérer que l'absence de réponse ne veut pas dire que la réponse est négative, mais plutôt qu'on ne la connait pas: on fait alors l'hypothèse du monde ouvert (Open World Assumption). Cela revient à considérer qu'une base de données peut être incomplète et que certaines questions risquent de rester sans réponse. Pour pallier cet inconvénient, on peut chercher à proposer à l'utilisateur:

- des outils d'interrogation de la base lui permettant d'accéder à des données sémantiquement proches,
- des modèles d'estimation des données manquantes paramétrés avec des données sémantiquement proches présentes dans la base.

La première proposition, qui est celle à laquelle nous nous sommes intéressés dans ce mémoire, a été étudiée à notre connaissance de deux manières différentes dans la littérature: par l'expression de préférences dans les critères de sélection et par la généralisation des critères de sélection. Ces mécanismes permettent de compléter une réponse exacte peu informative, voire vide avec des réponses sémantiquement proches, jugées pertinentes.

Dans la première famille d'approche, le système d'interrogation ne cherche pas à vérifier si une donnée de la base vérifie un critère de sélection, mais plutôt dans quelle mesure elle satisfait ce critère, ce qui induit un ordonnancement des réponses. Trois catégories de travaux ont tout d'abord été réalisées pour résoudre ce problème (voir Bosc & Pivert 1992): l'utilisation de critères secondaires dans Lacroix & Lavency (1987), l'utilisation de distances de similarité (Ichikawa & Hirakawa 1986; Motro 1988) et l'expression de préférences linguistiques dans Rabitti & Savino (1990). Il a été montré (Bosc & Pivert 1992; Bosc *et al.* 1994) que l'ensemble de ces propositions peuvent être exprimées dans un formalisme commun: l'expression de préférences par des sous-ensembles flous. Ce formalisme permet à l'utilisateur de différencier les valeurs

idéales des valeurs acceptables pour un critère donné. A chacune des réponses correspondant à sa requête est associé un degré de pertinence qui mesure le degré d'adéquation de la réponse aux critères de sélection flous de la requête.

Dans la deuxième famille d'approche, l'idée est de reformuler la requête en la généralisant (Motro 1984). De cette manière, on obtient non seulement les réponses exactes à la requête, mais également d'autres réponses pertinentes. Dans une première catégorie de travaux, une hiérarchie de spécialisation de concepts est utilisée pour généraliser la requête en cas de réponse vide (voir par exemple Fargues 1989; Bidault *et al.* 2000). Dans une seconde catégorie de travaux, lorsque le critère de sélection est exprimé par un sous-ensemble flou, plusieurs techniques ont été proposées pour le généraliser. Dubois & Prade (1995) propose d'utiliser une relation de similarité sur le domaine de valeurs. Dans le cas où le sous-ensemble flou est défini sur un support à valeur numérique, Bosc *et al.* (2004*a*) propose un opérateur de généralisation flou utilisant une relation de proximité entre deux valeurs calculée à partir du quotient de celles-ci.

Guidés par notre domaine d'application dans lequel le vocabulaire employé est structuré en taxonomie, nous avons proposé de faire converger ces deux familles d'approches complémentaires dans le cas où les sous-ensembles flous sont définis sur un domaine de valeurs symboliques hiérarchisé.

3.3 Introduction élémentaire à la théorie des sous-ensembles flous

Préambule: Les rappels évoqués dans cette section sont en partie extraits de la synthèse récente publiée dans Bosc *et al.* (2004*b*).

Les sous-ensembles flous proposés par L. A. Zadeh (Zadeh 1965) sont une extension de la théorie des ensembles dont l'objectif principal est de représenter des ensembles d'objets pour lesquels l'appartenance d'un objet à l'ensemble n'est pas booléenne mais graduelle. Par exemple, si l'on désire représenter l'ensemble des résultats expérimentaux obtenus à *température de conservation*, l'ensemble ordinaire (dont la fonction caractéristique est représentée dans la figure 3.1) proposé pour traduire le concept de température de conservation, définit des frontières brutales qui représentent mal la réalité. En effet, seuls les résultats dont la valeur de température est comprise dans l'intervalle $[4, 8]$ font partie de l'ensemble. Dans la réalité, il est tout à fait possible que la température s'écarte légérement de cet intervalle de référence. Le sous-ensemble flou, dont la fonction d'appartenance est représentée dans la figure 3.2, prend mieux en compte cette réalité. Les valeurs de température dans l'intervalle $[4, 8]$ sont les plus plausibles, sans écarter les valeurs comprises entre 0 et 4 degrés d'une part et entre 8 et 12 degrés d'autre part. Le degré d'appartenance de ces valeurs au sous-ensemble flou décroît d'autant plus qu'elles s'éloignent de l'intervalle de référence $[4, 8]$.

Définition 1 Un *sous-ensemble flou* A ayant pour domaine de définition Dom est défini par sa

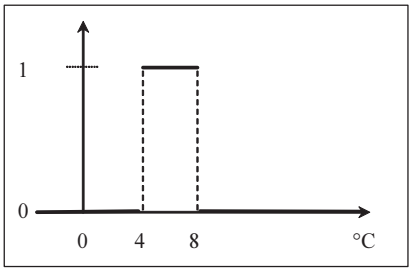


Fig. 3.1 – Exemple d'ensemble ordinaire représentant une température de conservation.

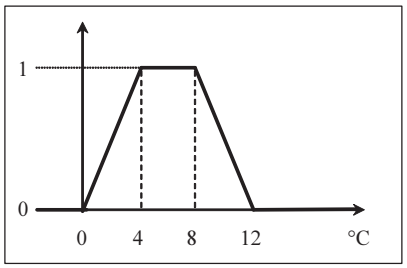


Fig. 3.2 – Exemple de sous-ensemble flou représentant une température de conservation.

fonction d'appartenance μ_A de Dom vers $[0, 1]$ qui associe à tout élément x de Dom son degré d'appartenance à A.

Le support et le noyau d'un sous-ensemble flou sont deux ensembles ordinaires définis de la manière suivante:

Définition 2 Soit un *sous-ensemble flou* A défini par sa fonction d'appartenance μ_A de Dom vers $[0, 1]$, le support de A, noté $supp(A)$, est défini par $supp(A) = \{x \mid x \in Dom \text{ et } \mu_A(x) > 0\}$.

Définition 3 Soit un *sous-ensemble flou* A défini par sa fonction d'appartenance μ_A de Dom vers $[0, 1]$, le noyau de A, noté $noy(A)$, est défini par $noy(A) = \{x \mid x \in Dom \text{ et } \mu_A(x) = 1\}$.

La hauteur d'un sous-ensemble flou A est définie par le degré de l'élément d'appartenance maximale à A. On dit que le sous-ensemble flou A est normalisé si sa hauteur vaut 1.

Par extension de la définition de la cardinalité d'un ensemble ordinaire, la cardinalité d'un sous-ensemble flou A ayant pour domaine de définition Dom est définie par : $\Sigma_{x \in Dom} \mu_A(x)$.

En théorie des ensembles, deux types de comparaison sont souvent utilisés: l'inclusion et l'égalité d'ensembles. On peut se demander ce que deviennent ces deux types de comparaison lorsque les ensembles sont flous. Deux démarches sont possibles: la réponse peut être définie de manière booléenne comme en théorie des ensembles, mais aussi de manière graduelle.

En théorie des ensembles, l'inclusion de l'ensemble A dans l'ensemble B peut se définir en terme de contrainte entre les valeurs des fonctions caractéristiques des ensembles A et B: $(A \sqsubseteq B) \Leftrightarrow (\forall x \in Dom, (f_A(x) \leq f_B(x)))$. Cette définition peut être étendue aux sous-ensembles flous.

Définition 4 Soient deux *sous-ensembles flous* A et B ayant pour domaine de définition Dom et pour fonction d'appartenance μ_A et μ_B , $(A \sqsubseteq B) \Leftrightarrow (\forall x \in Dom, (\mu_A(x) \leq \mu_B(x)))$.

Cette définition de l'inclusion entre sous-ensembles flous est booléenne et ne permet pas de distinguer, lorsque la réponse est négative, le cas où la condition d'inclusion est "presque" vérifiée par rapport au cas où elle ne l'est pas du tout. La notion d'inclusion graduelle permet de distinguer des situations différentes conduisant à une réponse négative. En prenant comme point de départ, la définition suivante de l'inclusion en théorie des ensembles:

$$(A \sqsubseteq B) \Leftrightarrow ((A \cap B) = A) \Leftrightarrow (card(A \cap B) = card(A)) \Leftrightarrow \frac{card(A \cap B)}{card(A)} = 1$$

on peut donner la définition graduelle suivante de l'inclusion entre sous-ensembles flous:

Définition 5 Soient deux *sous-ensembles flous* A et B ayant pour domaine de définition Dom et pour fonction d'appartenance μ_A et μ_B , soit T une norme, le degré d'inclusion de A dans B, noté $deg(A \sqsubseteq B)$, est donné par:

$$deg(A \sqsubseteq B) = \frac{card(A \cap B)}{card(A)} = \frac{\Sigma_{x \in Dom} T(\mu_A(x), \mu_B(x))}{\Sigma_{x \in Dom} \mu_A(x)}$$

La notion de norme permet dans la théorie de la logique floue d'étendre la notion d'intersection. Si l'on choisit, dans la formule précédente, l'opérateur minimum comme norme, la fonction d'appartenance μ_A joue un rôle de seuil.

L'égalité des ensembles ordinaires s'exprime habituellement par l'égalité des fonctions caractéristiques ou par la double inclusion. L'égalité de deux sous-ensembles flous peut être définie de manière booléenne comme suit:

Définition 6 Soient deux *sous-ensembles flous* A et B ayant pour domaine de définition Dom et pour fonction d'appartenance μ_A et μ_B , $(A = B) \Leftrightarrow (\forall x \in Dom, (\mu_A(x) = \mu_B(x)))$.

La notion de degré d'égalité de deux sous-ensembles flous peut être déduite de la définition 5.

Définition 7 Soient deux *sous-ensembles flous* A et B ayant pour domaine de définition Dom et pour fonction d'appartenance μ_A et μ_B , soit T une norme, le degré d'égalité de A et B, noté $deg(A = B)$, est donné par: $deg(A = B) = T(deg(A \sqsubseteq B), deg(B \sqsubseteq A))$.

3.4 Sous-ensembles flous dans le contexte des bases de données

Préambule: Les rappels évoqués dans cette section sont en partie extraits de la synthèse récente publiée dans Bosc *et al.* (2004*b*).

Dans le contexte des bases de données, les sous-ensembles flous ont été tout d'abord utilisés pour représenter des critères de sélection flous dans le langage d'interrogation. Les premiers travaux réalisés dans ce domaine sont ceux de Tahani (Tahani 1977). Un critère de sélection flou est défini par une expression de la forme ($attribut \approx valeurFloue$), $valeurFloue$ étant un sous-ensemble flou exprimant une disjonction priorisée de préférences. Le critère de sélection flou est évalué pour chaque donnée stockée en base ayant une valeur pour cet attribut (elle-même définie par un ensemble précis ou flou) et prend une valeur de vérité dans l'intervalle unité $[0, 1]$ traduisant l'adéquation de la donnée au critère de sélection flou. Plusieurs constructeurs permettant d'exprimer des conditions vagues à partir de critères de sélection flous ont été proposés: la négation, les modificateurs, les connecteurs flous et les propositions quantifiées floues.

La négation est obtenue en calculant le complément à 1 de la fonction d'appartenance du sous-ensemble flou. Un modificateur (voir Bouchon-Meunier & Yao 1992) est une fonction de $[0, 1]$ dans $[0, 1]$ qui s'applique à la fonction d'appartenance du sous-ensemble flou. Il permet de modéliser un adverbe du langage naturel comme "très", "plus ou moins", etc. Par exemple, la fonction $\mu_{modP}(x) = (\mu_P(x))^n$ permet d'obtenir un effet de concentration du sous-ensemble flou P si $n > 1$ et de dilatation si $n < 1$.

Les connecteurs flous permettent de modéliser un compromis entre plusieurs critères de sélection flous. L'agrégation conjonctive (resp. disjonctive) est généralement obtenue en appliquant l'opérateur minimum (resp. maximum) à la liste des valeurs de vérité obtenues pour chaque critère de sélection flou. L'utilisation du minimum (resp. maximum) pour interpréter la conjonction (resp. disjonction) revient à privilégier le critère de sélection flou qui est le moins (resp. le plus) satisfait pour représenter la satisfaction globale de l'ensemble des critères de sélection. Ce choix a le mérite d'être simple mais peut être discutable dans certaines applications. Deux types de solutions ont été proposées pour pallier cet inconvénient. Premièrement, des connecteurs plus sophistiqués permettent de pondérer l'importance des critères de sélection flous dans l'agrégation, de manière statique (Dubois & Prade 1986) ou dynamique (Yager 1988). Deuxièmement, les propositions quantifiées floues (Kacprzyk & Ziolkowski 1986) permettent d'introduire un deuxième niveau d'assouplissement dans la requête (si l'on considère que l'utilisation des critères de sélection flous correspond au premier niveau). Elles combinent les critères de sélection flous avec des quantificateurs flous comme "presque tous", "la plupart", ce qui permet de ne satisfaire qu'une partie des critères de sélection flous de la requête.

Le langage de requête MIEL, qui sera présenté dans la section 3.6, permet l'expression de critères de sélection flous. Une requête du langage MIEL permettant uniquement d'exprimer

une conjonction de critères de sélection flous, l'agrégation est effectuée en utilisant l'opérateur minimum. Nous aurions pu envisager d'utiliser des constructeurs plus sophistiqués, comme ceux présentés ci-dessus. Mais notre choix actuel satisfait pleinement les utilisateurs de notre système d'intégration de données. Nous n'avons donc pas ressenti le besoin, pour l'instant, d'enrichir le langage de requête avec des opérateurs plus sophistiqués.

3.5 Sous-ensembles flous hiérarchiques (SEFH)

Nous avons vu dans la section précédente que les sous-ensembles flous permettent d'exprimer des critères de sélection flexibles en associant aux valeurs recherchées des degrés de préférence. Nous verrons dans le chapitre 4 que les distributions de possibilités (Zadeh 1978), qui peuvent se représenter par des sous-ensembles flous normalisés, permettent de modéliser des données mal connues dans le cadre des bases de données. Ces deux approches ont en commun la définition d'une relation d'ordre (ordre de préférence ou ordre de possibilité) sur le domaine de valeurs du sous-ensemble flou. Dans cette section, nous nous intéressons au cas où le domaine de valeurs du sous-ensemble flou est organisé en taxonomie. Nous appellerons par la suite ce type de domaine, une *hiérarchie*. Elle introduit une deuxième relation d'ordre (partiel) définie par la relation "sorte de" entre les valeurs du domaine. A la différence d'un sous-ensemble flou défini sur un domaine de valeurs "plat" (non organisé en taxonomie), l'hypothèse implicite d'indépendance des valeurs entre elles n'est plus vraie. Par exemple, on ne peut plus supposer qu'une préférence associée à une valeur de la hiérarchie n'a pas d'implication sur les degrés associés à d'autres valeurs, particulièrement les valeurs comparables (plus spécifiques ou plus générales). Par voie de conséquence, les deux relations d'ordre évoquées ci-dessus doivent être mises en adéquation. Plusieurs questions se posent: quelle signification apporte-t-on au fait que deux valeurs comparables, au sens de la relation "sorte de", ont des degrés d'appartenance différents au sous-ensemble flou ? Dans le cas où le sous-ensemble flou modélise des préférences d'interrogation, peut-on utiliser la relation "sorte de" pour élargir le critère de sélection tout en respectant les ordres de préférence définis par l'utilisateur afin d'obtenir un plus grand nombre de réponses pertinentes ?

Dans la littérature, nous avons identifié deux catégories de travaux dans le domaine du flou ayant des similitudes avec nos préoccupations:

- dans les ontologies possibilistes (Loiseau *et al.* 2005), les termes de l'ontologie ont une description floue. Des comparaisons floues entre ontologies sont réalisées en utilisant ces descriptions floues. Ces travaux sont proches de ceux visant à définir des attributs à valeur floue dans le modèle objet (Rossazza *et al.* 1998).
- dans les pseudo-thesaurii flous (Miyamoto & Nakayama 1986; De Cock *et al.* 2004), des relations floues sont définies entre les termes. Elles représentent des relations de similarité entre termes, calculées à partir des fréquences de co-occurrence des termes dans un ensemble de documents de référence. Ces relations floues permettent d'élargir une requête

exprimée à partir des termes d'un thesaurus aux termes les plus proches.

Dans notre approche, ni les termes de la taxonomie, ni les relations de spécialisation entre ces termes ne sont flous. Nous n'avons donc pas pu nous inspirer des travaux antérieurs pour résoudre les questions que nous nous sommes posées ci-dessus. Dans cette section, nous commençons par définir dans la sous-section 3.5.1 la notion de sous-ensemble flou hiérarchique (noté SEFH par la suite). Puis, dans la sous-section 3.5.2, nous expliquons pourquoi et comment nous calculons la fermeture d'un SEFH. Nous étendons ensuite, dans la sous-section 3.5.3, les opérateurs de comparaison d'ensembles flous présentés dans la section 3.3 aux SEFH. Nous montrons dans la sous-section 3.5.4 que la notion de fermeture permet de regrouper les SEFH en classes d'équivalence et que chacune de ces classes a un représentant unique dit *minimal*. Enfin nous proposons dans la sous-section 3.5.5 une méthode de généralisation d'un SEFH basée sur le SEFH minimal.

3.5.1 Notion de SEFH

Lorsque le domaine de valeurs d'un attribut est hiérarchisé, si l'utilisateur veut définir un SEFH exprimant ses préférences d'interrogation, alors il le définit toujours sur un sous-ensemble du domaine de valeurs[2]. En effet, il ne choisit que les termes qui l'intéressent et considère implicitement d'une part que tous les termes plus spécifiques que ceux qu'il a choisis doivent être pris en compte et d'autre part que les autres termes ne doivent pas être pris en compte (par exemple, les non comparables). Par la suite, on dit qu'un terme t du domaine est plus général qu'un autre terme t' (noté $t' \leq t$) si t' est un prédécesseur de t dans l'ordre partiel induit par la relation "sorte de". Un exemple de hiérarchie est donné dans la figure 3.3.

Un SEFH est défini de la manière suivante:

Définition 8 Un SEFH est un sous-ensemble flou dont le domaine de définition est un sous-ensemble de sa hiérarchie de référence.

Exemple 1 *Le SEFH défini dans la figure 3.4 a pour domaine de définition l'ensemble des termes = {Lait entier, lait demi-écrémé, lait écrémé} qui est un sous-ensemble de la hiérarchie présentée dans la figure 3.3. Ce SEFH peut également être noté 1.0/Lait entier+0.9/Lait demi-écrémé+0.8/Lait écrémé.*

Aucune restriction n'est imposée sur les termes qui peuvent apparaître dans un SEFH. Notamment, des termes comparables peuvent figurer dans un SEFH avec des degrés différents. Soient d un degré associé à un terme t et d' un degré associé à un terme t' tel que t' est plus spécifique que t, $d' \leq d$ correspond à une sémantique de restriction de t' par rapport à t alors que $d' \geq d$ correspond à une sémantique de renforcement de t' par rapport à t.

[2] le domaine de valeurs est appelé par la suite hiérarchie de référence ou plus simplement hiérarchie

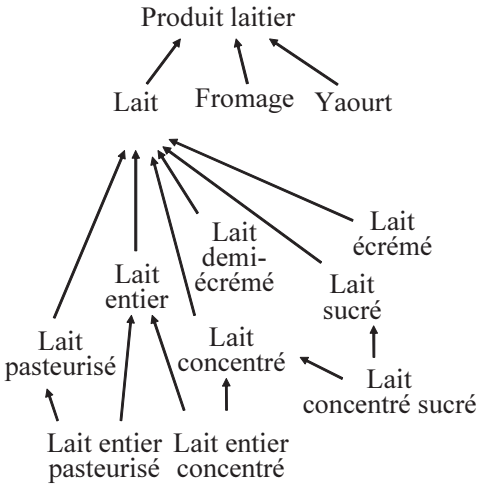


Fig. 3.3 – Un exemple de hiérarchie de référence pour les produits alimentaires. Les différentes valeurs sont reliées par la relation "sorte de".

Exemple 2 *Par exemple, si l'utilisateur est intéressé par les propriétés des laits à faible teneur en matière grasse, mais qu'il considère également comme pertinent de l'information sur les autres sortes de lait, il pourra l'exprimer par le SEFH: 1.0/Lait écrémé+0.5/Lait. Dans cet exemple, Lait écrémé a un degré supérieur à celui de Lait, ce qui s'interprète comme une sémantique de renforcement de Lait écrémé par rapport à Lait. A l'opposé, si l'utilisateur est intéressé par tous les types de laits sauf les laits concentrés à cause de leur plus faible teneur en eau, il l'exprimera par: 1.0/Lait +0.3/Lait concentré. Le degré associé à Lait concentré est inférieur à celui associé au terme plus général Lait, ce qui correspond à une sémantique de restriction.*

3.5.2 Fermeture d'un SEFH

Deux remarques peuvent être faites sur les SEFH:

- la première est de nature sémantique. Supposons que le SEFH *1.0/Lait écrémé+0.5/Lait* représente des préférences dans une requête. On peut noter que ce SEFH donne implicitement des informations sur les autres termes de la hiérarchie. Par exemple, on déduira que l'utilisateur n'est pas intéressé par les fromages et les yaourts, même si le degré 0 ne leur a pas été explicitement associé. On peut également supposer que toutes les sortes de laits écrémés (stérilisés, pasteurisés, etc) intéressent l'utilisateur avec un degré de 1.
- la seconde est de nature opérationnelle. D'après la définition 8, deux SEFH ayant la même hiérarchie de référence n'ont pas forcément le même domaine de définition. Ceci pose un problème si l'on veut leur appliquer des opérateurs de comparaison de sous-ensembles

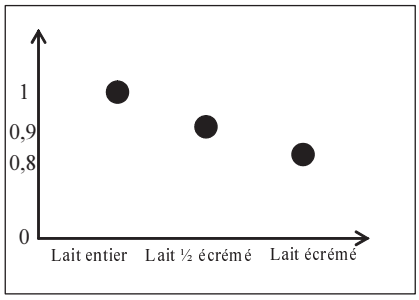


Fig. 3.4 – Un exemple de SEFH ayant pour domaine de définition un sous-ensemble de la hiérarchie présentée dans la figure 3.3.

flous comme, par exemple, ceux présentés dans les définitions 4, 5, 6 et 7.

Ces remarques nous amènent à la notion de fermeture d'un SEFH. Elle permet de définir un SEFH sur toute la hiérarchie de référence. Pour ce faire, nous proposons d'utiliser la relation "sorte de" afin de propager le degré associé à un terme appartenant au SEFH à tous les termes plus spécifiques dans la taxonomie. Par exemple, si le terme *Lait* appartient au SEFH, son degré est propagé à tous les termes plus spécifiques (*Lait entier, Lait pasteurisé, etc*). Par contre, on considère que les termes plus généraux ne doivent pas être considérés à cause des risques d'éloignement sémantique (par exemple: *Produits laitiers, Produits alimentaires, etc*).

Définition 9 Soient F un SEFH défini sur un sous-ensemble D d'une hiérarchie H et μ_F sa fonction d'appartenance définie sur D. La fermeture de F, notée $ferm(F)$, est un SEFH défini sur H dont la fonction d'appartenance est définie comme suit: pour tout terme t de H, soit $E_t = \{t_1, \ldots, t_n\}$ l'ensemble des plus petits super-termes de t dans D (i.e. $t_i \geq t$):

- si E_t est non vide: $\mu_{ferm(F)}(t) = max_{1 \leq i \leq n}(\mu_F(t_i))$
- sinon $\mu_{ferm(F)}(t) = 0$.

Cette définition de la fermeture d'un SEFH F vérifie les règles suivantes: pour tout terme t de H,

- si le terme t appartient à F, il conserve le même degré d'appartenance dans la fermeture (cas où $E_t = \{t\}$);
- si t a un unique plus petit super-terme t_1 qui appartient à F, le degré associé à t_1 dans F est propagé à t dans la fermeture de F (cas où $E_t = \{t_1\}$ avec $t_1 > t$);
- si t a plusieurs plus petits super-termes $\{t_1, \ldots, t_n\}$ qui appartiennent à F, avec des degrés différents, un choix doit être fait pour savoir quel degré doit être propagé à t. La proposition faite dans la définition 9 consiste à propager le maximum des degrés associés aux termes $\{t_1, \ldots, t_n\}$. Nous avons fait ce choix, car dans le contexte de notre système d'interrogation élargie, il permet d'augmenter potentiellement le nombre de réponses pour une requête spécifiée avec un seuil d'adéquation minimum différent de 0. Considérons par exemple le terme t qui a deux plus petits super-termes t_1 et t_2 ayant respectivement pour

degrés 0.5 et 0.3. Pour une requête ayant un seuil d'adéquation minimum de 0.4, le choix du maximum permet de récupérer les réponses correspondant au terme t.

– tous les autres termes de H, plus généraux ou non comparables aux termes de F sont considérés comme non pertinents. Le degré 0 leur est associé dans la fermeture de F (cas où $E_t = \emptyset$).

Exemple 3 *La figure 3.5 montre la fermeture du SEFH: 1.0/Lait entier+0.8/Lait+0.3/Lait concentré. Le degré 1.0 est associé à Lait entier concentré dans la fermeture, correspondant au maximum des degrés associés dans le SEFH à ses deux plus petits super-termes, Lait entier et Lait concentré. Le cas de Lait concentré sucré est différent: bien qu'étant comparable à deux termes auxquels l'utilisateur a associé un degré (0.8 pour le terme Lait, 0.3 pour Lait concentré), le seul plus petit super-terme de Lait concentré sucré appartenant au SEFH est Lait concentré. Par conséquent, le degré 0.3 lui est associé.*

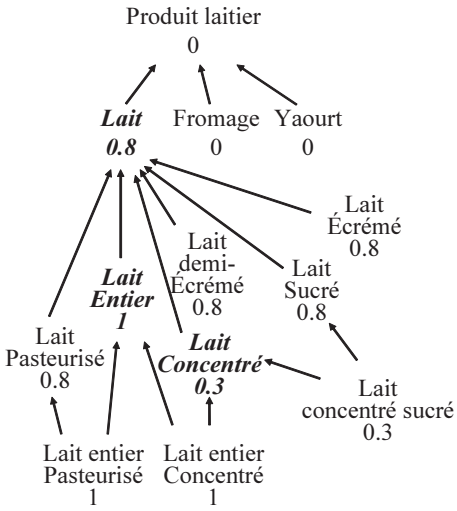


Fig. 3.5 – Fermeture du SEFH de l'exemple 3: les termes du SEFH et leurs degrés associés apparaissent en italique gras.

Il a été démontré dans Thomopoulos *et al.* (2006) que le calcul de la fermeture d'un SEFH F défini sur un domaine $Dom \subset H$ a une complexité en $\mid H \mid \times \mid Dom \mid^2$ sous l'hypothèse que la comparaison de deux termes de H peut être réalisée en temps constant. Comme en général le domaine de définition de F est limité à quelques valeurs, le temps de calcul de la fermeture de F est peu important. Le calcul de fermeture d'un SEFH a été implémenté dans le système d'intégration de données MIEL++. Le médiateur calcule les fermetures des SEFH d'une requête avant de la soumettre à chacun de ses sous-systèmes.

3.5.3 Comparaison de SEFH

L'introduction de la notion de fermeture permet d'étendre la définition d'un SEFH sur l'ensemble de la hiérarchie, quel que soit son domaine de définition. Cette notion permet de comparer deux SEFH en utilisant les opérateurs classiques de la logique floue.

Définition 10 Soient F_1 et F_2, deux SEFH ayant la même hiérarchie de référence H,
- $F_1 \subseteq F_2$ si $ferm(F_1) \subseteq ferm(F_2)$ au sens des inclusions des définitions 4 et 5;
- $F_1 = F_2$ si $ferm(F_1) = ferm(F_2)$ au sens des égalités des définitions 6 et 7.

Exemple 4 *Les fermetures des SEFH 0.2/Lait+1.0/Lait écrémé et 1.0/Lait+0.5/Lait concentré sont présentées dans la figure 3.6. Leur comparaison montre que 0.2/Lait + 1.0/Lait écrémé est inclus dans 1.0/Lait+0.5/Lait concentré au sens de la définition 4.*

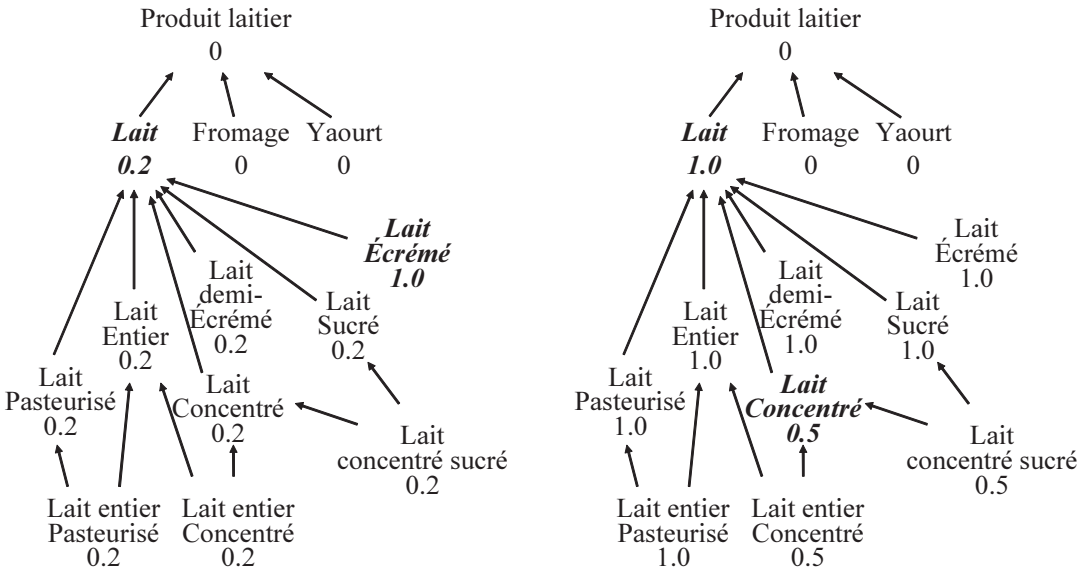


Fig. 3.6 – Fermetures des SEFH *1.0/Lait écrémé+0.2/Lait* et *1.0/Lait+0.5/Lait concentré*: les termes des SEFH et leurs degrés associés apparaissent en italique gras.

3.5.4 Notion de SEFH minimal

Nous avons vu dans la section précédente que tout SEFH a une fermeture définie sur la hiérarchie de référence. Nous nous intéressons dans cette section au fait que deux SEFH distincts ayant même hiérarchie de référence, peuvent avoir même fermeture. Considérons l'exemple suivant:

Exemple 5 *Les SEFH $Pref_1$=1.0/Lait et $Pref_2$=1.0/Lait+1.0/Lait écrémé ont la même fermeture.*

Comme le degré associé à *Lait écrémé* dans la fermeture de $Pref_1$ est le même que celui qui lui est associé dans $Pref_2$, on en conclut que le terme *Lait écrémé* est déductible dans $Pref_2$.

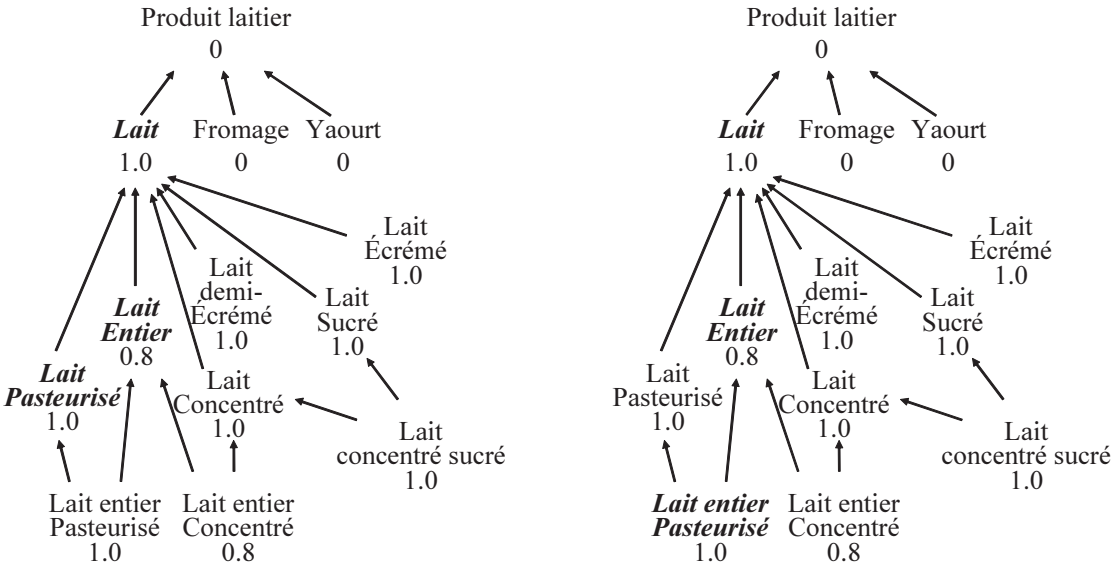


Fig. 3.7 – Fermeture commune aux SEFH $Pref_3$=*1.0/Lait+0.8/Lait entier+1.0/Lait pasteurisé* et $Pref_4$=*1.0/Lait+0.8/Lait entier+1.0/Lait entier pasteurisé*: les termes des SEFH et leurs degrés associés apparaissent en italique gras dans la hiérarchie de gauche pour $Pref_3$ et dans celle de droite pour $Pref_4$.

Définition 11 Soient F un SEFH défini sur $Dom(F) = \{t_1, \ldots, t_j, \ldots, t_n\}$ et $F{-}j$ le SEFH défini comme la restriction de F au domaine $Dom(F)\backslash\{t_j\}$, t_j est déductible de F si :

$$\mu_{ferm(F_{-j})}(t_j) = \mu_F(t_j)$$

A première vue, on pourrait penser que supprimer un terme déductible d'un SEFH permet d'éliminer une information redondante. Mais, l'élimination d'un terme déductible peut modifier sa fermeture. En effet, la définition 11 garantit que l'élimination d'un terme déductible n'a pas d'impact sur son degré d'appartenance dans la fermeture, mais par contre, elle peut en avoir sur les degrés associés dans la fermeture aux termes plus spécifiques que le terme déductible. Considérons l'exemple suivant:

Exemple 6 *Les SEFH $Pref_3$=1.0/Lait+0.8/Lait entier+1.0/Lait pasteurisé et $Pref_4$=1.0/Lait +0.8/Lait entier+1.0/Lait entier pasteurisé ont la même fermeture, présentée dans la figure 3.7. Dans $Pref_4$, aucun terme n'est déductible. Par contre, dans $Pref_3$ le terme Lait pasteurisé est déductible au sens de la définition 11. En le supprimant, on ne modifie pas son degré dans la fermeture, mais le degré du terme plus spécifique Lait entier pasteurisé est modifié (il passe de 1.0 à 0.8).*

Cette constatation nous amène à la définition d'un SEFH *minimal*.

Définition 12 Dans une classe d'équivalence donnée (c'est-à-dire, pour une fermeture donnée $FERM$), un SEFH F est dit *minimal* si $ferm(F) = FERM$ et si aucun des termes de son domaine de définition n'est déductible.

Exemple 7 *Les SEFH* $Pref_1$ *et* $Pref_4$ *sont minimaux, contrairement aux SEFH* $Pref_2$ *et* $Pref_3$.

Un algorithme de calcul d'un SEFH minimal à partir d'une fermeture $FERM$ a été proposé dans Thomopoulos *et al.* (2004). Il a été prouvé dans Thomopoulos *et al.* (2006) que le SEFH minimal pour une fermeture donnée est unique. Etant donnée une fermeture sur une hiérarchie de référence H, la complexité du calcul d'un SEFH minimal ayant pour domaine de définition Dom est en $\mid H \mid \times \mid Dom \mid^2$.

3.5.5 Généralisation d'un SEFH

Le fait d'utiliser des SEFH comme valeurs de sélection floues (cf section 3.4) exprimant des préférences dans une requête ne garantit pas d'obtenir un nombre suffisant de réponses. Il peut être intéressant de généraliser le SEFH représentant l'expression de préférences de l'utilisateur pour obtenir des réponses complémentaires pertinentes. Les approches proposées dans la bibliographie pour traiter le problème de la généralisation de sous-ensembles flous, au sens de l'inclusion (cf définition 4) ne sont pas bien adaptées aux SEFH. Certaines approches concernent uniquement les sous-ensembles flous définis sur un domaine de valeurs numériques (Bouchon-Meunier & Yao 1992; Bosc *et al.* 2004*a*). L'approche de Dubois & Prade (1995) permet de généraliser un sous-ensemble flou à partir de la définition d'une relation de similarité. Mais, elle s'adapte mal aux SEFH. Par exemple, un terme peut être ajouté au support du sous-ensemble flou généralisé alors que des termes plus spécifiques que celui-ci peuvent rester en dehors du support (cf Buche *et al.* 2005*a* pour plus de détails). Dans cette section, nous proposons une méthodologie de généralisation adaptée aux SEFH présentée dans Thomopoulos *et al.* (2006).

Nous commençons par définir une opération élémentaire de généralisation d'un SEFH. Puis, la notion de règle de généralisation qui permet de paramétrer la généralisation d'un SEFH selon plusieurs critères est introduite. Enfin, nous proposons une opération de généralisation qui consiste à appliquer plusieurs opérations élémentaires de généralisation de manière itérative.

Généralisation élémentaire d'un SEFH Une opération élémentaire de généralisation d'un SEFH F consiste à créer un SEFH F_g plus général que F au sens de l'opération d'inclusion booléenne étendue aux SEFH (cf définition 10).

Définition 13 Une généralisation élémentaire d'un SEFH F est une opération qui crée un SEFH F_g en ajoutant à $Dom(F)$ un super-terme d'un terme t de $Dom(F)$, noté t_g, et en lui

associant un degré donné d_g. Le terme t_g doit vérifier la condition suivante: soit t un terme de $Dom(F)$, t_g est un super-terme de t tel que $\nexists t' \in Dom(F)(t_g \leq t')$.

En d'autres termes, t_g ne doit, ni appartenir à $Dom(F)$, ni être plus spécifique qu'un terme appartenant à $Dom(F)$. Il a été prouvé dans Thomopoulos *et al.* (2006) que cette opération élémentaire de généralisation permet d'obtenir à partir d'un SEFH F un SEFH F_g plus général, au sens de la relation d'inclusion booléenne définie sur les SEFH (cf définition 10).

Exemple 8 *Soit le SEFH F = 1.0/Lait entier concentré + 0.5/Fromage, pour t = Lait entier concentré, on considère dans la hiérarchie H de la figure 3.3 le super-terme t_g= Lait et d_g= 0.2 pour effectuer l'opération élémentaire de généralisation. On obtient F_g= 1.0/Lait entier concentré + 0.5/Fromage + 0.2/Lait.*

Règle de généralisation d'un SEFH Nous considérons que la généralisation d'un SEFH F dépend essentiellement des 3 paramètres suivants: (i) le choix des termes de F à généraliser et l'ordre dans lequel ils sont pris en compte, (ii) le choix, pour un terme de F donné, des super-termes candidats à la généralisation, (iii) le degré qui leur est associé. Une règle de généralisation permet de fixer ces paramètres.

Définition 14 Une règle de généralisation R_g est définie par un triplet $(ord, gen, calc)$ où:
- *ord* est un ordre total de parcours des termes d'un SEFH F;
- *gen* est une application qui associe à tout terme de $Dom(F)$ un sous-ensemble de ses super-termes dans la hiérarchie;
- *calc* est une application qui associe un degré entre 0 et 1 à tout couple (t, t_g) tel que $t \in Dom(F)$ et $t_g \in gen(t)$.

Exemple 9 *Soit la règle de généralisation suivante:*
- *$gen(t)$ est l'ensemble des plus petits super-termes de t dans la hiérarchie H. Ce choix permet de minimiser le risque d'obtenir des réponses trop générales;*
- *ord est l'ordre des termes généralisables de F triés selon leur degré d'appartenance décroissant à F. Ce choix permet de généraliser en priorité les termes auxquels l'utilisateur a affecté les préférences les plus fortes;*
- *$calc(t, t') = min_{\{x \in Dom(F) | \mu_F(x) > 0\}} \mu_F(x) \times \mu_F(t) \times 0,9$ est le produit du plus petit degré non nul associé aux termes de F multiplié par le degré de t multiplié par 0,9. Ce choix permet d'obtenir en priorité les réponses concernant les termes spécifiés par l'utilisateur.*

Chaque terme de F n'a pas nécessairement un terme plus général qui peut être ajouté à F pour opérer une généralisation élémentaire. Comme nous l'avons vu dans la définition 13, ce terme plus général doit satisfaire une condition. Nous définissons maintenant la notion de terme généralisable de F selon une règle de généralisation donnée.

Définition 15 Soit le SEFH F, un terme $t \in Dom(F)$ est dit généralisable dans F, selon la règle de généralisation R_g, s'il existe un terme $t_g \in gen(t)$ tel que $\nexists t' \in Dom(F)(t_g \leq t')$.

Généralisation d'un SEFH Comme nous l'avons vu dans la section 3.5.2, le système d'intégration de données MIEL++ calcule la fermeture des SEFH d'une requête avant de la transmettre aux sous-systèmes. Ainsi, deux requêtes exécutées en utilisant deux SEFH différents, mais appartenant à la même classe d'équivalence, donnent la même réponse. Afin de préserver cette propriété lorsque l'on effectue une généralisation, celle-ci est opérée, non pas sur le SEFH directement, mais sur le SEFH minimal, représentant unique de la classe d'équivalence à laquelle le SEFH appartient.

Définition 16 La généralisation d'un SEFH F suivant une règle de généralisation R_g est une opération qui calcule un SEFH F_g obtenu de la manière suivante:

- La généralisation de degré 0 de F, notée F_0, est le SEFH minimal équivalent à F,
- Soit F_n, la généralisation de degré n de F:
  - s'il existe un terme t étant le premier terme (au sens de la relation ord) de $Dom(F_0) \subseteq Dom(F_n)$ généralisable dans F_n alors F_{n+1} est obtenu par une opération élémentaire de généralisation de F_n en ajoutant t_g à F_n où t_g est un plus petit super-terme de t dans $gen(t)$ vérifiant la condition de la définition 15 et $d_g = calc(t, t_g)$.
  - sinon, $F_g = F_n$.

Exemple 10 *Soit la règle de généralisation de l'exemple 9 et le SEFH F = 1.0/Lait entier + 1.0/Lait entier concentré + 0.8/Lait demi-écrémé + 0.2/Yaourt.*

- *le SEFH minimal équivalent à F est F_0 = 1.0/Lait entier + 0.8/Lait demi-écrémé + 0.2/Yaourt.*
- *le premier élément généralisable dans F_0, selon ord, est Lait entier. Lait est le plus petit super-terme de Lait entier, selon gen. Donc, F_1, généralisation élémentaire de F_0 = 1.0/Lait entier + 0.8/Lait demi-écrémé + 0.2/Yaourt + 0.18/Lait.*
- *le premier élément de F_0, généralisable dans F_1, selon ord, est Yaourt. Produits laitiers est le plus petit super-terme de Yaourt, selon gen. Donc, F_2 = 1.0/Lait entier + 0.8/Lait demi-écrémé + 0.2/Yaourt + 0.18/Lait + 0.0324/Produits laitiers.*
- *Il n'y a pas de terme dans F_0 généralisable dans F_2: $F_g = F_2$.*

Il a été prouvé dans Thomopoulos *et al.* (2006) que le nombre d'itérations est fini et que la généralisation obtenue est bien plus générale au sens de l'inclusion des SEFH.

3.6 Langage de requête MIEL

Le langage de requête MIEL repose sur 3 concepts: les concepts d'attribut, de vue et de requête.

Attribut: Un attribut du langage de requête MIEL est un attribut au sens habituel des bases de données. Il modélise une information élémentaire présente dans le système d'intégration de données. Il est défini sur un domaine de valeurs qui peut être de trois types:

– numérique: le domaine de valeurs est un sous-ensemble de $\mathbb{R}$,
– symbolique: le domaine de valeurs est un ensemble fini de constantes symboliques,
– symbolique hiérarchisé: le domaine de valeurs est un ensemble fini de constantes symboliques hiérarchisées selon la relation de spécialisation "sorte de".

Par exemple, l'attribut *Température* est de type numérique, l'attribut *Auteur* est de type symbolique et l'attribut *Produit* est de type symbolique hiérarchisé. Une partie du domaine de valeurs de l'attribut *Produit* est présentée dans la figure 3.8.

Remarque 1 *La notion de domaine de valeurs hiérarchisées est proche de la notion d'ordre partiel défini sur le domaine de valeurs d'un attribut dans Ginsburg & Hull 1983.*

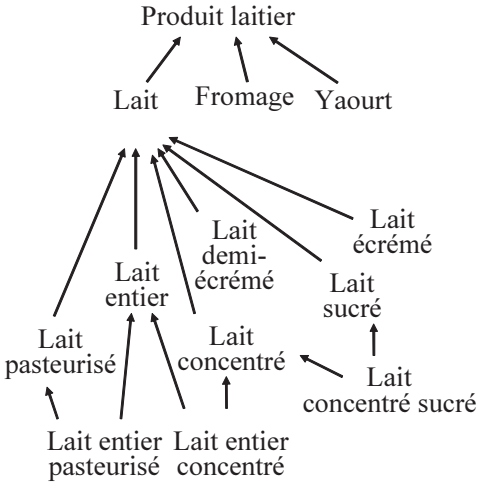


Fig. 3.8 – Une partie du domaine de valeurs de l'attribut *Produit*. Les différentes valeurs sont reliées par la relation "sorte de".

Remarque 2 *On suppose dans ce chapitre que la valeur associée à un attribut a, accessible par notre système d'intégration de données, notée $\tau(a)$, est une valeur atomique appartenant au domaine de valeurs de l'attribut. On verra dans le chapitre 4 comment est représentée une valeur stockée dans la base de données lorsque la donnée est imprécise.*

Vue: Une vue dans une base de données relationnelle est une relation virtuelle qui regroupe plusieurs relations dans lesquelles sont effectivement réparties les données. Elle permet de masquer la complexité du schéma relationnel à l'utilisateur qui veut interroger la base. Au niveau du schéma d'intégration, cette vue est définie par son nom et un ensemble d'attributs *interrogeables* qui constituent sa signature. Un attribut interrogeable peut être utilisé comme attribut

de sélection et/ou comme attribut de projection, au sens des bases de données relationnelles. Nous définissons ci-dessous la notion de vue dans un formalisme proche du calcul relationnel (Ullman 1988).

Définition 17 Une vue V ayant n attributs interrogeables $a_1, \ldots, a_n$ est définie par $V = \{a_1, \ldots, a_n | P_V(a_1, \ldots, a_n)\}$ où P_V est un prédicat qui caractérise la construction de la vue V dans chaque sous-système.

Exemple 11 *La vue ExpérienceUnFacteur comporte 5 attributs interrogeables:* $ExperienceUnFacteur = \{Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse | P_{ExperienceUnFacteur}(Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse)\}$. *Cette vue permet d'obtenir des résultats concernant des expériences dans lesquelles un seul facteur est contrôlé (par exemple: la température). Le type de réponse obtenu peut être, par exemple, une cinétique de la contamination d'un produit par un microorganisme donné ou une vitesse de croissance d'un microorganisme dans un produit donné.*

Requête: Une requête MIEL est une spécialisation d'une vue dans laquelle l'utilisateur spécifie une liste d'attributs de projection et une liste d'attributs de sélection parmi l'ensemble des attributs interrogeables. La valeur associée à chaque attribut de sélection est un sous-ensemble flou qui exprime les préférences de l'utilisateur.

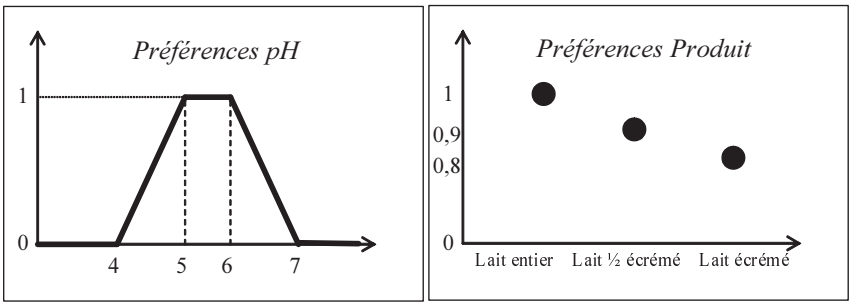


Fig. 3.9 – Sous-ensembles flous représentant les préférences d'interrogation de la requête de l'exemple 12.

Définition 18 Une requête Q exprimée dans une vue V ayant n attributs interrogeables $a_1, \ldots, a_n$ est définie par son degré de satisfaction minimum $\delta_{min} \in [0, 1]$ et par:

$$Q = \{a_1, \ldots, a_l | \exists a_{m+1}, \ldots, a_n (P_V(a_1, \ldots, a_n) \wedge (a_{l+1} \approx v_{l+1}) \wedge \ldots \wedge (a_m \approx v_m))\}$$

où $1 \leq l \leq m \leq n$, P_V est le prédicat qui caractérise la construction de la vue, $a_1, \ldots, a_l$ la liste des attributs de projection, $a_{m+1}, \ldots, a_n$ la liste des attributs interrogeables non utilisés dans la requête, $(a_{l+1} \approx v_{l+1}), \ldots, (a_m \approx v_m)$ la liste des critères de sélection flous dans lesquels $a_{l+1}, \ldots, a_m$ sont les attributs de sélection, et $v_{l+1}, \ldots, v_m$ la liste des sous-ensembles flous qui leur sont respectivement associés.

Exemple 12 *La requête Q est exprimée dans la vue ExpérienceUnFacteur : $Q = \{Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse | (P_{ExperienceUnFacteur}(Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse) \wedge (Produit \approx PreferencesProduit) \wedge (pH \approx PreferencespH))\}$. Les sous-ensembles flous $PreferencesProduit$ et $PreferencespH$ sont donnés dans la figure 3.9. Le degré de satisfaction minimum de la requête est fixé à $\delta_{min} = 0.8$.*

Réponse: La réponse à une requête est constituée d'un ensemble de tuples. Un tuple est une collection de valeurs atomiques, chacune de ces valeurs correspondant à un attribut de projection de la requête. Un degré d'adéquation aux critères de sélection de la requête est associé à ce tuple. Par conséquent, la réponse à une requête Q peut être formellement définie comme un sous-ensemble flou ayant pour domaine le produit cartésien des domaines de valeur associés aux attributs de projection de la requête.

Définition 19 La réponse A à la requête Q de la définition 18 est définie par $A = \{\tau_1, \ldots, \tau_r\}$, où τ_i, $i \in [1, r]$, est un tuple de la forme $\{\tau_i(a_1), \ldots, \tau_i(a_l)\}$. Chaque tuple τ_i de A satisfait les attributs de sélection de Q avec le degré $\delta_i \geq \delta_{min}$ tel que $\delta_i = min_{j=l+1,\ldots,m}\mu_{v_j}(\tau_i(a_j))$, $\tau_i(a_j)$ étant la valeur associée à l'attribut de sélection a_j dans le tuple τ_i et μ_{v_j} la fonction d'appartenance du sous-ensemble flou v_j.

Exemple 13 *Un exemple de réponse à la requête de l'exemple 12 formulée dans la vue ExpérienceUnFacteur est donné dans le tableau 3.1.*

TAB. 3.1 – Une partie de la réponse à la requête de l'exemple 12 formulée dans la vue *ExpérienceUnFacteur*

δ	Produit	Microorganisme	pH	Facteur	Type de réponse
1.0	Lait entier	Bacillus Cereus	5.1	Température	Cinétique de contamination
0.9	Lait demi-écrémé	Listeria	5.0	Température	Vitesse de croissance
0.8	Lait écrémé	Listeria	6.0	Température	Vitesse de croissance

3.7 Implémentation et expérimentation

Dans cette section, nous présentons les interfaces utilisateur du système d'intégration MIEL++ et nous donnons des résultats expérimentaux concernant les opérations de fermeture et de généralisation des SEFH.

Interfaces graphiques de notre système d'intégration Nous présentons quelques dialogues utilisateur à partir d'un exemple d'interrogation.

Fig. 3.10 – Interface graphique de saisie d'un SEFH.

L'utilisateur veut exécuter une requête dans la vue *ExpérienceUnFacteur* présentée dans l'exemple 11. Il dispose des attributs interrogeables suivants: *Produit, Microorganisme, pH, NomFacteurEtudie et TypeReponse*. Il choisit de spécifier ses préférences sur les aliments sous la forme du SEFH (1.0/Fromage à pâte molle+0.9/Fromage) associé à l'attribut Produit et sur leur valeur de pH sous la forme d'un sous-ensemble flou défini sur un support numérique (intervalle flou [4, 5, 6, 7]) associé à l'attribut de même nom. La figure 3.10 présente la fenêtre dans laquelle l'utilisateur saisit son SEFH. Il a accès à la hiérarchie H dans le cadre *Hiérarchie des aliments*. Il enregistre, dans des zones de saisie à listes déroulantes, ses préférences dans le cadre *Ensemble de valeurs aliment*. Les listes déroulantes permettent un accès à la hiérarchie par ordre alphabétique. Dans le cadre *Hiérarchie des aliments*, le bouton *Visualiser les choix* demande au système de colorer en rouge les termes de la hiérarchie faisant partie de la fermeture du SEFH et en violet les termes obtenus par l'opération de généralisation du SEFH. Cette dernière opération est déclenchée lorsque la case *Elargir la sélection* du cadre *Ensemble de valeurs aliment* est cochée.

La figure 3.11 présente la fenêtre de saisie d'un sous-ensemble flou à valeurs numériques, utilisée dans l'exemple pour définir l'intervalle flou sur les valeurs de pH autorisées.

Une partie de la réponse à la requête est présentée dans la figure 3.12. La première colonne contient le degré d'adéquation de la réponse (cf définition 19) à la requête. Ces réponses sont téléchargeables sur le poste client sous la forme d'un fichier Excel afin de permettre à l'utilisateur de les manipuler.

Expérimentation Nous avons mené une expérimentation pour évaluer la pertinence des calculs de fermeture et de généralisation des SEFH. Cette expérimentation a exclusivement porté sur la base de données relationnelle qui contient environ 10 000 données. Nous avons défini, avec

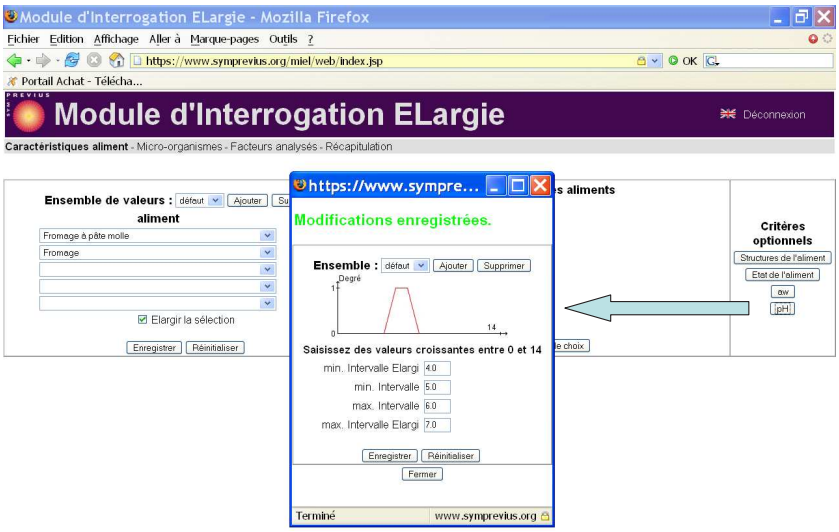
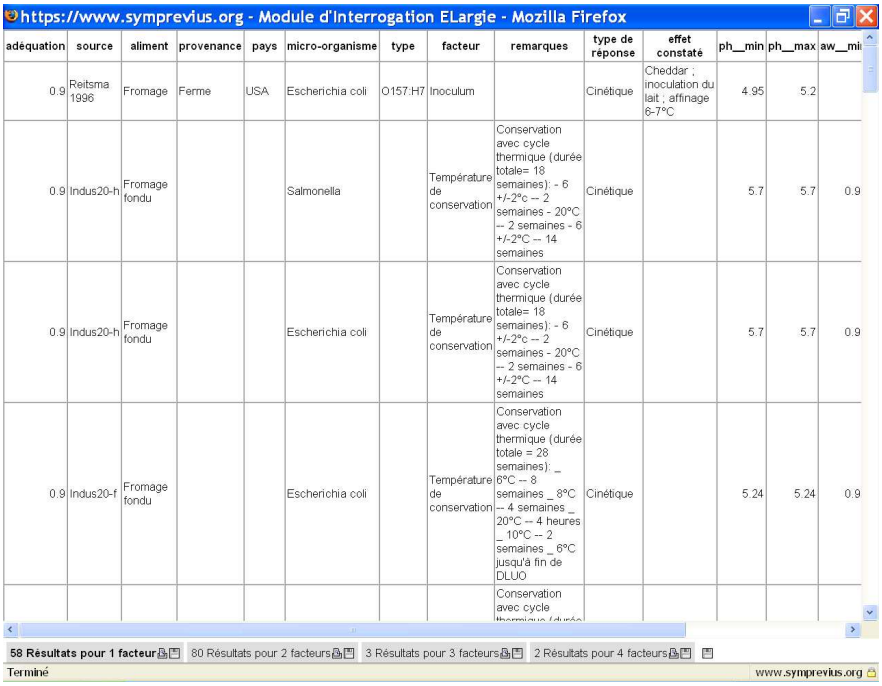


Fig. 3.11 – Interface graphique de saisie d'un sous-ensemble flou à support numérique.

les experts du domaine, 7 requêtes dont les résultats permettent de couvrir 10 % du nombre de données en base. Le tableau 3.2 présente les résultats obtenus pour chacune des 7 requêtes représentant au total 1132 réponses. La même requête a été exécutée trois fois: sans calcul de fermeture (requête standard), avec calcul de fermeture, avec calcul de fermeture et généralisation. Les réponses exactes sont les réponses obtenues soit par les requêtes standard, soit par les requêtes avec fermeture. Les réponses obtenues par le calcul de fermeture ont toutes été considérées pertinentes par les experts du domaine. Elles représentent 99 % des réponses exactes (1 % des réponses exactes sont obtenues par les requêtes standard), ce qui constitue un résultat excellent pour l'opération de fermeture. Parmi les résultats obtenus par généralisation, les réponses pertinentes (80 % du nombre total de réponses obtenues par généralisation) sont celles qui ont les plus forts degrés d'adéquation (compris entre 0.6 et 0.8) alors que les réponses non pertinentes ont des degrés d'adéquation variant entre 0.2 et 0.6. On peut donc considérer que les résultats d'évaluation de la méthode de généralisation sont également bons puisque (i) les résultats pertinents peuvent facilement être identifiés en utilisant un seuil sur le degré d'adéquation des réponses (0.6), (ii) les réponses pertinentes obtenues par généralisation constituent un nombre important de réponses complémentaires aux réponses exactes (56% du nombre total de réponses pertinentes).

3.8 Conclusion du chapitre

Nous avons présenté dans ce chapitre la problématique de l'interrogation de bases de données incomplètes. Nous avons rappelé deux types d'approches pour y faire face:(i) l'utilisation de critères de sélection flous pour exprimer des requêtes graduelles, (ii) l'utilisation de méthodes de généralisation de requêtes. Une première originalité de notre système d'intégration de données



adéquation	source	aliment	provenance	pays	micro-organisme	type	facteur	remarques	type de réponse	effet constaté	ph__min	ph__max	aw__mi
0.9	Reitsma 1996	Fromage	Ferme	USA	Escherichia coli	O157:H7	Inoculum		Cinétique	Cheddar ; inoculation du lait ; affinage 6-7°C	4.95	5.2	
0.9	Indus20-h	Fromage fondu			Salmonella		Température de conservation	Conservation avec cycle thermique (durée totale= 18 semaines): - 6 +/-2°c -- 2 semaines - 20°C -- 2 semaines - 6 +/-2°C -- 14 semaines	Cinétique		5.7	5.7	0.9
0.9	Indus20-h	Fromage fondu			Escherichia coli		Température de conservation	Conservation avec cycle thermique (durée totale= 18 semaines): - 6 +/-2°c -- 2 semaines - 20°C -- 2 semaines - 6 +/-2°C -- 14 semaines	Cinétique		5.7	5.7	0.9
0.9	Indus20-f	Fromage fondu			Escherichia coli		Température de conservation	Conservation avec cycle thermique (durée totale = 28 semaines): _ 6°C -- 8 semaines _ 8°C -- 4 semaines _ 20°C -- 4 heures _ 10°C -- 2 semaines _ 6°C jusqu'à fin de DLUO	Cinétique		5.24	5.24	0.9
								Conservation avec cycle					

Fig. 3.12 – Résultat d'une interrogation dans la vue *ExpérienceUnFacteur*

est de proposer l'utilisation de critères de sélection flous dans son langage d'interrogation. Cette proposition n'a, à notre connaissance, jamais été faite auparavant dans le cadre de la conception d'un système d'intégration de données. Dans un contexte où l'information accessible par le système d'intégration est relativement rare, l'utilisation de critères de sélection flous permet à l'utilisateur de gérer l'élargissement de ses critères de sélection. Nous avons illustré cette proposition à travers la présentation du langage de requête de notre système d'intégration de données.

Etant fortement influencés par un domaine d'application, le risque alimentaire, dans lequel l'information est structurée en taxonomies, nous avons présenté une deuxième originalité de notre système d'intégration de données, le concept de sous-ensemble flou défini sur une hiérarchie, appelé SEFH. Il permet de rendre compatibles les deux relations d'ordre définies dans ce type de sous-ensemble flou: celle qui représente la gradualité et celle qui modélise la relation de spécialisation entre les termes. Un SEFH est défini sur un sous-ensemble de la hiérarchie. La notion de fermeture permet de définir un SEFH sur l'ensemble de la hiérarchie en propageant les degrés grâce à la relation de spécialisation. Les SEFH ayant même fermeture peuvent être regroupés en classes d'équivalence. Chaque classe d'équivalence a un représentant unique, appelé SEFH minimal. La notion de SEFH minimal est utilisée par la méthode de généralisation des SEFH dont l'objectif est d'élargir les préférences que l'utilisateur exprime dans sa requête afin d'obtenir des réponses complémentaires pertinentes. Le concept de SEFH a été utilisé dans notre système d'intégration à la fois pour représenter des valeurs de sélection floues définis sur une hiérarchie, mais aussi pour modéliser des données imprécises, comme nous le verrons dans

TAB. 3.2 – Evaluation des opérations de fermeture et de généralisation

Critère de sélection	Nb réponses exactes avec une requête standard	Nb réponses exactes avec une fermeture	Nb réponses avec généralisation jugées pertinentes (avec leur degré)	Nb réponses avec généralisation jugées non pertinentes (avec leur degré)
Produit=1/Crustacés	0	4	66 (degré 0.8)	0
Produit=1/Fromages	5	152	267 (degré 0.8)	0
Produit=1/Fromages et Microorganisme=1/Listeria	0	53	87 (degré 0.8)	0
Produit=1/Oeuf	0	16	10 (degré 0.8)	87 (degrés $\in [0.2, 0.4]$)
Produit=1/Viandes salées et Microorganisme=1/Listeria	0	33	44 (degré 0.8)	63 (degrés $\in [0.4, 0.6]$)
Produit=1/Salade et Microorganisme=1/Listeria	0	17	25 (degré $\in [0.6, 0.8]$)	7 (degrés $\in [0.2, 0.6]$)
Produit=1/Viandes fraiches	0	217	136 (degré 0.8)	0

le chapitre 4.

Dans un avenir proche, nous étudierons la possibilité d'optimiser les algorithmes de comparaison des SEFH qui reposent actuellement sur l'opération de fermeture. Une solution envisagée consiste à considérer le SEFH minimal à la place de la fermeture.

Une autre perspective du travail présenté dans ce chapitre concerne l'extension du langage d'interrogation, volontairement simple car s'adressant à des utilisateurs non-informaticiens par l'introduction de critères de sélection flous et/ou de connecteurs flous plus complexes, comme ceux présentés dans la section 3.3.

Chapitre 4

Représentation de données imprécises

4.1 Introduction

Ce chapitre est consacré à la présentation des modèles de données utilisés pour représenter les données internes et externes de notre système d'intégration. Comme indiqué dans l'introduction de ce mémoire, notre domaine d'application nous a amenés à nous poser la question de la représentation de données imprécises dans ces modèles de données. L'extension du modèle des graphes conceptuels pour la représentation de données imprécises, présentée dans ce chapitre, a été réalisée dans le cadre de la thèse de Rallou Thomopoulos (Thomopoulos 2003), que j'ai co-encadrée avec Ollivier Haemmerlé. Elle a donné lieu aux publications suivantes:

- Conférence nationale RFIA (Reconnaissances des Formes et Intelligence Artificielle)": Thomopoulos *et al.* (2002),
- International Conference on Conceptual Structure, ICCS'2003: Thomopoulos *et al.* (2003*b*)[1],
- International Conference of the North American Fuzzy Information Processing Society (NAFIPS'01 et NAFIPS'03): Buche *et al.* (2001) et Thomopoulos *et al.* (2003*a*),
- Revue Fuzzy Sets and Systems: Thomopoulos *et al.* (2003*c*).

L'extension du modèle XML pour la représentation de données imprécises, également présentée dans ce chapitre, a été réalisée en collaboration avec Juliette Dibie-Barthélemy et Ollivier Haemmerlé. Son utilisation pour représenter des annotations sémantiques floues associées à des données externes provenant du Web a été étudiée dans le cadre de la thèse de Gaëlle Hignette que je co-encadre avec Ollivier Haemmerlé et Juliette Dibie-Barthélemy. Ce travail a donné lieu aux publications suivantes:

- International Conference Flexible Querying Answering Systems, FQAS'2004: Buche *et al.* (2004),
- International Conference on Advanced Information Systems Engineering, Workshop DisWeb'2005: Hignette *et al.* (2005),

[1] déjà citée dans le chapitre 3

– Journal of Intelligent Information Systems: Buche *et al.* (2006*a*).

Nous commençons ce chapitre par une présentation de l'état de l'art sur la représentation de données imprécises dans les bases de données. Dans la section 4.3, nous rappelons les principes de la théorie des possibilités et les raisons pour lesquelles nous nous sommes fondés sur cette théorie pour étendre nos modèles de données. Puis, comme l'utilisation de la théorie des possibilités a déjà été largement étudiée dans le cadre du modèle relationnel, nous centrons notre présentation sur trois propositions d'extension spécifiques à notre système d'intégration: (i) dans la section 4.4, celle concernant le langage d'interrogation de notre système d'intégration de données, (ii) dans la section 4.5, celle concernant le modèle des graphes conceptuels et (iii) dans la section 4.6.1, celle concernant le modèle de données XML. Dans la section 4.6.2, une utilisation du modèle de données XML étendu est proposée pour annoter sémantiquement des données externes provenant du Web.

4.2 Données imprécises et bases de données

La notion d'imprécision est relative à une information qui est connue avec certitude sans pour autant la connaître avec précision. Par exemple, le fait de savoir que la température de conservation d'un produit alimentaire à un instant donné est comprise entre 2 et 10 degrés Celsius est une information imprécise. Cette information est imprécise car la valeur exacte de la température est inconnue, cependant il est certain qu'elle appartient à l'intervalle [2,10] degrés Celsius. Le concept d'incertitude s'applique à une connaissance qui, comme son nom l'indique, ne peut pas être établie avec certitude. Par exemple, en s'appuyant sur l'information précédente concernant la température de conservation d'un produit alimentaire, la connaissance "la température est de 6 degrés" est incertaine car la seule certitude que l'on ait est qu'elle est comprise entre 2 et 10 degrés. Imprécision et incertitude sont liées car quand l'imprécision de la connaissance augmente, son incertitude diminue. Dans l'exemple précédent, affirmer que la température est à 6 degrés est très incertain (connaissance précise, incertitude forte), alors que l'on peut annoncer de manière certaine que la température est comprise entre 2 et 10 degrés.

Dans le contexte des bases de données, E. F. Codd (Codd 1979) a été l'un des premiers à s'intéresser à la prise en compte des données imprécises dans le cadre du modèle relationnel. Il a introduit le concept de *null value* qui représente la valeur d'un attribut qui est inconnue au moment où on interroge la base ou qui n'a pas de sens dans l'enregistrement où elle se trouve. Lipski (Lipski 1979, 1981) a prolongé l'approche de Codd qui était binaire (connaissance totale ou méconnaissance totale) afin de permettre l'expression de connaissances partielles. Il introduit la notion d'ensemble de valeurs vraisemblables sous la forme d'une disjonction exclusive de valeurs possibles. Par exemple, toutes les valeurs de température entre 2 et 10 degrés sont également possibles. La théorie des possibilités de Zadeh (Zadeh 1978) a été utilisée dans le cadre du modèle relationnel par Prade (1984) et Prade & Testemale (1984) pour compléter les

approches de Codd et de Lipski en introduisant une expression de classement sur les valeurs possibles. La théorie des possibilités permet d'exprimer par exemple que, parmi l'ensemble des valeurs possibles de température, les valeurs de l'intervalle [4, 8] degrés sont plus possibles que les valeurs dans les intervalles [2, 4 [et]8, 10].

4.3 Introduction élémentaire à la théorie des possibilités

Dans cette section, nous présentons les concepts nécessaires à la compréhension des propositions présentées dans les sections suivantes. Pour une présentation plus détaillée de la théorie des possibilités et de son utilisation dans le cadre des bases de données, le lecteur pourra se reporter à la synthèse récente publiée dans Bosc *et al.* (2004*b*).

Nous commençons cette introduction par une comparaison de la théorie des possibilités avec la théorie des probabilités. En effet, ces deux théories ont un objectif commun: mesurer la confiance que l'on peut avoir dans la réalisation d'un événement. Pour atteindre cet objectif, la théorie des probabilités utilise la fréquence de réalisation d'événements. Par voie de conséquence, elle permet d'ordonner des événements en fonction de leur fréquence d'apparition. La théorie des possibilités s'intéresse aux situations dans lesquelles les fréquences d'apparition des événements ne sont pas disponibles. Elle repose sur un ordonnancement, éventuellement subjectif, des événements.

Dans notre contexte applicatif (présenté dans l'introduction de ce mémoire), nous ne disposons quasiment jamais d'une information de nature fréquentielle concernant les données imprécises que nous avons à représenter. Par exemple, un niveau de contamination d'un produit alimentaire ou une vitesse de croissance d'un contaminant sont généralement déjà synthétisés dans les sources de données sous la forme d'un intervalle de valeurs. C'est la première raison qui nous a amenés à choisir le cadre de la théorie des possibilités pour représenter des données imprécises. Par ailleurs, comme nous le verrons dans la section 4.3.2, la théorie des possibilités permet de mesurer la réalisation d'un événement flou en la comparant à une donnée imprécise. Nous utilisons cette mesure dans notre système d'intégration, puisqu'un critère de sélection flou spécifié dans une requête MIEL peut s'exprimer comme un événement flou dans la théorie des possibilités. La théorie des possibilités offre donc un cadre homogène pour comparer les critères de sélection flous d'une requête MIEL aux données imprécises stockées dans notre système d'intégration. C'est la deuxième raison qui nous a amenés à choisir la théorie des possibilités pour étendre nos modèles de données.

Dans la sous-section 4.3.1, nous présentons la notion de mesure de possibilité d'un événement ordinaire (c'est à dire non flou) et la notion complémentaire de mesure de nécessité (ou mesure de certitude) de cet événement. Puis, dans la sous-section 4.3.2, les notions précédentes sont étendues pour pouvoir prendre en compte des événements flous.

4.3.1 Mesures de possibilité et de nécessité d'événements ordinaires

Une mesure de possibilité (Zadeh 1978) est une fonction ensembliste Π définie de $P(X)$, l'ensemble des parties d'un ensemble fini X, sur l'intervalle $[0, 1]$. Cette mesure vérifie les axiomes suivants:

- $\Pi(X) = 1$;
- $\Pi(\emptyset) = 0$;
- $\Pi(A \cup B) = max(\Pi(A), \Pi(B))$ avec A et B deux parties de X.

Si A est considéré comme un événement, $\Pi(A)$ est la mesure de possibilité de l'événement A. $\Pi(A) = 1$ signifie que l'événement est complètement possible. Comme $A \cup \bar{A} = X$, des premier et troisième axiomes, on déduit que $max(\Pi(A), \Pi(\bar{A})) = 1$. Cela veut dire que si un événement n'est pas complètement possible, alors son événement contraire est complètement possible. Dans la théorie des possibilités, un événement est constitué d'une disjonction d'événements élémentaires mutuellement exclusifs. Une mesure de possibilité s'exprime à partir d'une distribution de possibilités qui représente les valeurs de possibilités des événements élémentaires. Au moins un de ces événements élémentaires doit être complètement possible. Une distribution de possibilités est représentée par un sous-ensemble flou normalisé.

Définition 20 Une distribution de possibilités sur un ensemble X fini est une fonction π de X dans $[0, 1]$ tel que : $\exists x \in X$ avec $\pi(x) = 1$.

Définition 21 Soit π une distribution de possibilités définie sur un ensemble X. Une mesure de possibilité Π est définie de $P(X)$, l'ensemble des parties d'un ensemble fini X, sur l'intervalle $[0, 1]$ par $\forall A \in P(X), \Pi(A) = sup_{x \in A} \pi(x)$.

Exemple 14 *La température de conservation d'un produit alimentaire est donnée par la connaissance de la distribution de possibilités π définie sur l'ensemble des entiers naturels entre 2 et 10:* $\pi(2) = 0.2, \pi(3) = 0.5, \pi(4) = \pi(5) = \pi(6) = \pi(7) = \pi(8) = 1, \pi(9) = 0.5, \pi(10) = 0.2$. *Les valeurs entre 4 et 8 degrés sont complètement possibles. Puis, par ordre de possibilité décroissante, viennent les valeurs 3 et 9 degrés, puis 2 et 10 degrés. Les valeurs strictement inférieures à 2 et strictement supérieures à 10 sont complètement impossibles. La possibilité de l'événement "température comprise entre 2 et 7 degrés" est* $\Pi(2, 3, 4, 5, 6, 7) = max(\pi(2), \pi(3), \pi(4), \pi(5), \pi(6), \pi(7)) = 1$. *Cela signifie qu'il est entièrement possible que la température soit comprise entre 2 et 7 degrés. La possibilité de l'événement "température comprise entre 9 et 13 degrés" est* $\Pi(9, 10, 11, 12, 13) = max(\pi(9), \pi(10), \pi(11), \pi(12), \pi(13)) = 0.5$. *Cet événement est donc possible avec le degré de 0.5.*

Remarque 3 *La mesure de possibilité n'est pas suffisante car elle ne permet pas de discriminer deux situations extrêmes: les cas de l'ignorance totale* ($\Pi(A) = 1$ *et* $\Pi(\bar{A}) = 1$) *et de la certitude totale* ($\Pi(A) = 1$ *et* $\Pi(\bar{A}) = 0$) *d'un événement A. Dans les deux cas,* $\Pi(A)$ *vaut 1, l'événement A est complètement possible.*

Il est donc nécessaire de la compléter par une mesure de nécessité (ou de certitude) que l'événement soit réalisé. Elle est définie comme la mesure de l'impossibilité de l'événement opposé (Dubois & Prade 1980, Dubois & Prade 1985, Dubois & Prade 1988).

Définition 22 Une mesure de nécessité N est définie de $P(X)$, l'ensemble des parties d'un ensemble fini X, sur l'intervalle $[0, 1]$. $\forall A \in P(X), N(A) = 1 - \Pi(\bar{A})$.

A partir de la distribution de possibilités π, la mesure de nécessité est définie de la manière suivante:

Définition 23 Soit π une distribution de possibilités définie sur un ensemble X. Une mesure de nécessité N est définie de $P(X)$, l'ensemble des parties d'un ensemble fini X, sur l'intervalle $[0, 1]$. $\forall A \in P(X), N(A) = inf_{x \notin A}(1 - \pi(x))$.

Si $N(A) = 1$, alors par définition $\Pi(\bar{A}) = 0$: l'événement A est complètement certain.

Exemple 15 *Considérons la distribution de possibilités concernant la température de conservation d'un produit alimentaire présenté dans l'exemple 14. L'événement A "température comprise entre 2 et 7 degrés" a pour mesure de nécessité* $N(2, 3, 4, 5, 6, 7) = 1 - \Pi(8, 9, 10) = 1 - 1 = 0$. *En effet, l'événement élémentaire "Température = 8 degrés", qui appartient à* $\bar{A}$, *est complètement possible. La mesure de nécessité associée à l'événement "température comprise entre 2 et 8 degrés" est* $N(2, 3, 4, 5, 6, 7, 8) = 1 - \Pi(9, 10) = 1 - 0.5 = 0.5$. *En effet, l'événement élémentaire "Température = 9 degrés", qui appartient à* $\bar{A}$, *est possible avec le degré 0.5.*

4.3.2 Mesures de possibilité et de nécessité d'événements flous

Dans la section précédente, nous avons vu qu'un événement est défini comme une disjonction d'événements élémentaires. Puis nous avons introduit les mesures de possibilité et de nécessité d'un événement. Dans cette section, nous appellerons ce type d'événement un événement ordinaire. En effet, nous nous intéressons dans cette section aux mesures de possibilité et de nécessité d'événements flous. Un événement flou étend la notion d'événement ordinaire et est défini comme une disjonction *pondérée* d'événements élémentaires. La pondération représente le degré de conformité de l'événement élémentaire à l'événement flou.

Exemple 16 *Considérons l'événement flou "température de conservation" défini par le sous-ensemble flou représenté dans la figure 4.1. Une température comprise dans l'intervalle [4, 8] degrés est complètement conforme à l'événement. Par contre, les valeurs comprises dans les intervalles [0, 4[et]8, 12] ont une conformité moindre avec cet événement. Les valeurs en dehors de l'intervalle [0, 12] degrés ne sont pas du tout conformes à l'événement.*

La mesure de possibilité d'un événement ordinaire privilégie l'événement élémentaire de l'événement ordinaire ayant le plus fort degré de possibilité (cf définition 21). La mesure de

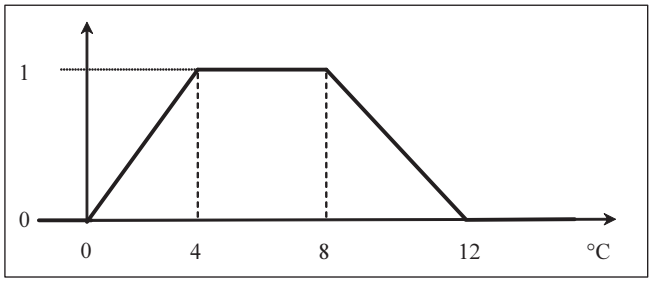


Fig. 4.1 – L'événement flou "Température de conservation"

possibilité d'un événement flou définie dans Zadeh (1978) privilégie le cas qui satisfait le mieux, à la fois les degrés de conformité des événements élémentaires de l'événement flou et la distribution de possibilités correspondant à la donnée imprécise avec laquelle on veut comparer l'événement flou.

Définition 24 Soit une distribution de possibilités π et un événement flou A, définis sur un ensemble X, la mesure de possibilité de A est: $\Pi(A) = sup_{u \in X} min(\mu_A(u), \pi(u))$.

La mesure de nécessité d'un événement flou est définie de manière analogue à celle d'un événement ordinaire:

Définition 25 Soit une distribution de possibilités π et un événement flou A, définis sur un ensemble X, la mesure de nécessité de A est:

$$N(A) = 1 - \Pi(\bar{A}) = 1 - sup_{u \in X} min(1 - \mu_A(u), \pi(u)) = inf_{u \in X} max(\mu_A(u), 1 - \pi(u)).$$

Exemple 17 *Considérons l'événement flou "température de conservation" ainsi que la distribution de possibilités "environ 6 degrés" définis par les sous-ensembles flous représentés dans la figure 4.2. Les mesures de possibilité et de nécessité de l'événement flou "température de conservation", noté A, valent* $\Pi(A) = N(A) = 1$.

4.4 Représentation de données imprécises dans le système d'intégration de données MIEL

Dans la section 3.6, nous faisions l'hypothèse que le langage de notre système d'intégration de données MIEL accédait uniquement à des données de valeur atomique. Dans cette section, nous présentons l'extension de notre langage de requête pour interroger des données imprécises. Nous considérons maintenant que la valeur associée à un attribut stocké en base est représentée par une distribution de possibilités.

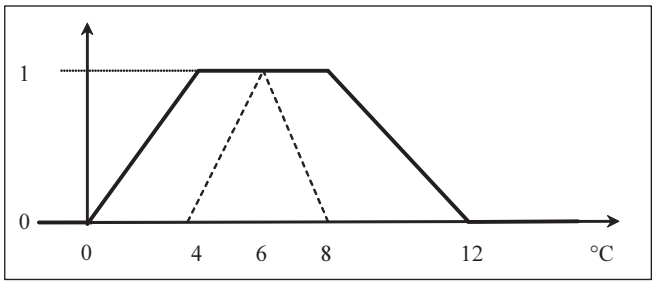


Fig. 4.2 – L'événement flou "Température de conservation" (en traits pleins) et la distribution de possibilités "environ 6 degrés" (en traits pointillés).

Définition 26 La valeur d'un attribut a est une distribution de possibilités π définie sur le domaine de valeurs de a. On note $\pi(x)$ le degré de possibilité que la valeur effective de a soit x.

Remarque 4 *Le domaine de définition d'un attribut a est l'ensemble de toutes les distributions de possibilités (ie tous les sous-ensembles flous normalisés) définissables sur le domaine de valeurs de a, noté* $Val(a)$*. Le cas où la valeur associée à l'attribut a est précise devient un cas particulier dans lequel:*

- *si le type du domaine de valeurs est numérique ou symbolique alors* $(\exists x \in Val(a) \mid \pi(x) = 1 \wedge (\forall y \in Val(a), y \neq x, \pi(y) = 0))$,
- *si le type du domaine de valeurs est symbolique hiérarchisé alors* $(\exists x \in Val(a) \mid \pi(x) = 1 \wedge (\forall y \in Val(a), y \leq x, \pi(y) = 1) \wedge (\forall y \in Val(a), y \neq x, y \not\leq x, \pi(y) = 0))$.

Dans la section 3.6, le degré de satisfaction d'un critère de sélection flou $attribut \approx valeurFloue$ par une donnée d de la base est défini comme le degré d'appartenance au sous-ensemble flou $valeurFloue$ (cf définition 19) de la valeur associée à l'attribut pour d. Cette définition doit être étendue puisque la valeur de l'attribut est maintenant représentée par une distribution de possibilités. Dans le cadre de la théorie des possibilités, un critère de sélection flou de la requête peut être vu comme un événement flou. Le degré de satisfaction d'un critère de sélection flou par la valeur imprécise associée à un attribut est donc maintenant encadré par deux mesures: une mesure de possibilité et une mesure de nécessité. Par conséquent, nous étendons les définitions d'une requête et d'une réponse à une requête, présentées dans la section 3.6, comme suit.

Définition 27 Une requête Q exprimée dans une vue V ayant n attributs interrogeables $a_1, \ldots, a_n$ est définie par ses degrés de possibilité et de nécessité minimum, notés $(\Pi_{min}, N_{min}) \in ([0,1])^2$ et par:

$$Q = \{a_1, \ldots, a_l | \exists a_{m+1}, \ldots, a_n (P_V(a_1, \ldots, a_n) \wedge (a_{l+1} \approx v_{l+1}) \wedge \ldots \wedge (a_m \approx v_m))\}$$

où $1 \leq l \leq m \leq n$, P_V est le prédicat qui caractérise la construction de la vue, $a_1, \ldots, a_l$ la liste des attributs de projection, $a_{m+1}, \ldots, a_n$ la liste des attributs interrogeables non utilisés

dans la requête, $(a_{l+1} \approx v_{l+1}), \ldots, (a_m \approx v_m)$ la liste des critères de sélection flous dans lesquels $a_{l+1}, \ldots, a_m$ sont les attributs de sélection, et $v_{l+1}, \ldots, v_m$ la liste des sous-ensembles flous qui leur sont respectivement associés.

Exemple 18 *La requête Q est exprimée dans la vue ExpérienceUnFacteur :* $Q = \{Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse | (P_{ExperienceUnFacteur}(Produit, Microorganisme, pH, NomFacteurEtudie, TypeReponse) \wedge (Produit \approx PreferencesProduit) \wedge (pH \approx PreferencespH))\}$. *Les sous-ensembles flous $PreferencesProduit$ et $PreferencespH$ sont donnés dans la figure 3.9. Les degrés de possibilité et de nécessité minimum de la requête sont fixés à* $\Pi_{min} = 0.8$ *et* $N_{min} = 0.6$.

Définition 28 La réponse A à la requête Q de la définition 27 est définie par $A = \{\tau_1, \ldots, \tau_r\}$, où τ_i, $i \in [1, r]$, est un tuple de la forme $\{\tau_i(a_1), \ldots, \tau_i(a_l)\}$. Chaque tuple τ_i de A satisfait les critères de sélection flous de Q avec les degrés $\Pi_i \geq \Pi_{min}$ et $N_i \geq N_{min}$ tels que $\Pi_i = min_{j=l+1,\ldots,m}\Pi(v_j, \tau_i(a_j))$ et $N_i = min_{j=l+1,\ldots,m}N(v_j, \tau_i(a_j))$, v_j le sous-ensemble flou associé à l'attribut de sélection a_j dans la requête Q, $\tau_i(a_j)$ étant la valeur imprécise associée à l'attribut de sélection a_j dans le tuple τ_i, Π une mesure de possibilité (cf définition 24) et N une mesure de nécessité (cf définition 25).

Exemple 19 *Un exemple de réponse à la requête de l'exemple 18 formulée dans la vue ExpérienceUnFacteur est donné dans le tableau 4.1.*

TAB. 4.1 – Une partie de la réponse à la requête de l'exemple 18 formulée dans la vue *ExpérienceUnFacteur*. L'attribut pH est un exemple de donnée imprécise.

(Π, N)	Produit	Microorganisme	pH [min, max]	Facteur	Type de réponse
$(1.0, 1.0)$	Lait entier	Bacillus Cereus	$[5.1, 5.2]$	Température	Cinétique de contamination
$(0.9, 0.9)$	Lait demi-écrémé	Listeria	$[5.0, 5.4]$	Température	Vitesse de croissance
$(0.8, 0.0)$	Lait écrémé	Listeria	$[6.0, 8.0]$	Température	Vitesse de croissance

Remarque 5 *Lorsque le domaine de valeurs d'un attribut de sélection est de type symbolique hiérarchisé, la valeur de sélection floue et la donnée imprécise qui doivent être comparés sont représentés par des SEFH (cf section 3.5). Afin de rendre leur comparaison possible sur le même domaine de définition, notre système d'intégration de données effectue un calcul de fermeture (cf définition 9) de chacun de ces deux SEFH.*

4.5 Modèle des graphes conceptuels

Comme nous l'avons indiqué dans l'introduction de ce mémoire, nous devons régulièrement intégrer dans notre système de nouvelles données biologiques dont la structure est hétérogène. Cette caractéristique des données pose un problème de maintenance du système d'intégration. Une première solution à ce problème consiste à adapter la structure de la base de données relationnelle à chaque intégration de nouvelles données. Cette solution a des inconvénients: (i) elle complexifie le schéma de la base, (ii) elle requiert l'intervention d'informaticiens car les modifications de structure nécessitent une réorganisation des données existantes et la propagation des modifications aux outils d'acquisition et d'interrogation de données. Il s'agit donc d'une opération qui, selon l'ampleur de la modification à opérer, peut être lourde. En pratique, dans le projet Sym'Previus où peu de moyens financiers sont consacrés à l'informatique, une seule mise à jour importante de la structure a été opérée en 6 ans.

Nous avons donc proposé un dispositif complémentaire, ne requérant pas l'intervention d'informaticiens, pour intégrer dans notre système des données en marge de la structure de la base de données relationnelle. Nous qualifions ces données de "faiblement structurées". Leur niveau de structuration est en effet très variable, allant du texte libre à une structuration proche d'un schéma relationnel (par exemple, un tableau de données expérimentales). Nous avons donc recherché un formalisme ne requérant pas la définition d'un schéma relationnel *a priori* tout en proposant un mécanisme efficace d'interrogation des données de la base. Un modèle de type graphe étiqueté nous a semblé bien adapté à ce genre de besoin, puisqu'il permet de représenter aisément, moyennant le respect de règles de syntaxe, des données de structures variables.

La notion de données faiblement structurées est à rapprocher de celles de données semi-structurées ou non-structurées également attribuées aux données modélisées par des graphes étiquetés (Buneman *et al.* 1996, Abiteboul *et al.* 1997, Abiteboul 1997). Dans la famille des modèles reposant sur l'utilisation des graphes étiquetés, nous avons fait le choix du modèle des graphes conceptuels (Sowa 1984) car, pour plusieurs raisons, il nous a semblé être bien adapté à nos données:

- Ce modèle permet de construire facilement des graphes de structures variées à partir d'un graphe conceptuel "patron" représentatif d'un type de données. Ajouter ou supprimer un attribut décrivant une donnée revient à ajouter ou supprimer des sommets et des arêtes du graphe. La construction de variations demande simplement d'enrichir le support terminologique si de nouveaux concepts doivent être introduits.
- Ce modèle permet très naturellement une manipulation graphique des données. Nos utilisateurs étant non-informaticiens, cette facilité d'utilisation a également été un argument important pour justifier notre choix.
- Le modèle requiert la définition d'un support terminologique utilisé dans la construction et la comparaison des graphes conceptuels représentant les données. Ce support terminologique inclut notamment une hiérarchie de types de concepts qui est tout à fait adaptée à la représentation des taxonomies de termes que nous manipulons dans nos applications

biologiques. Cette hiérarchie est utilisée par l'opération de projection, opération du modèle qui permet d'interroger une base de graphes conceptuels. Nous avons exploité ces éléments du modèle pour définir des opérations d'interrogation élargie de la base.
- Plusieurs plate-formes de développement sont disponibles pour implémenter un système d'interrogation de bases de graphes conceptuels. Nous avons utilisé la plateforme CoGITaNT (Genest & Salvat 1998), développée par David Genest, version étendue de la plateforme CoGITo (Guinaldo & Haemmerlé 1997), développée par Ollivier Haemmerlé dans le cadre de son travail de thèse.

Dans notre système d'intégration de données, nous nous sommes basés sur la formalisation définie dans Mugnier & Chein (1996) qui a l'avantage de disposer d'une sémantique logique bien définie. Nous introduisons brièvement ce modèle dans la sous-section 4.5.1. Puis, nous présentons dans la sous-section 4.5.2 l'extension du modèle que nous proposons pour représenter et comparer des graphes conceptuels incluant des données imprécises.

4.5.1 Introduction au modèle des graphes conceptuels

Les connaissances représentées dans une base de graphes conceptuels sont organisées en deux parties disjointes:
- la connaissance terminologique, appelée *support*, qui définit le vocabulaire utilisé pour représenter les connaissances factuelles,
- les connaissances factuelles, appelées *graphes conceptuels*, qui sont exprimées à partir du vocabulaire défini dans le support.

La relation de spécialisation et l'opération de projection sont les deux notions qui permettent de comparer des graphes conceptuels et par voie de conséquence d'interroger une base de graphes conceptuels.

Le support Le support définit le vocabulaire employé, son organisation hiérarchique et les contraintes portant sur son utilisation.

Définition 29 Un support S est un quintuplet $(T_C, T_R, \sigma, I, \tau)$ tel que:
- T_C est l'ensemble des types de concepts. Il est partiellement ordonné par la relation d'ordre "sorte de", notée $\leq$. Pour tous types de concept t_1, t_2 de T_C, si $t_1 \leq t_2$, alors t_1 est dit sous-type de t_2 et t_2, surtype de t_1. T_C possède un plus grand élément, noté $\top$, dit type universel, et un plus petit élément, noté $\bot$, dit type absurde.
- T_R est l'ensemble des types de relations. Il est partitionné en sous-ensembles T_{R_n} qui regroupent les types de relations de même arité n: $T_R = \cup_n T_{R_n}$, $(n \geq 1)$. Chaque sous-ensemble T_{R_n} est partiellement ordonné par la relation "sorte de", notée $\leq$.
- σ est une application appelée signature, qui a tout type de relation associe le type de concept maximal de chacun de ses arguments. A tout type de relation $t_r \in T_{R_n}$ est associé un n-uplet $\sigma(t_r) = (\sigma_1(t_r), \ldots, \sigma_i(t_r), \ldots, \sigma_n(t_r)) \in (T_C)^n$, où $\sigma_i(t_r)$ est le type de concept maximal associé au $i^{ème}$ argument de t_r.

- I est l'ensemble dénombrable des marqueurs individuels. Un marqueur individuel représente une instance d'un type de concept. Il existe également un marqueur générique, noté $*$, qui permet de représenter une instance non spécifiée d'un type de concept. L'ensemble $I \cup \{*\}$ est muni de l'ordre suivant: $*$ est plus grand que tous les marqueurs individuels, qui sont deux à deux incomparables.
- τ est une application de I dans $T_C \backslash \{\bot\}$, qui à tout marqueur individuel m associe un type de concept t: on dit que m est une instance de t. On appelle type de concept minimum du marqueur individuel m, le type de concept t tel que $\tau(m) = t$. Par ailleurs, on dit qu'un marqueur individuel est conforme à t si $\tau(m) \leq t$. L'ensemble des marqueurs individuels conformes à t ($\{m \in I \mid \tau(m) \leq t\}$) inclut l'ensemble des marqueurs individuels instances de t ($\{m \in I \mid \tau(m) = t\}$) .

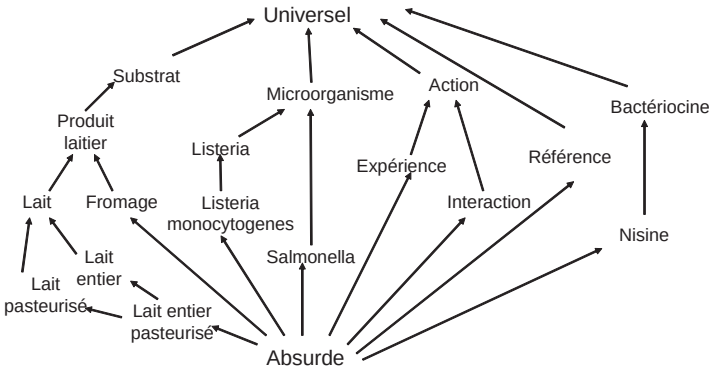


Fig. 4.3 – Une partie de l'ensemble des types de concepts T_C utilisé dans notre application biologique.

Exemple 20 *Nous présentons quelques exemples extraits du support utilisé dans le sous-système graphe conceptuel de notre système d'intégration de données.*

- *La figure 4.3 présente une partie de l'ensemble des types de concepts.*
- *Les types de relation utilisés ont principalement une sémantique de type grammatical. Il s'agit essentiellement de relations binaires comme Obj (a pour objet), Agt (a pour agent), Res (a pour résultat), Unité (a pour unité).*
- *La signature de Obj est (Action, Universel): Tout type d'action a pour objet n'importe quel type de concept (par exemple, une expérience peut avoir pour objet une interaction).*
- *Vialette2005 est un marqueur individuel, instance du concept Référence.*

Les graphes conceptuels Les graphes conceptuels, construits à partir du support, représentent les connaissances stockées dans la base de graphes. Un graphe conceptuel comporte deux types de sommets:

- les sommets concepts représentent les entités, attributs, événements, états; ils sont représentés par un rectangle.
- les sommets relations définissent les liens sémantiques entre les concepts; ils sont représentés par un ovale.

Définition 30 Un graphe conceptuel $G = (R, C, U, etiq)$ défini sur un support $S = (T_C, T_R, \sigma, I, \tau)$ est un multigraphe[2], non orienté[3], biparti[4] et non nécessairement connexe tel que:

- C et R sont les deux classes de sommets de G, appelés respectivement *sommets concepts* et *sommets relations*, avec $C \neq \emptyset$. Une fonction *type* est définie sur les sommets de G. Elle associe à tout sommet concept un type de concept de T_C, et à tout sommet relation un type de relation de T_R. Une fonction *marqueur* est définie sur les sommets concepts. Elle associe à tout sommet concept un marqueur individuel ou générique m ($m \in I \cup \{*\}$);
- U est l'ensemble des arêtes. L'ensemble des arêtes adjacentes à un sommet relation r (reliant r à ses sommets concepts voisins) est totalement ordonné. Cet ordre est représenté par une numérotation des arêtes de 1 à *degré(r)*, où *degré(r)* représente le nombre d'arêtes adjacentes au sommet relation r et doit être égal à l'arité du type de relation r.
- *etiq* est une application qui à tout sommet associe une étiquette:
  - pour chaque sommet $r \in R$, $etiq(r) \in T_R$: l'étiquette est le type du sommet relation;
  - pour chaque sommet $c \in C$, $etiq(c) \in (T_C \setminus \{\bot\}) \times (I \cup \{*\})$: l'étiquette est un couple (t, m) où t est le type de concept de c et m son marqueur (individuel ou générique).
- *etiq* vérifie les contraintes fixées dans S par les applications σ et τ:
  - pour tout sommet relation $r \in R$, de type t_r et de degré n, pour tout $i \in [1, n]$, il faut que $t_i \leq \sigma_i(t_r)$, où t_i est le type de concept du $i^{ème}$ voisin de r;
  - pour tout sommet concept ayant un marqueur individuel (t, m), il faut que $\tau(m) \leq t$.

Exemple 21 *Le graphe conceptuel de la figure 4.4 peut être interprété par: "L'interaction I1, décrite dans Jung 1992, entre la Listeria et la nisine dans du lait écrémé a pour résultat une réduction."*

La relation de spécialisation et l'opération de projection L'ensemble des graphes conceptuels d'une base de graphes est partiellement préordonné par une relation de spécialisation (notée $\leq$), qui peut être calculée par l'opération de projection (morphisme de graphes autorisant la restriction des étiquettes des sommets, conformément au support). $G' \leq G$ si et seulement si il existe une projection de G dans G' (Chein & Mugnier 1992). L'opération de projection est donc une opération fondamentale du modèle puisqu'elle permet d'interroger une base de graphes.

[2]Il peut exister plusieurs arêtes entre deux sommets.

[3]Les sommets sont reliés par des arêtes et non des arcs.

[4]Il existe deux classes de sommets, tels que les sommets d'une classe ne peuvent être liés qu'aux sommets de l'autre classe.

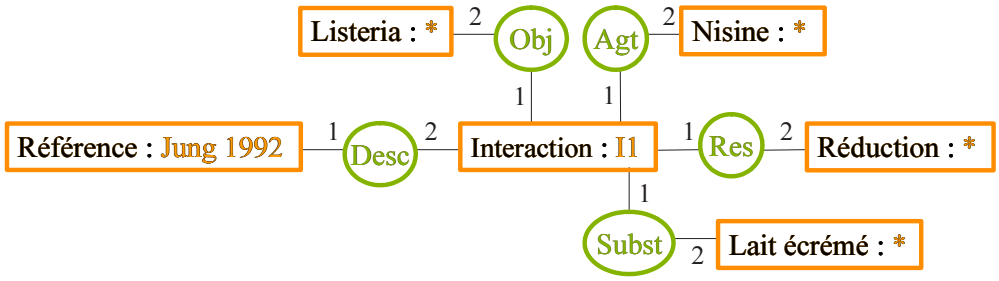


Fig. 4.4 – Un exemple de graphe conceptuel.

Définition 31 Une projection d'un graphe conceptuel $G = (R, C, U, etiq)$ dans un graphe conceptuel $G' = (R', C', U', etiq')$, définis sur un support commun S, est un couple d'applications $\Pi = (f, g)$, $f : R \to R'$, $g : C \to C'$, tel que:

- les arêtes et la numérotation des arêtes sont conservées: pour toute arête reliant un sommet relation r à un sommet concept c de U numérotée i, $f(r)g(c)$ est une arête de U' numérotée i;
- les étiquettes des sommets peuvent être restreintes:
  - pour tout sommet relation $r \in R$, $etiq'(f(r)) \leq etiq(r)$;
  - pour tout sommet concept $c \in C$, $etiq'(g(c)) \leq etiq(c)$, c'est à dire, si $etiq(c) = (t, m)$ et $etiq'(g(c)) = (t', m')$ alors $t' \leq t$ et $m' \leq m$;

Exemple 22 *Dans la figure 4.5, le graphe conceptuel G se projette dans le graphe conceptuel G'. G' est plus spécifique que G.*

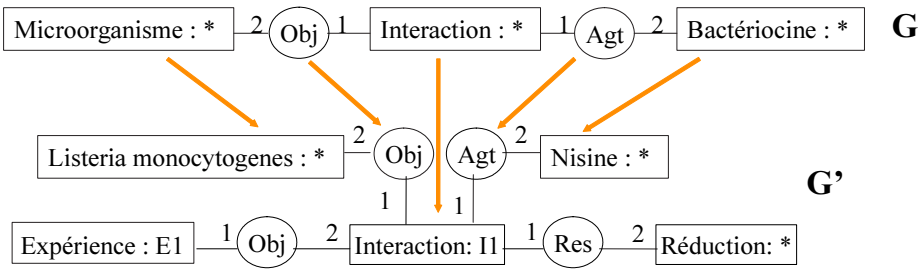


Fig. 4.5 – Projection du graphe conceptuel G dans G' ($G' \leq G$).

4.5.2 Extension du modèle des graphes conceptuels pour représenter et comparer des données imprécises

Nous avons vu dans la section 4.4 que le langage d'interrogation de notre système d'intégration de données est capable de prendre en compte une donnée imprécise représentée par une distribution de possibilité. Dans cette section, nous proposons une extension du modèle des graphes conceptuels pour représenter des données imprécises afin de pouvoir stocker les données internes faiblement structurées de notre système d'intégration.

Plusieurs travaux (Rundensteiner & Bandler 1986, Brouard 2000, Omri & Chouigui 2001) ont proposé d'introduire des notions floues dans les réseaux sémantiques. En ce qui concerne plus précisémenent le modèle des graphes conceptuels, le premier travail connu est la thèse de Morton (Morton 1987). Puis, plusieurs travaux (Wuwongse & Manzano 1993, Wuwongse & Tru 1996, Cao & Creasy 1998, Cao 1999) fondés sur la thèse de Morton ont été proposés.

Une étude détaillée de ces approches a été faite dans Thomopoulos (2003). Les critiques essentielles que nous avons faites de ces travaux sont les suivantes: (i) le domaine de valeurs des sous-ensembles flous représentant la valeur associée à une donnée imprécise n'est pas intégré au support du modèle des graphes conceptuels (Morton 1987); (ii) plusieurs extensions floues (sommets concepts flous, sommets relations flous) proposées sous des formes diverses (sous-ensemble flou, valeur de vérité floue, ...) font que la sémantique des graphes conceptuels flous peut être ambiguë (Wuwongse & Manzano 1993, Wuwongse & Tru 1996); (iii) la proposition de la notion de type flou conjonctif (Cao 1999) remet en cause le fait qu'un marqueur individuel est rattaché à un unique type de concept.

Nous avons préféré définir une extension du modèle des graphes conceptuels moins expressive, mais qui préserve l'homogénéité du modèle. Nous définissons les concepts de marqueurs et de types flous à partir du seul concept de sous-ensemble flou. De plus, les sous-ensembles flous sont toujours définis sur le support terminologique du modèle des graphes conceptuels, ce qui n'est pas le cas dans les approches comparables à la nôtre évoquées ci-dessus.

Nous proposons de représenter les valeurs associées aux données imprécises dans les sommets concepts. Notre extension, guidée par nos applications, ne concerne pas les sommets relations car nous n'avons pas eu besoin de mettre du flou dans les relations entre les données: l'essentiel de la sémantique est porté par les sommets concepts.

Nous considérons donc deux types de sommets concepts flous:

- le type est connu précisément, mais pas le marqueur: le marqueur est un sous-ensemble flou représentant une distribution de possibilités, défini sur l'ensemble des marqueurs individuels I;
- le type n'est pas connu précisément: le type est alors un sous-ensemble flou représentant une distribution de possibilités, défini sur l'ensemble des types de concepts T_C.

Marqueur flou Nous définissons la notion de marqueur flou pour décrire une valeur floue représentée sous forme de marqueur (cas des valeurs numériques ou des valeurs symboliques

non organisées en taxonomie, par exemple des noms d'auteurs).

Définition 32 Un marqueur flou M d'un type de concept t est un sous-ensemble flou défini sur l'ensemble des marqueurs individuels I. Il associe une valeur comprise entre 0 et 1 à tout marqueur individuel conforme à t et la valeur 0 aux autres.

Exemple 23 *Dans la figure 4.6, le graphe conceptuel comporte un sommet concept avec un marqueur flou, de type ValeurNumérique.*

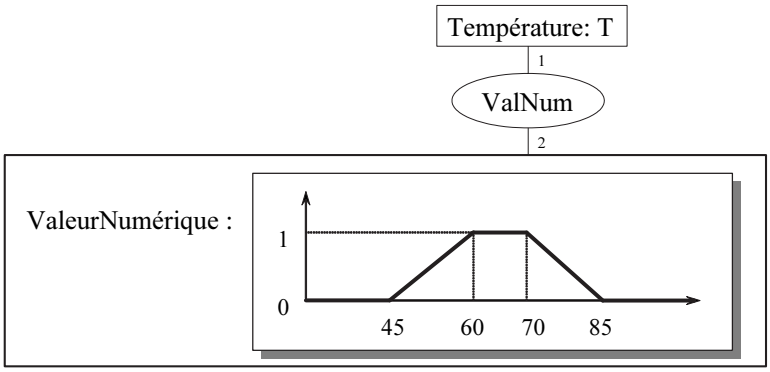


Fig. 4.6 – Un exemple de graphe conceptuel comportant un sommet concept avec un marqueur flou.

Il est important de noter que tous les marqueurs flous sont définis sur le même domaine de valeurs, I. Il est donc possible de les comparer en utilisant les opérateurs définis dans la théorie de la logique floue (voir section 3.3). Par conséquent, la relation de spécialisation entre deux marqueurs flous peut être définie comme une inclusion de sous-ensembles flous.

Définition 33 Soient deux marqueurs flous M et M' définis sur I, M' est une spécialisation de M si et seulement si $M' \subseteq M$, où $\subseteq$ désigne l'opération d'inclusion entre deux sous-ensembles flous (cf définition 4).

Type flou La notion de type flou permet de décrire une valeur floue lorsque celle-ci est représentée sous forme d'un type de concept (cas des valeurs symboliques organisées en taxonomie, par exemple des noms de produits alimentaires). Nous utilisons le concept de SEFH, sous-ensemble flou défini sur un sous-ensemble d'une hiérarchie de spécialisation (cf section 3.5.1), pour définir un type flou.

Définition 34 Un type flou T est un SEFH dont la hiérarchie de référence est l'ensemble des types de concepts T_C.

Exemple 24 *Dans la figure 4.7, le graphe conceptuel comportant un sommet concept avec un type flou s'interprète de la manière suivante: "Une interaction entre la Listeria et la nisine a été étudiée dans un substrat qui est certainement du lait, mais il n'est pas exclu que ce soit du yaourt".*

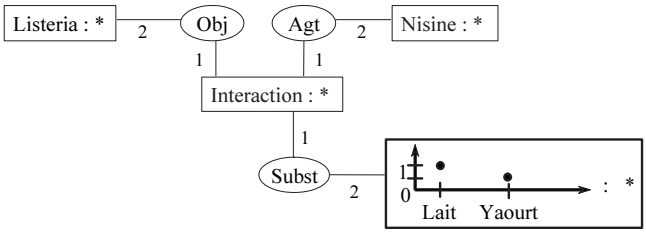


Fig. 4.7 – Un exemple de graphe conceptuel comportant un sommet concept avec un type flou (en gras).

La notion de fermeture de SEFH introduite dans la section 3.5.1 permet d'étendre la définition d'un type flou sur l'ensemble des types de concepts T_C. Tout comme les marqueurs flous, il est donc possible de comparer deux types flous en utilisant les opérateurs de la théorie de la logique floue. La relation de spécialisation peut donc également être définie entre deux types flous comme une inclusion de sous-ensembles flous.

Définition 35 Soient deux types flous T et T', $ferm(T)$ et $ferm(T')$ leurs fermetures respectives (définition 9) définies sur T_C, T' est une spécialisation de T si et seulement si $ferm(T') \subseteq ferm(T)$, où $\subseteq$ désigne l'opération d'inclusion entre sous-ensembles flous (cf définition 4).

Relation de spécialisation sur les graphes conceptuels flous Nous venons de voir que la relation de spécialisation entre deux marqueurs flous et entre deux types flous reposent toutes les deux sur la notion d'inclusion de sous-ensembles flous. Nous pouvons maintenant définir la relation de spécialisation entre deux sommets concepts flous.

Définition 36 Soient $e = (T, M)$ et $e' = (T', M')$ deux étiquettes de concepts, où T et T' sont des types flous, M et M' des marqueurs flous. e' est une spécialisation de e si et seulement si T' est une spécialisation de T et M' une spécialisation de M.

L'opération de projection des graphes conceptuels reste définie comme un morphisme de graphes autorisant la spécialisation des étiquettes des sommets, où la spécialisation des étiquettes des sommets concepts flous de la définition 36 remplace la définition standard.

Exemple 25 *Dans la figure 4.8, le graphe conceptuel G se projette dans le graphe conceptuel G', ces deux graphes comportant chacun un marqueur flou.*

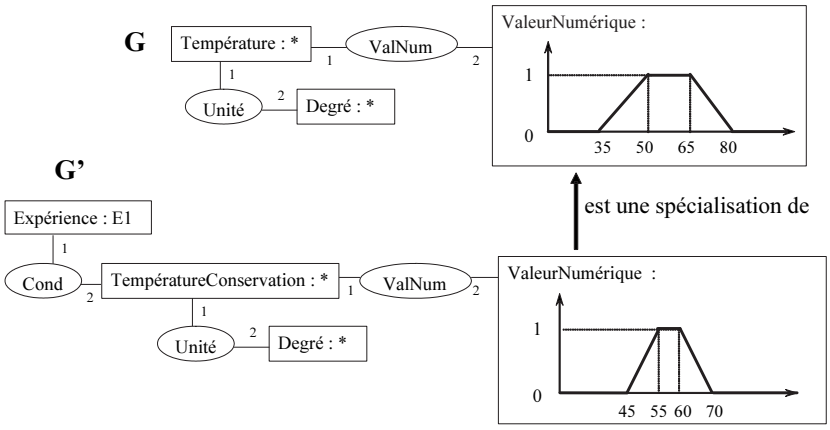


Fig. 4.8 – Un exemple de projection d'un graphe G dans un graphe G' faisant intervenir des marqueurs flous.

Comparaison plus souple de graphes conceptuels flous L'utilisation de la relation de spécialisation entre deux graphes conceptuels flous mène à un résultat binaire: soit un graphe G' est une spécialisation d'un graphe G, soit il ne l'est pas. Nous avons vu dans la section 4.3 que des sous-ensembles flous peuvent être comparés de manière plus souple dans le cadre de la théorie des possibilités en utilisant des mesures de possibilité et de nécessité d'événements flous. Nous introduisons une comparaison souple entre graphes conceptuels flous, basée sur ces deux mesures, que nous utilisons pour faire de l'interrogation élargie de la base de graphes conceptuels (cf section 5.2). Nous définissons pour cela une opération de projection étendue dans laquelle ces deux mesures permettent d'assouplir les comparaisons entre sommets concepts flous. Après avoir introduit la notion de compatibilité d'une part entre deux marqueurs flous et d'autre part entre deux types flous, nous étendons la notion de restriction d'étiquettes de sommets concepts utilisée dans la définition de l'opération de projection classique par la notion de compatibilité entre étiquettes de sommets concepts flous.

Définition 37 Soient deux marqueurs flous M et M' définis sur I, M' est compatible avec M avec le degré de possibilité $\Pi(M, M')$ et le degré de nécessité $N(M, M')$ introduits dans les définitions 24 et 25.

Définition 38 Soient deux types flous T et T', $ferm(T)$ et $ferm(T')$ leurs fermetures respectives (définition 9) définies sur T_C, T' est compatible avec T avec le degré de possibilité $\Pi(ferm(M), ferm(M'))$ et le degré de nécessité $N(ferm(M), ferm(M'))$.

La compatibilité entre étiquettes de sommets concepts flous est mesurée par l'agrégation conjonctive des degrés de compatibilité entre marqueurs flous et entre types flous, en utilisant l'opérateur *min*.

Définition 39 Soient $e = (T, M)$ et $e' = (T', M')$ deux étiquettes de sommets concepts flous, où T et T' sont des types flous, M et M' des marqueurs flous. e' est compatible avec e avec le degré de possibilité $\Pi(e, e')$ et le degré de nécessité $N(e, e')$ définis par:

- $\Pi(e, e') = min(\Pi(ferm(T), ferm(T')), \Pi(M, M'))$;
- $N(e, e') = min(N(ferm(T), ferm(T')), N(M, M'))$.

Nous proposons maintenant une extension de l'opération de projection basée sur la notion de compatibilité entre étiquettes de sommets concepts flous.

Définition 40 Soient $G = (R, C, U, etiq)$ et $G' = (R', C', U', etiq')$ deux graphes conceptuels flous définis sur un support commun S. G' est compatible avec G avec le degré de possibilité $\Pi(G, G')$ et le degré de nécessité $N(G, G')$ s'il existe un couple d'applications $P = (f, g)$, $f : R \rightarrow R'$, $g : C \rightarrow C'$, tel que:

- les arêtes et la numérotation des arêtes sont conservées: pour toute arête reliant un sommet relation r à un sommet concept flou c de U numérotée i, $f(r)g(c)$ est une arête de U' numérotée i;
- les étiquettes des sommets relations peuvent être restreintes.

$\Pi(G, G')$ et $N(G, G')$ sont définis par:

- $\Pi(G, G') = min_{c \in C}(\Pi(etiq(c), etiq'(g(c)))$;
- $N(G, G') = min_{c \in C}(N(etiq(c), etiq'(g(c)))$.

Exemple 26 *Un exemple de graphe conceptuel flou R utilisé comme requête d'interrogation de la base de graphes conceptuels est présenté dans la figure 4.9. Un exemple de graphe conceptuel flou D représentant une connaissance stockée dans la base de graphes est présenté dans la figure 4.10. Le graphe requête R est compatible avec le graphe donnée D avec un degré de possibilité $\Pi = 0.8$ et un degré de nécessité $N = 0$.*

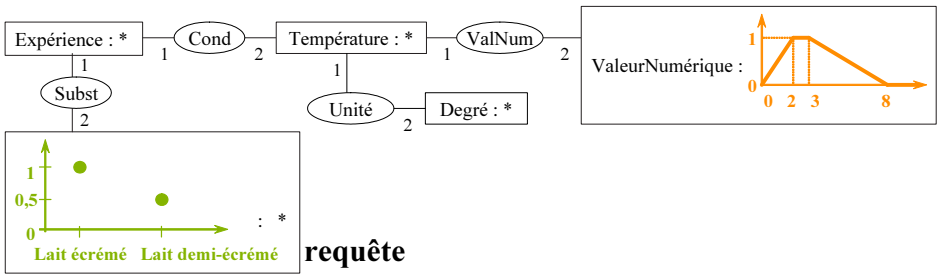


Fig. 4.9 – Un exemple de graphe conceptuel flou modélisant une requête.

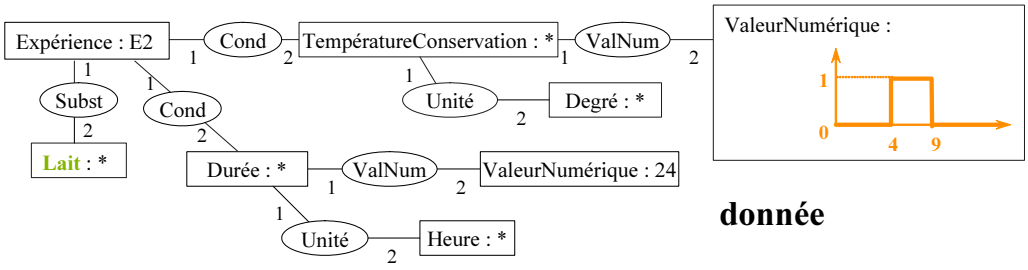


Fig. 4.10 – Un exemple de graphe conceptuel flou modélisant une donnée.

4.6 Modèle XML et application à l'annotation sémantique floue

Nous avons présenté dans le chapitre 3 une première solution au problème de l'incomplétude des données accessibles par notre système d'intégration. Cette solution repose sur l'utilisation de techniques d'interrogation flexible et de généralisation de requêtes. Nous proposons dans cette section une deuxième solution qui est complémentaire à la première. Elle consiste à enrichir les données accessibles par notre système d'intégration de données avec des données externes provenant du Web. Le Web est une source de données intéressante car: (i) il permet d'accéder à une grande quantité d'informations, (ii) il permet également d'accéder rapidement à l'information la plus récente sur un sujet donné. Ce dernier argument est particulièrement important en ce qui concerne nos domaines d'application (risque alimentaire microbiologique et chimique) dans lesquels la connaissance évolue rapidement.

Nous avons fait le choix de suivre une approche de type entrepôt de données pour intégrer à notre système les données externes provenant du Web. Ce choix est motivé par le fait que les données pertinentes ne sont pas nécessairement maintenues en permanence sur le Web, certaines peuvent disparaître. Il est donc nécessaire de les extraire du Web, de les transformer et de les stocker dans une base de données locale. Par ailleurs, nous voulons que l'utilisateur de notre système d'intégration puisse interroger simultanément les données internes et externes en utilisant l'ontologie du système d'intégration de données. Nous rappelons que cette ontologie est composée: (i) d'une taxonomie de termes hiérarchisés par la relation "sorte de", (ii) d'un ensemble de relations sémantiques qui sont utilisées dans les bases pour stocker les données internes. Pour atteindre cet objectif, nous proposons d'utiliser une méthode d'annotation des données externes guidée par l'ontologie. Cette approche a l'avantage d'être générique puisqu'il suffit de changer d'ontologie pour changer de domaine d'application. Nous la détaillons ci-dessous.

Nous avons fait le choix de stocker les données externes dans un entrepôt de données au format XML (Bray *et al.* 2004) car XML devrait devenir à terme le standard d'échange des données sur le Web. Malheureusement, pour le moment, très peu de documents sont disponibles

au format XML. Nous avons donc décidé de traduire les documents pertinents disponibles dans des formats variés (HTML, PDF, Word, ...) au format XML. Nous avons pour cela conçu et implémenté un logiciel d'acquisition de données externes provenant du web, baptisé AQWEB. Cette acquisition s'effectue en trois grandes étapes représentées dans la figure 4.11:

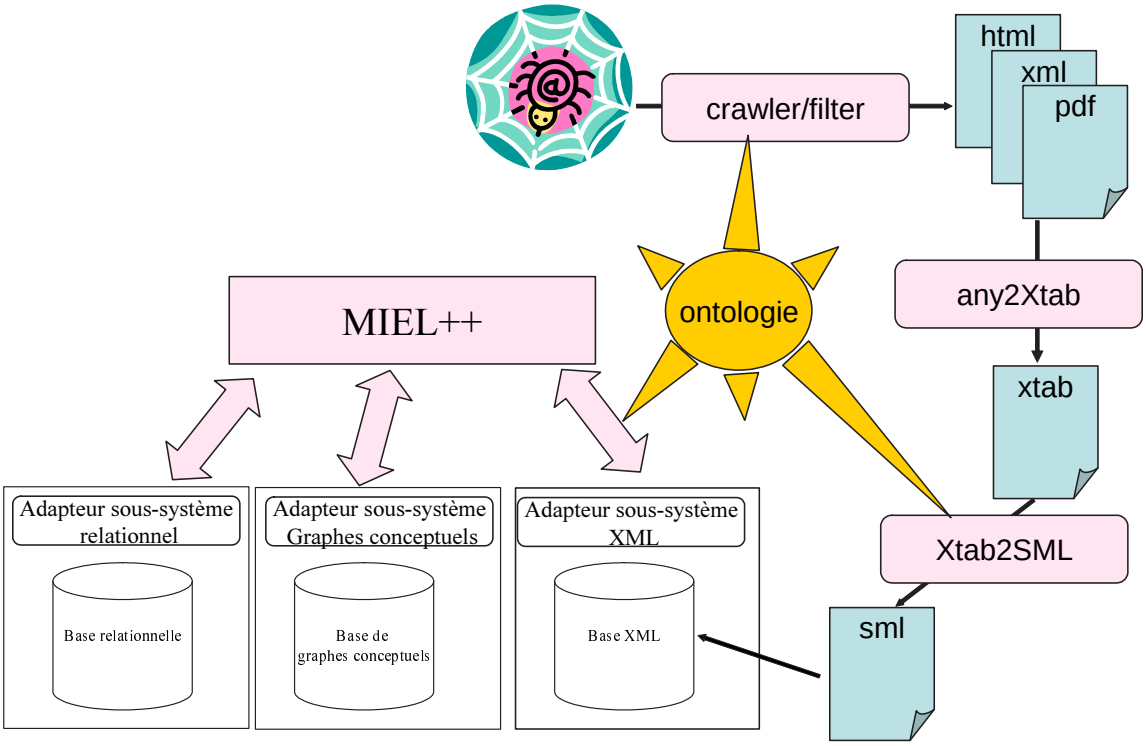


Fig. 4.11 – Figure présentant à la fois le système d'acquisition de données externes provenant du Web AQWEB et le système d'interrogation MIEL++.

- Etape 1: recherche-filtrage: AQWEB utilise un outil de recherche sur le Web (Google dans la version actuelle) pour récupérer des documents pertinents à partir de combinaisons de termes provenant de l'ontologie. Nous nous sommes principalement intéressés aux documents scientifiques contenant des tableaux de données. Nous avons fait ce choix pour deux raisons: (i) ces tableaux contiennent en général une synthèse pertinente des données publiées dans le document, (ii) la structuration de l'information en tableau facilite son traitement par un processus automatique.
- Etape 2: traduction en XML (ANY2XTAB): AQWEB traduit les tableaux de données issus des documents récupérés sur le Web dans un format XML générique, noté XTAB, permettant de représenter un tableau comme un ensemble de lignes, chaque ligne étant constituée d'un ensemble de cellules.
- Etape 3: annotation sémantique (XTAB2SML): L'annotation sémantique des tableaux avec les termes et les relations de l'ontologie est réalisée afin de permettre une interrogation homogène des tableaux du Web et des données internes en utilisant le langage

d'interrogation MIEL (cf sections 3.6 et 4.4) qui s'appuie sur cette ontologie. Chaque tableau, stocké dans un document XTAB, est annoté sémantiquement. AQWEB associe à chaque terme apparaissant dans le tableau XTAB la liste des termes de l'ontologie les plus proches en leur associant un degré de pertinence de l'appariement. Notre méthode d'évaluation de cette proximité sera précisée dans la suite de ce chapitre. Nous avons fait le choix de représenter cette liste de termes de l'ontologie pondérés par un degré de pertinence comme une donnée imprécise modélisée par une distribution de possibilités. Cette distribution de possibilités a la signification habituelle d'une disjonction exclusive pondérée de termes candidats à l'appariement. Les termes de l'ontologie ayant un degré de possibilité égal à un sont considérés comme les meilleurs appariements possibles. L'hypothèse sous-jacente qui peut sembler forte qu'un seul terme de l'ontologie soit le bon est dans la pratique tout à fait acceptable. Il n'arrive en effet que dans de rares exceptions que deux ou plusieurs termes de l'ontologie soient également acceptables. Enfin AQWEB cherche à instancier une ou plusieurs relations sémantiques de l'ontologie dans les colonnes du tableau. Le contenu initial du tableau XTAB et les annotations sémantiques ajoutées par AQWEB sont stockés dans un document SML.

Nous proposons dans la section 4.6.1 une extension du modèle de données XML pour représenter des données imprécises. Cette extension est notamment utilisée pour représenter les données imprécises générées par le processus d'annotation sémantique présenté ci-dessus (XTAB2SML). Nous présentons ensuite dans la section 4.6.2 le processus d'annotation sémantique XTAB2SML de manière plus détaillée. Nous développons plus spécifiquement notre contribution qui consiste à produire des annotations sémantiques floues associées aux termes des tableaux du Web.

4.6.1 Extension du modèle de données XML pour représenter des données imprécises

Nous présentons dans cette section le modèle de données XML utilisé pour représenter la base de données XML de notre système d'intégration de données. Puis nous proposons une extension de ce modèle pour représenter des données imprécises.

Alors que beaucoup de travaux se sont intéressés à la représentation de données imprécises dans les bases de données relationnelles et objet (voir Bosc *et al.* 2004*b* pour une synthèse récente), à notre connaissance très peu de travaux ont été réalisés dans le cadre des bases de données XML. Dans un domaine voisin, Turowski & Weng (2002) propose une représentation XML de variables linguistiques incluant la définition de sous-ensembles flous à domaine de valeurs numériques. Mais l'objectif de ce travail est de construire un système d'aide à la décision incluant une base de règles dans lesquelles ces variables linguistiques sont utilisées. A notre connaissance, dans le contexte des bases de données XML, seul Nierman & Jagadish (2002) a proposé une représentation de données probabilistes en XML dont s'est inspiré Zongmin (2005) pour l'adapter aux données possibilistes. L'extension proposée dans Zongmin (2005) permet de

représenter deux types d'informations floues : (i) un degré de possibilité peut être associée à un sous-arbre, (ii) une distribution de possibilités peut être associé à un élément feuille ou à un élément contenant un sous-arbre. Guidés par nos besoins applicatifs, l'extension que nous proposons (Buche *et al.* 2004, Buche *et al.* 2006*a*) ne concerne pour l'instant que la représentation d'une valeur imprécise associée à un attribut du système d'intégration, cet attribut étant défini dans un élément feuille d'un arbre XML. Notre extension est donc moins expressive que celle de Zongmin (2005). Par contre, nous proposons un mécanisme d'interrogation adapté à cette extension, ce qui n'est pas le cas de Zongmin (2005). Ce point sera abordé dans la section 5.3.

Nous définissons une base de données XML comme un ensemble d'arbres de données, chacun de ces arbres correspondant à un document XML. Nous nous sommes inspirés pour cela du modèle d'arbre étiqueté proposé par Aguiléra *et al.* (2000) et Xyleme (2001).

Définition 41 Un arbre de données est un triplet (a, e, v) où a est un arbre fini, e une fonction qui associe une étiquette à chaque nœud de a et v une fonction qui associe une valeur aux nœuds feuilles de a. Le couple (a, e) est appelé un arbre étiqueté.

Le schéma d'un arbre de données est défini par un *arbre de type* qui est un arbre étiqueté dans lequel aucun nœud ne peut avoir deux fils de même étiquette. Un arbre de données (a, e, v) est dit instance d'un arbre de type (a_t, e_t) s'il existe un homomorphisme de type strict de (a, e) vers (a_t, e_t) :

Définition 42 Soient (a, e) et (a', e') deux arbres étiquetés. La fonction h qui associe aux nœuds de a des nœuds de a' est appelée homomorphisme de structure strict si et seulement si (i) h préserve la racine de a: $racine(a') = h(racine(a))$, et (ii) h préserve la structure de a: pour tout nœud m, fils de n dans a, alors $h(m)$ est fils de $h(n)$ dans a'. h est appelé homomorphisme de type strict si et seulement si h est un homomorphisme de structure strict qui préserve les étiquettes de a: pour tout nœud n de a alors $e(n) = e'(h(n))$.

Le schéma de la base de données XML est défini par l'ensemble des arbres de type dont les arbres de données de la base sont des instances.

Nous proposons une extension de ce modèle de données XML pour représenter des données imprécises définies comme des distributions de possibilités sur des domaines de valeurs numériques ou symboliques (hiérarchisées ou non). Ces distributions de possibilités sont représentées par des arbres de données. Nous faisons l'hypothèse que les distributions de possibilités définies sur un domaine de valeurs numériques peuvent être représentées par un trapèze.

Définition 43 Soit f un sous-ensemble flou de forme trapézoïdale représentant une distribution de possibilités et défini sur un domaine de valeurs numériques. f est représenté par un arbre de données composé: (i) d'un élément racine étiqueté CFS (pour "Continuous Fuzzy Set"), et (ii) de 4 éléments fils étiquetés $minSup$, $minNoy$, $maxNoy$, $maxSup$ qui ont respectivement pour valeur associée: $min(support(f))$, $min(noyau(f))$, $max(noyau(f))$ et $max(support(f))$.

Définition 44 Soit f un sous-ensemble flou représentant une distribution de possibilités et défini par sa fonction d'appartenance μ_f sur un domaine de valeurs symboliques, noté Dom. f est représenté par un arbre de données composé de: (i) un élément racine étiqueté DFS (pour "Discrete Fuzzy Set"), et (ii) pour tout $x \in Dom$, il existe un élément fils étiqueté $ValF$ ayant lui-même deux sous-élements étiquetés $Item$ et MD (pour "Membership Degree") qui ont respectivement pour valeurs associées x et $\mu_f(x)$.

Exemple 27 *La figure 4.12 présente un exemple de donnée imprécise modélisée par un arbre de racine CFS et la figure 4.13 un exemple de donnée imprécise modélisée par un arbre de racine DFS.*

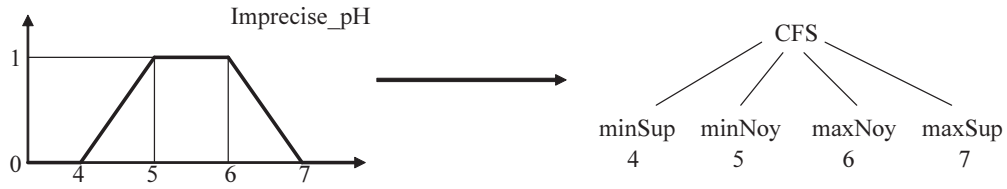


Fig. 4.12 – Exemple de donnée imprécise concernant une valeur de pH et sa représentation sous la forme d'un arbre de racine CFS.

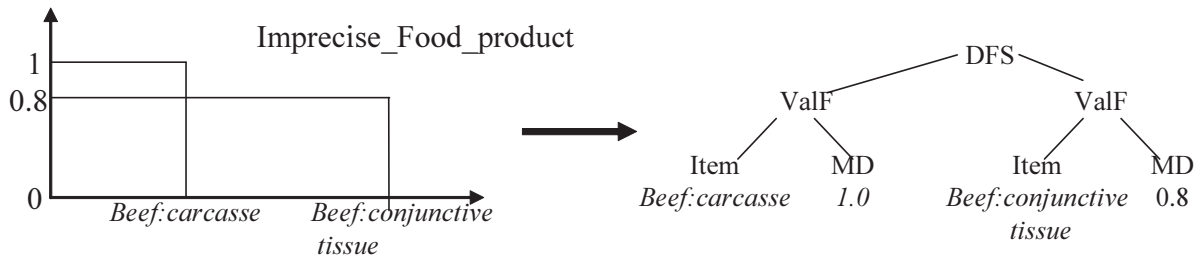


Fig. 4.13 – Exemple d'annotation floue modélisée par une distribution de possibilités et sa représentation sous la forme d'un arbre de racine DFS.

Notre entrepôt de données est composé d'un ensemble de documents XML pouvant contenir des données imprécises (par exemple des annotations sémantiques imprécises). Nous proposons de modéliser les documents XML comme des arbres de données flous dans lesquels les éléments feuilles peuvent contenir des valeurs imprécises.

Définition 45 Un arbre de données flou est un triplet (a, e, v) où a est un arbre fini, e une fonction qui associe une étiquette à chaque nœud de a et v une fonction qui associe une valeur

précise ou floue aux nœuds feuilles de a. Dans le cas où la valeur associée à un nœud n est précise, la fonction v associe au nœud n une valeur atomique. Dans le cas où la valeur associée à un nœud n est floue, la fonction v associe au nœud n un arbre de données de racine CFS ou DFS (cf définitions 43 et 44).

Exemple 28 *La figure 4.14 présente des exemples d'arbres de données flous contenant des éléments feuilles (ontoVal) dont la valeur associée est un arbre de données de racine DFS.*

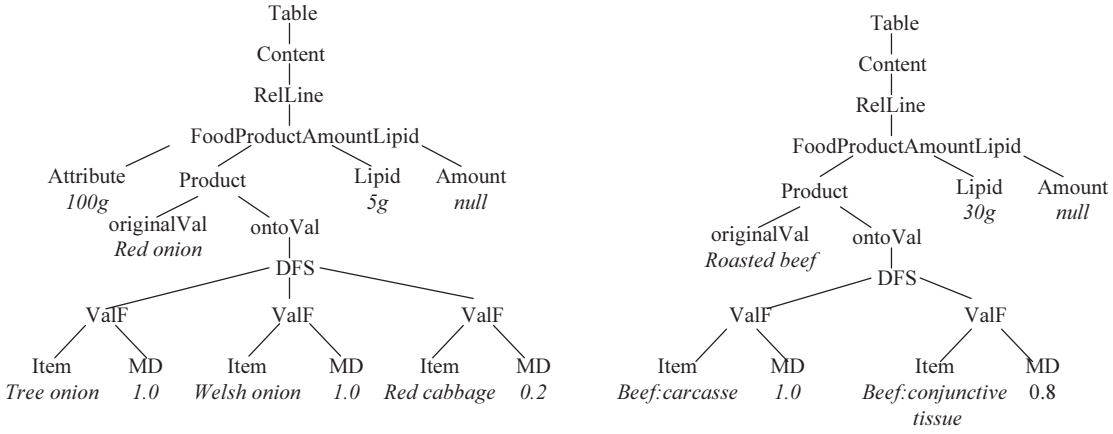


Fig. 4.14 – Exemples d'arbres de données flous XML contenant des éléments feuilles ($ontoVal$) dont la valeur associée est un arbre de données de racine DFS.

4.6.2 Annotation sémantique de documents

Le travail d'annotation sémantique de tableaux que nous présentons maintenant se situe dans le domaine de l'extraction d'informations. En effet, nous parlons d'annotation de tableaux dans la mesure où l'information initiale est conservée dans le document afin de garantir la traçabilité du processus d'annotation sémantique. Mais notre objectif est bien d'extraire du tableau les données intéressantes, c'est-à-dire celles que l'on peut mettre en correspondance avec les termes et les relations sémantiques définis dans l'ontologie. Plusieurs approches ont été proposées pour faire de l'extraction de données pertinentes à partir du Web.

– Extraction à partir de documents structurés: Ces méthodes dont le représentant le plus connu est Lixto (Baumgartner *et al.* 2001) sont bien adaptées à l'extraction d'informations dans des pages Web dynamiques générées à partir de bases de données. Elles requièrent la définition d'une règle d'extraction, nécessitant une intervention humaine, adaptée à chaque nouvelle structure de présentation des données. Ces méthodes ne sont pas adaptées à notre problème car nous devons traiter un grand nombre de tableaux contenant parfois peu de lignes et dont la structure est très variable. Il serait alors nécessaire d'envisager une intervention humaine pour définir la règle d'extraction adaptée à chaque tableau, ce qui n'est pas envisageable.

- Extraction à partir de documents non structurés: tout comme pour les documents structurés, les méthodes proposées pour extraire de l'information pertinente à partir d'un texte en langue naturelle sont fondées sur la définition de règles d'extraction. Aussi bien les approches supervisées (Ciravegna 2001, Freitag & Kushmerick 2000) qui nécessitent de définir un lot important d'exemples annotés manuellement que les méthodes non supervisées (Yangarber *et al.* 2002) utilisent le contexte linguistique dans lequel apparaissent les données pertinentes à extraire. Or, dans nos tableaux de données, les cellules contiennent rarement des phrases entières. Elles contiennent souvent simplement un ou plusieurs termes. Ces méthodes ne semblent donc pas non plus adaptées à notre besoin.
- Annotation de tableaux: Pivk *et al.* 2004 propose une méthode d'annotation de tableaux qui permet de découvrir la signature de la relation sémantique représentée dans le tableau. Ce travail de découverte est réalisé en utilisant des outils sémantiques génériques comme WordNet ou Google Sets. Notre objectif n'étant pas de découvrir des relations génériques, mais de reconnaître des relations prédéfinies dans une ontologie d'un domaine spécialisé, cette approche n'est pas adaptée à notre problème.

Nous avons donc été amenés à concevoir une méthode adaptée à notre besoin. Une première étude (Gagliardi *et al.* 2005) a été réalisée par nos partenaires du LRI dans le cadre du projet E.dot en collaboration avec Ollivier Haemmerlé. Nous en expliquons les grands principes avant de présenter nos propres propositions qui visent à améliorer cette méthode. Comme indiqué dans l'introduction de la section 4.6, chaque tableau récupéré sur le Web est annoté sémantiquement et stocké dans un document SML.

Annotation sémantique SML Le processus d'annotation s'effectue en trois étapes:

- Etape 1: Pour chaque terme apparaissant dans le tableau, on recherche la liste (non pondérée) des termes de l'ontologie les plus proches. Chaque terme du tableau et de l'ontologie est représenté par l'ensemble de mots lemmatisés qui le composent. Par ordre de priorité décroissante, trois types de comparaison sont effectués entre ensembles de mots: l'égalité, l'inclusion et l'intersection. L'annotation associée à chaque terme du tableau est la liste (éventuellement vide) des termes de l'ontologie ainsi trouvés et le type de comparaison qui a permis de les obtenir.
- Etape 2: Pour chaque colonne du tableau, on cherche à identifier un terme de l'ontologie représentatif de son contenu (appelé terme-attribut). Cette identification est effectuée en deux étapes: (i) pour chaque terme-attribut candidat, les termes de l'ontologie identifiés dans les cellules de la colonne au cours de l'étape 1, s'ils sont plus spécifiques que celui-ci, sont sélectionnés; (ii) le terme-attribut candidat ayant une proportion majoritaire de termes sélectionnés par rapport au nombre de cellules de la colonne est retenu comme terme-attribut de la colonne. Si cette identification échoue, le terme apparaissant dans l'entête de la colonne est comparé à la liste des termes de l'ontologie avec la méthode de comparaison de l'étape 1. Si cette deuxième tentative d'identification échoue également, le terme-attribut générique *attribute*, indiquant que l'identification n'a pas été possible,

est associé à la colonne.

– Etape 3: La signature du tableau, définie comme l'ensemble des termes-attributs associés aux colonnes dans l'étape 2, est comparée à chaque signature de relation sémantique de l'ontologie (également spécifiée comme une liste de termes de l'ontologie). Cette comparaison est effectuée sur chaque ligne du tableau. Elle a pour résultat une ou plusieurs instanciations de relations sémantiques de l'ontologie. Ces instanciations peuvent être totales ou partielles. Une instanciation partielle est autorisée si au moins deux termes-attributs de la relation sémantique sont instanciables dans la ligne. Si aucune instanciation n'est possible, la relation générique *relation* est associée à la ligne du tableau.

Exemple 29 *La figure 4.15 présente deux exemples de tableaux de données extraits du Web et annotés sémantiquement.*

Dans le tableau de gauche, la première colonne a été identifiée comme correspondant au terme-attribut Product et la troisième au terme-attribut Lipid. Par contre, la deuxième colonne, qui correspond à la quantité de produit analysé, n'a pas été identifiée car son entête contient l'abréviation Qty. Elle est donc annotée avec le terme-attribut générique Attribute. La relation sémantique FoodProductAmountLipid(Product, Amount, Lipid) de l'ontologie est partiellement instanciée sur chaque ligne du tableau. En effet, l'attribut Amount, qui correspond à la colonne Qty, n'a pas été reconnu. Cette colonne annotée avec le terme-attribut générique Attribute est quand même ajoutée à l'instanciation de la relation FoodProductAmountLipid afin d'éviter une mauvaise interprétation des données. L'arbre de données SML de gauche présenté dans la figure 4.16 correspond à la troisième ligne du tableau.

Dans le tableau de droite, la première colonne a été identifiée comme correspondant au terme-attribut Product et la deuxième au terme-attribut Lipid. La relation sémantique FoodProductAmountLipid(Product, Amount, Lipid) de l'ontologie est partiellement instanciée car il n'y a pas de colonne correspondant à l'attribut Amount dans le tableau. L'arbre de données SML de droite présenté dans la figure 4.16 correspond à la deuxième ligne du tableau.

Annotation sémantique SML floue Dans ce paragraphe, nous proposons une amélioration de la méthode précédente concernant les annotations effectuées sur les termes du tableau. Nous nous sommes rendu compte par l'expérimentation que le nombre de termes de l'ontologie obtenus par égalité des ensembles de mots était très faible. Dans la grande majorité des cas, les annotations sont obtenues par des comparaisons de type inclusion ou intersection de mots. Il nous a alors semblé intéressant d'associer à un terme du tableau non pas une liste de termes de l'ontologie jugés pertinents, mais plutôt une liste ordonnée par un degré de pertinence. Classiquement, en recherche d'information, ce degré de pertinence est calculé à partir de la proportion de mots communs. Nous proposons un calcul de pertinence qui tient compte non seulement du nombre de mots communs, mais également de l'importance sémantique de ces mots. En effet, on voit bien que si l'on veut comparer le terme *Oignon rouge* avec les termes *Chou rouge* et *Oignon de printemps*, la simple prise en compte de la proportion de mots communs ne

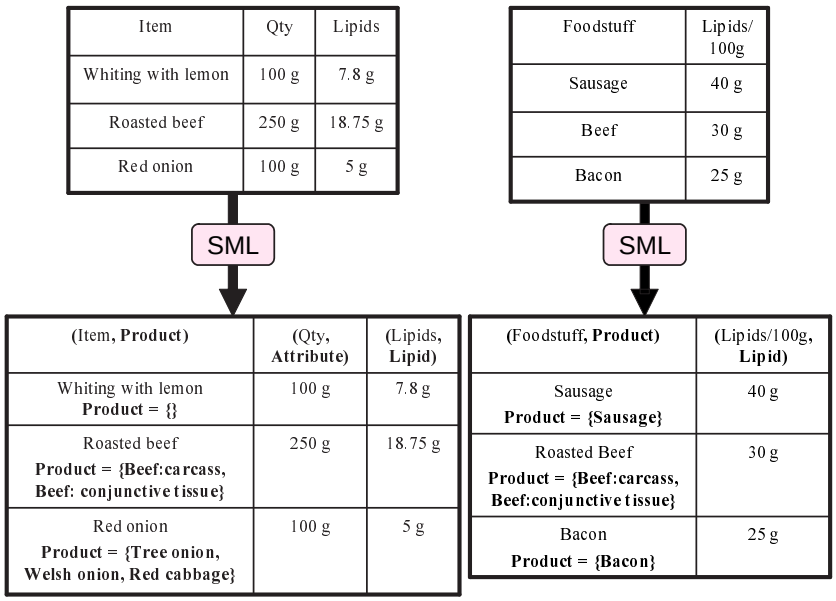


Fig. 4.15 – Deux exemples de tableaux de données provenant du Web et les annotations sémantiques associées (en gras).

permettra pas de distinguer les deux appariements.

Pour effectuer ce type de comparaison, il faut préalablement déterminer l'importance sémantique des mots composant chacun des termes de l'ontologie. Nous proposons pour cela d'associer à chaque mot un poids, compris entre 0 et 1, représentatif de son importance dans le terme. Pour simplifier la pondération qui est réalisée manuellement, seulement trois valeurs sont utilisées: 0 pour les mots vides (par exemple les conjonctions de coordination, les articles, ...); 1 pour les mots les plus importants et une valeur comprise entre 0 et 1 pour les mots modificateurs de sens (une valeur de 0.2 a été retenue après expérimentation).

Exemple 30 *Les termes suivants sont pondérés par:*
- *chou rouge: (chou: 1; rouge: 0.2);*
- *oignon de printemps: (oignon: 1; printemps:0.2, de:0);*
- *fruits et légumes: (fruit: 1; légumes: 1, et: 0);*
- *produits laitiers: (produits: 0.2, laitiers: 1).*

En ce qui concerne les termes du Web, la pondération ne peut pas être effectuée manuellement car on ne connaît pas *a priori* leur sens. Une pondération automatique basée sur le rôle grammatical du mot dans le terme (poids fort pour les noms et poids faible pour les adjectifs) n'a pas été retenue au vu du grand nombre de contre-exemples rencontrés dans l'ontologie (par exemple: dans *produits laitiers*, c'est l'adjectif *laitiers* qui est porteur de sens). Par conséquent, tous les mots non vides sont considérés comme d'égale importance. Un poids de 1 leur est associé.

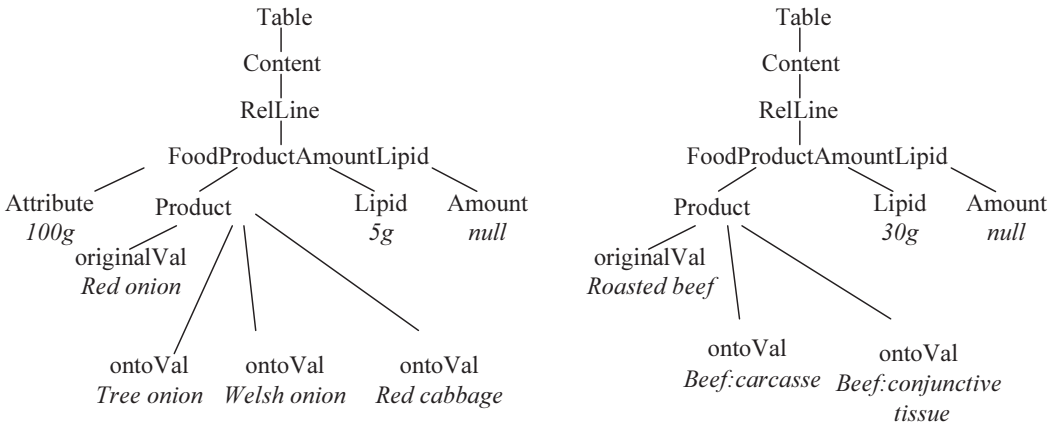


Fig. 4.16 – Deux exemples d'arbres de données SML issus des tableaux présentés dans la figure 4.15.

Afin de calculer le degré de pertinence entre un terme du Web et un terme de l'ontologie, nous avons comparé les résultats obtenus avec deux mesures souvent utilisées en recherche d'information, le coefficient de Dice (Lin 1998) et la mesure cosinus (van Rijsbergen 1979). Nous n'avons pas obtenu de différence significative: pour un terme du Web, l'ordre des termes de l'ontologie défini par le degré de pertinence reste le même. La mesure cosinus étant la plus couramment utilisée, nous l'avons choisie pour calculer le degré de pertinence entre deux termes. Chaque terme est représenté sous la forme d'un vecteur pondéré. Chaque vecteur a autant de coordonnées que de mots lemmatisés possibles (tous les mots contenus dans les termes de l'ontologie et les mots contenus dans le terme du Web à annoter). Chaque coordonnée a pour valeur le poids, dans le terme considéré, du mot correspondant à cette coordonnée : 0 si le mot n'appartient pas au terme, le poids du mot dans le terme sinon.

Définition 46 Soient w un terme du Web et o un terme de l'ontologie, représentés par les vecteurs de mots lemmatisés suivants, $\vec{w} = (w_1, \ldots, w_n)$ et $\vec{o} = (o_1, \ldots, o_n)$, leur mesure de similarité par cosinus est:

$$cos(w, o) = \frac{\Sigma_{i=1}^{n} w_i o_i}{\sqrt{\Sigma_{i=1}^{n} {w_i}^2 \times \Sigma_{i=1}^{n} {o_i}^2}}$$

Exemple 31 *Les degrés de pertinence entre le terme du Web Oignon rouge et les termes de l'ontologie Oignon de printemps et Chou rouge sont: cos(Oignon rouge, Oignon de printemps)=0.693 et cos(Oignon rouge, Chou rouge)=0.139.*

Dans la suite de ce mémoire, nous considérerons que la liste ordonnée de termes de l'ontologie associée à un terme du tableau est une annotation sémantique imprécise, modélisée par une distribution de possibilités.

Exemple 32 *Les arbres de données SML de la figure 4.14 correspondent à ceux présentés dans la figure 4.16. Dans la figure 4.14, les arbres de données SML sont flous. Les termes de*

l'ontologie associés à un terme du tableau sont considérés comme une annotation sémantique imprécise contenue dans l'élément ontoVal et représentée par un arbre de données de racine DFS.

La mesure de pertinence entre termes basée sur l'importance sémantique de leurs mots a été évaluée expérimentalement. Pour cela, 185 noms d'aliments (en anglais) ont été collectés à partir de tableaux de données provenant du Web. Cette liste de termes a été annotée avec deux ontologies différentes: le Codex Alimentarius, taxonomie de 1644 termes regroupés en 3 niveaux utilisée par l'OMS et l'ontologie des aliments Sym'Previus, taxonomie de 507 termes ayant une profondeur de spécialisation maximale de 7 niveaux. Les termes de ces deux ontologies ont été pondérés en utilisant les trois valeurs indiquées ci-dessus. Pour chacun des termes du Web, la meilleure correspondance (notée "best match") dans chacune des deux ontologies a été déterminée manuellement.

Le tableau 4.2 donne les résultats obtenus en utilisant le degré de pertinence de la définition 46. Les résultats sont globalement meilleurs lorsque l'on utilise l'ontologie Sym'Previus pour annoter. Cela s'explique par le fait que dans le domaine du risque alimentaire, on s'intéresse à la fois aux matières premières et aux produits transformés. L'ontologie Sym'Previus a été réalisée dans cet esprit alors que le Codex Alimentarius est orienté exclusivement matières premières. L'utilisation des pondérations sur les mots permet d'obtenir dans tous les cas de meilleurs résultats. Les résultats (surtout ceux obtenus pour l'ontologie Sym'Previus) montrent que l'ordre défini par le degré de pertinence permet souvent d'obtenir le best match dans les premiers termes. En effet, 65% des best matches sont classés dans les 5 premiers alors que le nombre moyen de termes de l'ontologie associés à un terme du tableau est de 16.

Tab. 4.2 – Résultats expérimentaux concernant l'annotation de 185 noms d'aliments.

	Best matches (%) en première position	Best matches (%) dans les 5 premiers	Best matches (%) annotés
avec le Codex Alimentarius			
mots pondérés	34%	52%	60%
poids à 1	30%	46%	60%
avec l'ontologie Sym'Previus			
mots pondérés	46%	65%	78%
poids à 1	45%	61%	78%

4.7 Conclusion du chapitre

Nous avons présenté dans ce chapitre un aspect original de notre système d'intégration de données: sa capacité à représenter et à rendre interrogeables des données imprécises internes ou

externes. Pour cela, nous avons premièrement défini une extension du langage d'interrogation de notre système d'intégration de données.

Deuxièmement, nous avons proposé une extension du modèle des graphes conceptuels permettant de représenter des données imprécises sous la forme de types flous et de marqueurs flous. Nous avons cherché dans cette extension à préserver l'homogénéité du modèle. Pour cela, les sous-ensembles flous utilisés dans la définition des types flous et des marqueurs flous sont définis sur le support terminologique du modèle des graphes conceptuels. Nous avons ensuite étudié les conséquences de cette extension sur les opérations de comparaison utilisées dans le modèle des graphes conceptuels. Nous avons redéfini l'opération de projection en nous appuyant sur l'opération d'inclusion de sous-ensembles flous. Cette extension préserve l'aspect binaire de l'opération de projection. Nous avons ensuite défini la notion de compatibilité entre deux graphes conceptuels qui permet d'effectuer une comparaison plus souple. C'est cette deuxième extension qui est utilisée par l'adapteur qui exécute une requête MIEL dans le sous-système graphes conceptuels de notre système d'intégration.

Troisièmement, nous avons proposé une extension du modèle XML pour représenter des données imprécises modélisées par des arbres de données flous. Puis, après avoir rappelé les grandes lignes de la méthode d'annotation sémantique de tableaux provenant du Web conçue par nos partenaires du LRI dans le cadre du projet e.dot en collaboration avec O. Haemmerlé (Gagliardi *et al.* (2005)), nous avons présenté une amélioration de cette méthode. Cette amélioration concerne l'annotation sémantique de chaque terme du tableau par les termes les plus proches de l'ontologie de notre système d'intégration de données. Cette annotation, modélisée comme une donnée imprécise, est représentée dans la base de données XML dans un arbre de données flou.

Dans le cadre de la thèse de Gaëlle Hignette et du projet WebContent, nous allons continuer à améliorer la méthode d'annotation sémantique de tableaux. Dans Gagliardi *et al.* (2005), une méthode permettant de typer les colonnes symboliques a été proposée. La prochaine étape de notre travail consistera à proposer une méthode de catégorisation des colonnes qui intégrera également un typage des colonnes numériques. Pour cela, nous comptons enrichir l'ontologie avec des connaissances permettant d'effectuer ce typage, notamment en décrivant les unités de mesure utilisées dans les colonnes numériques. Une fois obtenue la signature du tableau composé d'un ensemble de colonnes de type numérique ou symbolique, nous étudierons comment représenter l'incertitude associée à l'instanciation des relations sémantiques de l'ontologie, notamment lorsque cette instanciation est partielle.

Chapitre 5

Adapteurs de requêtes

5.1 Introduction

Ce chapitre est consacré à la présentation des adapteurs qui transforment une requête MIEL en une requête exécutable par les sous-systèmes de notre système d'intégration de données. La conception de l'adapteur de requête MIEL en requête graphes conceptuels a été initiée en collaboration avec Ollivier Haemmerlé. L'adapteur a été étendu à la prise en compte de données numériques et de données imprécises dans le cadre de la thèse de Rallou Thomopoulos (Thomopoulos 2003), que j'ai co-encadrée avec Ollivier Haemmerlé. Ce travail a donné lieu aux publications suivantes:

- Conférence nationale INFORSID 2000: Buche & Haemmerlé (2000),
- International Conference on Conceptual Structure, ICCS'2000: Buche & Haemmerlé (2000*a*),
- International Conference Flexible Querying Answering Systems, FQAS'2000: Buche & Haemmerlé (2000*b*),
- International Symposium on Methodologies for Intelligent Systems, ISMIS'2003: Buche *et al.* (2003),
- Journal of Intelligent Information Systems: Haemmerlé *et al.* (2007).

La conception de l'adapteur de requête MIEL en requête XML, également présentée dans ce chapitre, a été initiée en collaboration avec Juliette Dibie-Barthélemy et Ollivier Haemmerlé. Son extension pour dériver des requêtes XML approximatives a été réalisée en collaboration avec Juliette Dibie-Barthélemy et Fanny Wattez, maître de conférences contractuelle à l'INA P-G en 2005-06. Ce travail a donné lieu aux publications suivantes:

- International Conference Flexible Querying Answering Systems, FQAS'2004: Buche *et al.* (2004)[1],
- International Conference Flexible Querying Answering Systems, FQAS'2006: Buche *et al.* (2006*d*),

[1] déjà citée dans le chapitre 4

– Journal of Intelligent Information Systems: Buche *et al.* (2006*a*)[2].

Nous avons vu dans le chapitre 2 consacré à l'architecture de notre système d'intégration qu'une requête MIEL est exécutée simultanément par chacun des trois sous-systèmes (relationnel, graphes conceptuels et XML). L'interrogation flexible d'une base de données relationnelle en utilisant des critères de sélection flous ayant été déjà explorée dans de nombreux travaux (cf la synthèse récente dans Bosc *et al.* (2004*b*)), nous présentons uniquement dans ce chapitre les adapteurs des sous-systèmes graphes conceptuels et XML qui représentent les aspects les plus originaux de notre travail. Le lecteur intéressé par l'adapteur de requête du sous-système relationnel en trouvera une présentation synthétique dans Haemmerlé *et al.* (2007).

Dans ce chapitre, nous considérons le langage de requête MIEL défini dans la section 3.6 avec son extension présentée dans la section 4.4 qui permet de comparer un critère de sélection flou à une donnée imprécise stockée dans notre système d'intégration de données.

Nous présentons dans la section 5.2 l'adapteur qui traduit une requête MIEL en un graphe conceptuel requête. Les concepts de type flou et de marqueur flou proposés dans l'extension du modèle des graphes conceptuels présentée dans la section 4.5.2 sont utilisés pour construire un graphe conceptuel requête flou. L'interrogation de la base de graphes conceptuels est effectuée en utilisant la notion de compatibilité entre deux graphes conceptuels flous également présentée dans la section 4.5.2.

Nous présentons dans la section 5.3 l'adapteur qui traduit une requête MIEL en un arbre requête XML flou. La base de données XML de notre système d'intégration est composée d'un ensemble d'arbres de données XML flous (voir section 4.6.2). Ces arbres de données sont annotés sémantiquement avec l'ontologie de notre système d'intégration. Nous avons vu dans la section 4.6.2 que ces annotations comportent des instanciations de relations de l'ontologie qui peuvent être partielles ou complétées avec des colonnes qui ne font pas partie de leur signature. L'imperfection de ces annotations nous a amené à proposer un mécanisme de dérivation de requêtes XML approximatives prenant en compte ces variations de structure. Notre adapteur combine donc la flexibilité procurée par l'utilisation de critères de sélection flous et celle apportée par le système de dérivation de requêtes approximatives permettant l'insertion, la suppression et le renommage de nœuds de l'arbre requête.

5.2 Adapteur du sous-système graphes conceptuels

Comme nous l'avons indiqué dans la section 2.5, l'ontologie du système d'intégration doit être répliquée dans le support terminologique du sous-système graphes conceptuels pour que celui-ci puisse fonctionner correctement. Nous indiquons dans la section 5.2.1 comment le support du sous-système graphes conceptuel est construit à partir de l'ontologie. Nous présentons les choix de modélisation que nous avons faits dans la base, notamment pour la représentation

[2]déjà citée dans le chapitre 4

des valeurs numériques, dans la section 5.2.2. Le mécanisme d'interrogation de la base repose sur ces choix. Puis, dans la section 5.2.3, nous présentons le mécanisme de traduction d'une requête MIEL en requête graphe conceptuel flou.

5.2.1 Construction du support terminologique

Comme indiqué dans la section 4.5.1, le support terminologique d'une base de graphes conceptuels est composée d'un ensemble de types de concepts, d'un ensemble de marqueurs individuels et d'un ensemble de relations. Nous présentons ci-dessous les règles de construction de ces trois ensembles pour la base de graphes conceptuels de notre système d'intégration.

L'ensemble des types de concepts : L'ensemble des types de concepts (cf section 4.5.1, définition 29) est construit à partir de l'ontologie de notre système d'intégration de la manière suivante:

1. un type de concept t_a est associé à chaque attribut interrogeable a, c'est-à-dire un attribut présent dans au moins une signature de relation de l'ontologie;
2. si le domaine de valeurs de l'attribut a, constitué d'un ensemble de valeurs v_i, est hiérarchique (cf section 3.6):
   (a) un type de concept t_{v_i} est associé à chaque valeur v_i;
   (b) la liste des types de concepts t_{v_i} est partiellement ordonnée selon l'ordre partiel défini sur le domaine de valeurs de l'attribut a;
   (c) t_a est défini comme un généralisant commun à la liste des types de concept t_{v_i};
3. Tous les types de concepts construits dans l'étape 2(a) ayant même étiquette sont fusionnés;
4. Tous les types de concepts construits dans les étapes précédentes sont rassemblés dans un ensemble de types de concept unique en leur ajoutant un sur-type commun (*Universel*) et un sous-type commun (*Absurde*).

Exemple 33 *La figure 5.1 présente un exemple d'ensemble de types de concepts extrait de la base de graphes conceptuels de notre système d'intégration. L'attribut Substrat et son domaine de valeurs hiérarchique apparaît dans cet ensemble de types de concepts.*

L'ensemble des marqueurs individuels : Toutes les valeurs appartenant aux domaines de valeurs des attributs de l'ontologie de type numérique ou symbolique non hiérarchique sont insérées dans l'ensemble des marqueurs individuels.

Exemple 34 *La valeur 3 associée à l'attribut Durée est représentée comme un marqueur individuel. La valeur Rivituso associée à l'attribut Auteur est également représentée par un marqueur individuel.*

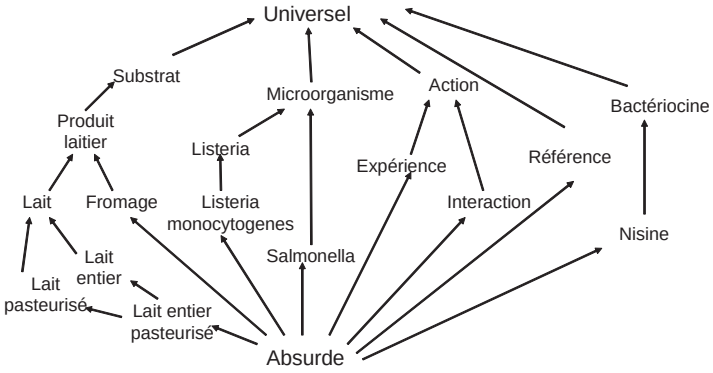


Fig. 5.1 – Une partie de l'ensemble des types de concepts T_C utilisé dans notre application biologique.

L'ensemble des types de relation: Cet ensemble ne joue pas un rôle important dans notre base de graphes conceptuels car l'essentiel de la sémantique des connaissances est contenu dans les sommets concepts. Les relations que nous utilisons indiquent principalement les liens grammaticaux entre les concepts lorsque l'on traduit un graphe conceptuel en langage naturel (par exemple: les types de relation *objet*, *agent* ou *caractéristique*).

5.2.2 Représentation des valeurs dans la base de graphes conceptuels

Les valeurs associées aux attributs sont représentées de manière différente dans la base de graphes conceptuels selon le type de l'attribut.

Le type de l'attribut est numérique: si l'attribut a est de type numérique et que l'on note v sa valeur, le couple (a, v) est représenté par le graphe conceptuel composé du sommet concept générique de type a, du sommet concept de type *ValeurNumérique* ayant pour marqueur individuel v, ces deux sommets étant reliés par le sommet relation étiqueté $ValNum$.

Exemple 35 *La figure 5.2 présente un exemple de graphe conceptuel représentant le couple (Durée, 3).*

Le type de l'attribut est symbolique: si l'attribut a est de type symbolique non hiérarchique et que l'on note v sa valeur, le couple (a, v) est représenté par un sommet concept étiqueté par le type de concept a et le marqueur v.

Exemple 36 *La figure 5.3 présente le sommet concept représentant le couple* $(Auteur, Rivituso)$.

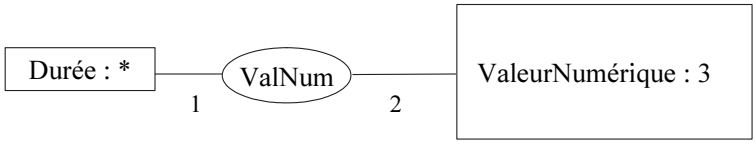


Fig. 5.2 – Un exemple de graphe conceptuel représentant un couple (attribut, valeur), le type de l'attribut étant numérique: *(Durée, 3)*.

Fig. 5.3 – Un exemple de graphe conceptuel représentant un couple (attribut, valeur), le type de l'attribut étant symbolique: $(Auteur, Rivituso)$.

Le type de l'attribut est symbolique hiérarchisé: si l'attribut a est de type symbolique hiérarchisé et que l'on note v sa valeur, le couple (a, v) est représenté par un sommet concept générique étiqueté par le type de concept v.

Exemple 37 *La figure 5.4 présente le sommet concept représentant le couple* $(Substrat, Lait)$.

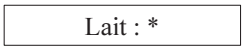


Fig. 5.4 – Un exemple de graphe conceptuel représentant un couple (attribut, valeur), le type de l'attribut étant symbolique hiérarchisé: $(Substrat, Lait)$.

5.2.3 Traduction d'une requête MIEL en un graphe conceptuel requête

Dans le langage MIEL, une requête s'exécute dans une vue. Nous avons introduit dans Buche & Haemmerlé (2000*a*) la notion de graphe-schéma qui est une interprétation de la notion de vue en termes de graphes conceptuels. Un graphe-schéma est un graphe conceptuel qui relie sémantiquement tous les attributs interrogeables d'une vue.

Définition 47 Un graphe-schéma S associé à une vue V ayant n attributs interrogeables $a_1, \ldots, a_n$ est un couple {G, C} où G est un graphe conceptuel acyclique et C=$\{c_1, \ldots, c_n\}$ un ensemble de sommets concepts distincts correspondant aux attributs interrogeables de la vue. Le type de chaque sommet concept c_i doit correspondre au type de concept associé à l'attribut a_i.

Exemple 38 *La figure 5.5 présente un graphe-schéma correspondant à la vue Interaction qui comporte 5 attributs interrogeables: Substrat, Pathogène, Durée, Température, Résultat Expé.*

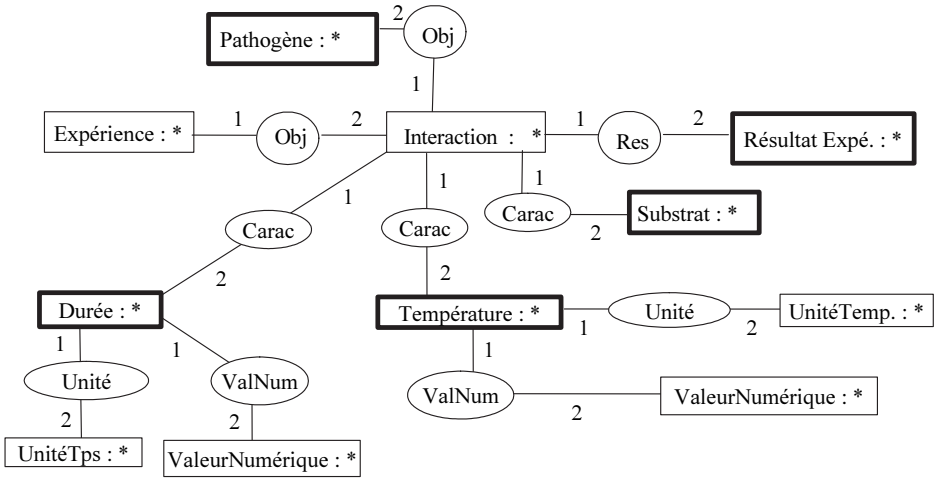


Fig. 5.5 – Un exemple de graphe-schéma pour la vue *Interaction*. Les sommets concept de C sont mis en gras.

Lorsqu'une requête MIEL est adressée au sous-système graphes conceptuels, l'adapteur du sous-système graphes conceptuels doit sélectionner les graphes conceptuels de la base qui satisfont tous les critères de sélection de la requête et dans les graphes sélectionnés, extraire les valeurs associées aux attributs de projection. Pour ce faire, le graphe-schéma correspondant à la vue interrogée est spécialisé en "instanciant" les sommets concept de C pour prendre en compte les valeurs associées aux attributs de sélection. Le résultat de cette spécialisation est appelé graphe-requête.

Définition 48 Soit $(a \approx v)$ un critère de sélection d'une requête MIEL Q exprimée dans une vue V et $S = \{G, C\}$ le graphe-schéma associé à la vue V. Soit c le sommet concept associé à a dans G. La spécialisation de G pour obtenir un graphe-requête est obtenue de la manière suivante:

- si le type de a est symbolique, le marqueur générique de c est remplacé par le marqueur individuel précis ou flou v;

– si le type de a est numérique, le marqueur générique du sommet concept de type *ValeurNumérique* relié à c par le sommet relation $ValNum$ est remplacé par le marqueur individuel précis ou flou v;
– si le type de a est symbolique hiérarchique, le type du sommet concept c est restreint au type de concept correspondant à v dans l'ensemble des types de concept (cf section 5.2.1).

Exemple 39 *La figure 5.6 présente un exemple de graphe-requête, spécialisation du graphe-schéma de la figure 5.5. Ce graphe-requête correspond à la requête MIEL* $Q = \{Substrat, Pathogène, Durée, Température, Résultat\ Expé | (P_{Interaction}(Substrat, Pathogène, Durée, Température, Résultat\ Expé) \wedge (Substrat \approx (1/Lait\ écrémé + 0.5/Lait\ demi\text{-}écrémé)) \wedge (Pathogène \approx Listeria) \wedge (Durée \approx (0;2;3;8))\}$. *Tous les attributs interrogeables de la vue sont des attributs de projection de la requête. Les attributs Substrat, Pathogène et Durée sont également attributs de sélection. L'attribut Durée étant de type numérique, le marqueur générique du sommet concept de type ValeurNumérique relié à Durée par le sommet relation* $ValNum$ *est remplacé par le marqueur flou correspondant au sous-ensemble flou trapézoidal (0;2;3;8). L'attribut Pathogène étant de type symbolique hiérarchique, le type du sommet concept Pathogène est restreint au type Listeria. C'est la même règle qui est appliquée à l'attribut Substrat, le type du sommet concept Substrat est restreint au type flou correspondant au sous-ensemble flou (1/Lait écrémé + 0.5/Lait demi-écrémé).*

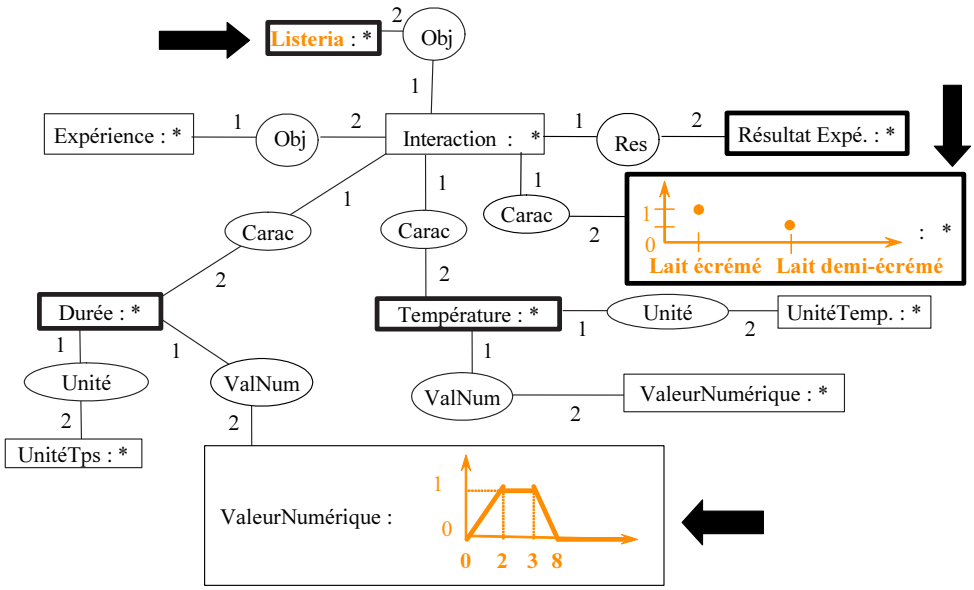


Fig. 5.6 – Un exemple de graphe-requête correspondant au graphe-schéma de la figure 5.5. Les flèches indiquent les sommets concept spécialisés pour prendre en compte les critères de sélection. Les sommets concept mis en gras correspondent aux attributs de projection.

Etant donné que le graphe-requête peut contenir des sommets concept flous et que les graphes conceptuels stockés dans la base peuvent contenir des données imprécises, l'interrogation de la base est effectuée en utilisant la notion de compatibilité entre deux graphes conceptuels flous présentée dans la section 4.5.2 (cf définition 40). La question de l'existence d'une projection d'un graphe conceptuel dans un autre est un problème NP-complet. Cependant, il existe des cas polynomiaux. Notre algorithme qui implémente l'opération de compatibilité entre graphes flous repose sur l'algorithme polynomial de Mugnier & Chein (1996) qui pose la question de l'existence d'une projection d'un graphe acyclique dans un autre graphe. Nous avons montré dans Thomopoulos *et al.* (2006) que notre algorithme reste polynomial. C'est la raison pour laquelle nous avons limité la définition des graphes-vue à des graphes conceptuels acycliques.

A chaque fois que l'adapteur trouve un graphe de la base compatible avec le graphe-requête, un tuple réponse est construit pour chaque projection obtenue en extrayant du graphe de la base les valeurs associées aux attributs de projection.

Définition 49 Soit a un attribut de projection d'une requête MIEL Q exprimée dans une vue V. Soit G le graphe-requête associé à Q et c le sommet concept associé à a dans G. Soit c' le sommet concept image de c dans G', G' étant un graphe de la base compatible avec G. La valeur v associée à l'attribut de projection a est obtenue de la manière suivante:

- si le type de a est symbolique, v est le marqueur de c';
- si le type de a est numérique, v est le marqueur du sommet concept de type *ValeurNumérique* relié à c' par le sommet relation $ValNum$;
- si le type de a est symbolique hiérarchique, v est le type du sommet concept c'.

Exemple 40 *Le résultat de la comparaison du graphe-requête de la figure 5.6 avec le graphe de la base de la figure 5.7 permet de construire le tuple réponse :* $((\Pi = 0.8, N = 0)$, *Substrat=Lait écrémé, Pathogène=Listeria monocytogenes, Durée= [4, 10], Température=20, Résultat Expé=Croissance).*

5.3 Adapteur du sous-système XML

De nombreuses approches ont été proposées dans la bibliographie pour introduire de la flexibilité dans le processus de comparaison entre un arbre requête XML et un arbre de données XML. Dans Damiani & Tanca (2000), l'arbre de données XML est encodé. Cet encodage est utilisé pour introduire des nœuds intermédiaires dans l'arbre requête afin de pouvoir le comparer aux arbres de données. Cette approche ne permet ni le renommage, ni la suppression de nœuds. L'approche de Amer-Yahia *et al.* (2002) est basée sur la transformation de la requête et permet l'introduction, le renommage et la suppression de nœuds. L'approche de Schlieder (2002) est une combinaison des deux premières. Les arbres de données sont encodés et la requête est transformée. Cette approche a pour avantage d'effectuer une évaluation précise du coût de transformation de l'arbre requête pour le faire correspondre avec l'arbre de données. Mais elle

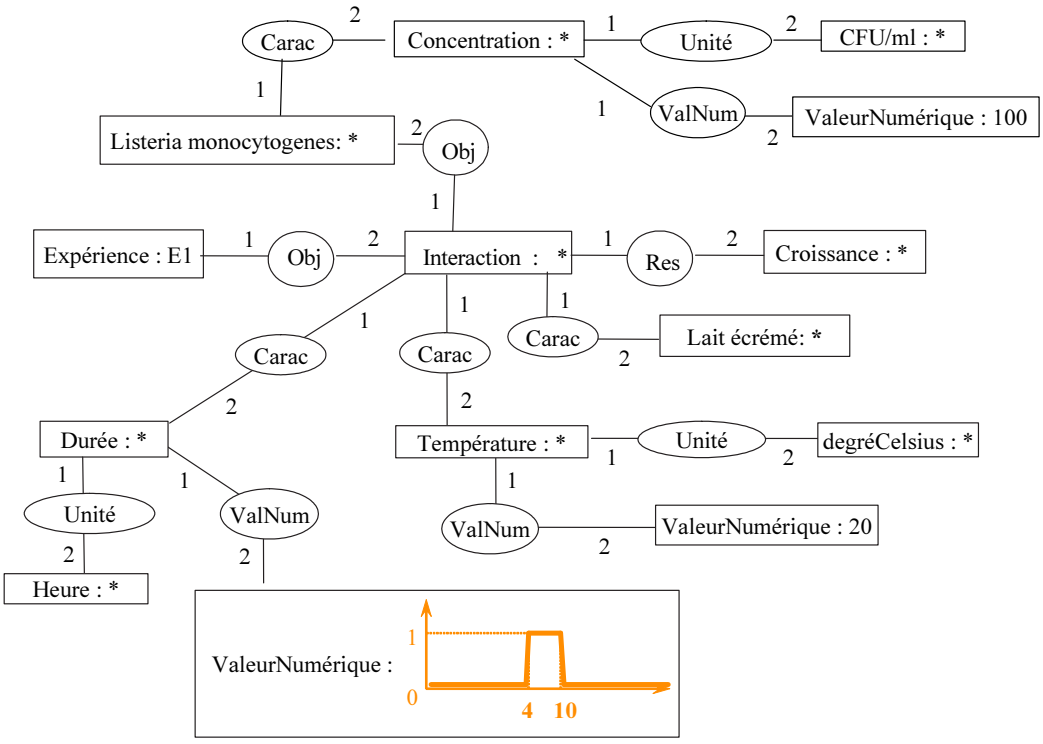


Fig. 5.7 – Un exemple de graphe stocké dans la base.

est difficile à utiliser car elle nécessite de redéfinir l'encodage et les mécanismes d'indexation des arbres de données XML, et ne permet donc pas de s'appuyer sur des systèmes de gestion de base de données XML existants. L'approche de Braga *et al.* (2002) permet d'utiliser des critères de sélection flous dans l'arbre requête pour exprimer des conditions de sélection flexible et faire de la mise en correspondance flexible de sous-arbres. Mais cette approche ne permet ni la suppression, ni le renommage de nœuds.

Dans cette section, nous présentons l'adapteur de requête MIEL du sous-système XML de notre système d'intégration. Dans le langage MIEL, une requête s'exécute dans une vue. L'adapteur du sous-système XML interprète la notion de vue sous la forme d'un arbre XML qui permet de relier sémantiquement les attributs interrogeables de la vue. L'adapteur doit sélectionner les arbres de données de la base qui satisfont tous les critères de sélection de la requête et dans les arbres de données sélectionnés, extraire les valeurs associées aux attributs de projection. Dans la conception de cet adapteur, nous avons cherché à développer une approche originale en combinant la flexibilité apportée d'une part par l'utilisation de critères de sélection flous et d'autre part par la transformation de la structure de l'arbre requête XML (incluant insertion, suppression et renommage de nœuds) pour effectuer une comparaison approximative entre un arbre requête XML et un arbre de donnée XML. Notre approche supporte la représentation de données imprécises modélisées par des distributions de possibilité dans les

arbres de données XML. Ce point est également une originalité de notre travail car il n'y a pas eu, à notre connaissance, dans la littérature d'autre proposition comparable s'appuyant sur la théorie des possibilités (cf le positionnement bibliograhique de la section 4.6.1). Enfin, notre proposition est compatible avec les standards actuels concernant la technologie XML puisque la transformation finale des requêtes, avant exécution, est effectuée en XQuery, standard du W3C (http://www.w3.org/XML/Query/).

Dans la section 5.3.1, nous présentons l'interprétation par notre adapteur XML des concepts de vue et de requête du langage MIEL. Puis, nous définissons dans la section 5.3.2 la notion de requête approximative et dans la section 5.3.3 la réponse à une requête exacte puis à une requête approximative.

5.3.1 Vue et requête dans l'adapteur du sous-système XML

Nous avons vu dans la section 4.6.2 que la base de données XML de notre système d'intégration est alimentée de manière semi-automatique à partir de tableaux de données extraits du Web. Ces tableaux ont été annotés sémantiquement avec les relations et les termes de l'ontologie de notre système d'intégration de données. Nous avons vu dans la section 4.6.2 que ces annotations sont imparfaites. Les instanciations de relations de l'ontologie peuvent être partielles ou complétées avec des colonnes qui ne font pas partie de la signature. Des relations et des colonnes peuvent également ne pas être identifiées, elles sont alors annotées respectivement avec le nom de relation générique *Relation* et le terme-attribut générique *Attribute*. Les tableaux annotés sont stockés sous forme d'arbres de données SML flous dans la base XML. Une vue MIEL s'interprète dans l'adapteur du sous-système XML comme une instance d'un arbre de type dans laquelle les nœuds correspondant aux attributs interrogeables sont identifiés. Chaque vue est construite à partir des relations et des termes de l'ontologie afin de pouvoir interroger la base SML. Afin de pouvoir prendre en compte les variations de structure engendrées par l'annotation dans les documents SML, l'adapteur XML va générer des requêtes approximatives dans lesquelles des nœuds appartenant à la vue auront été soit supprimés, soit renommés. Un coût de transformation est associé à chaque requête approximative. L'information nécessaire pour générer les requêtes approximatives est stockée dans la définition des vues. A chaque nœud de l'arbre XML matérialisant une vue (sauf le nœud racine), il est possible d'associer un coût de suppression et une liste de renommages exprimés par des couples (renommage de l'étiquette, coût du renommage).

Définition 50 Une vue conforme à un arbre de type (a_t, e_t) est un quadruplet $V=(a_V, e_V, w_V, c_V)$ où (a_V, e_V) est une instance de (a_t, e_t), w_V est une fonction qui associe la valeur *ql* (pour "queriable leaf") aux nœuds feuilles précis ou flous (cf définitions 43, 44 et 45) de a_V qui sont alors considérés comme nœuds interrogeables, c_V est une fonction qui associe aux nœuds de a_V (sauf le nœud racine) leurs coûts de transformation. Les coûts de transformation d'un nœud n sont représentés par un couple de la forme $(cs_n, \{(r_n^1, cr_n^1), \ldots, (r_n^p, cr_n^p)\})$ où cs_n est

le coût de suppression du nœud n et r_n^1, ..., r_n^p sont les renommages possibles du nœud n avec leur coût associé cr_n^1, ..., cr_n^p.

Exemple 41 *La figure 5.8 présente un exemple de vue utilisant la relation* FoodProductAmountLipid *de l'ontologie dans laquelle quatre attributs sont définis comme interrogeables:* Product.originalVal, Product.finalVal, Amount *et* Lipid. *Un coût de suppression de* 1000 *(resp.* 100*) est associé au nœud* FoodProductAmountLipid *(resp.* Amount*). Un renommage possible en Relation (resp. Attribute) est associé au nœud* FoodProductAmountLipid *(resp.* Amount*) avec un coût de renommage de* 50 *(resp.* 10*).*

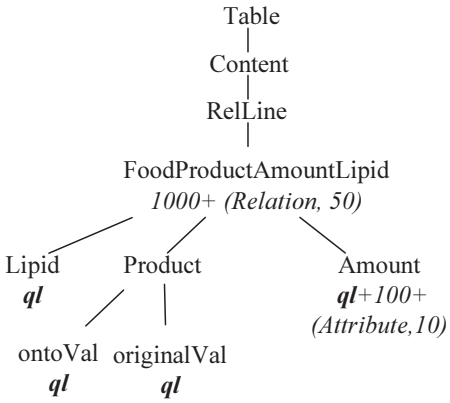


Fig. 5.8 – Un exemple d'arbre XML représentant une vue dans le sous-système XML. Les nœuds interrogeables (*ql*) sont indiqués en italique gras. Un coût de suppression et un couple (renommage, coût de renommage) sont associés à certains nœuds de l'arbre.

Un arbre requête est construit à partir d'un arbre vue dans lequel l'utilisateur a spécifié d'une part, parmi l'ensemble des nœuds interrogeables de la vue, les nœuds de projection et les couples (nœud de sélection, valeur de sélection) de la requête et d'autre part, un coût de transformation maximum de la requête. L'utilisateur a la possibilité, s'il le désire, de modifier les coûts de transformation (suppression, renommage) définis dans la vue.

Définition 51 Une requête conforme à un arbre de type (a_t, e_t) est un n-uplet Q=(a_Q, e_Q, w_Q, c_Q, ct_{max}, p_Q, s_Q, ws_Q) où:

- (a_Q, e_Q, w_Q, c_Q) est une vue conforme à (a_t, e_t);
- ct_{max} est le coût de transformation maximum de la requête;
- p_Q est la fonction qui associe la valeur *pl* aux nœuds interrogeables de la vue considérés comme les *nœuds de projection* de la requête;
- s_Q est la fonction qui associe la valeur *sl* aux nœuds interrogeables de la vue considérés comme les *nœuds de sélection* de la requête;

– ws_Q est la fonction qui associe une valeur de sélection à chaque nœud de sélection, cette valeur pouvant être précise ou floue. Dans ce dernier cas, elle est représentée par un arbre de données de racine CFS ou DFS (cf définitions 43 et 44).

Exemple 42 *La requête Q de la figure 5.9 indique que l'utilisateur souhaite obtenir le nom du produit, la quantité du produit et la quantité de lipide dans la vue présentée dans la figure 5.8 utilisant la relation* FoodProductAmountLipid. *La valeur floue associée au critère de sélection* Product.originalVal *peut être interprété de la manière suivante:* Beef carcass *est le produit recherché en priorité,* Tree onion *est également accepté mais avec un intérêt moindre. La valeur floue associée au critère de sélection* Amount *peut être interprété de la manière suivante:* 100g *est la taille d'échantillon recherchée en priorité,* 250g *est également accepté mais avec un intérêt moindre. Le coût de transformation maximum tc_{max} vaut 500.*

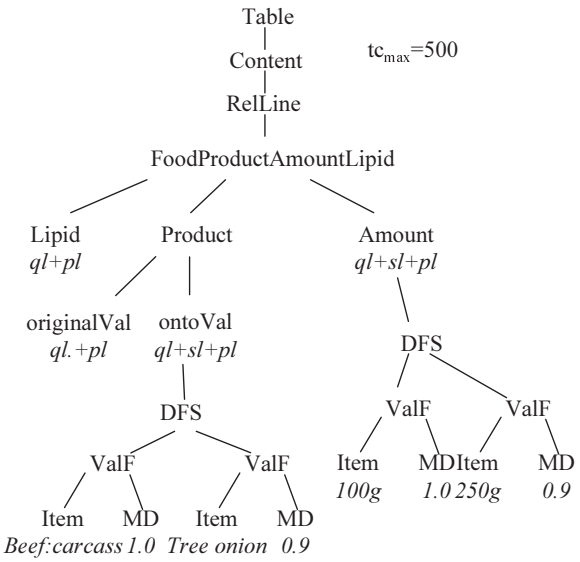


Fig. 5.9 – Un exemple d'arbre requête instancié dans la vue présentée dans la figure 5.8.

5.3.2 Notion de requête approximative dans l'adapteur du sous-système XML

Un ensemble de requêtes approximatives est généré à partir d'un arbre requête en respectant les deux règles suivantes: (*i*) chaque nœud, intermédiaire ou feuille, de la requête initiale auquel est associé un coût de suppression peut être supprimé, (*ii*) chaque nœud, intermédiaire ou feuille, de la requête initiale auquel est associé un ou plusieurs couples (renommage, coût de renommage) peut être renommé. Chaque requête approximative générée doit conserver au moins un attribut de sélection et un attribut de projection de la requête initiale, les deux n'étant pas forçément distincts. Par ailleurs, le coût de transformation de la requête initiale est calculé pour

chaque requête approximative générée. Ce coût est égal à la somme des coûts de suppression des nœuds supprimés et des coûts de renommage des nœuds renommés. Seules les requêtes approximatives ayant un coût de transformation inférieur au coût de transformation maximum associé à la requête initiale sont conservées.

Définition 52 Une requête approximative générée à partir d'une requête Q=(a_Q, e_Q, w_Q, c_Q, ct_{max}, p_Q, s_Q, ws_Q) est un n-uplet A=(a_A, e_A, w_A, p_A, s_A, ws_A, ct_A) où:

- il existe un *homomorphisme de structure faible* h des nœuds de a_Q vers les nœuds de a_A: (i) h préserve la racine de a_Q: $h(\text{racine}(a_Q)) = \text{racine}(a_A)$ et (ii) h préserve la relation ascendant-descendant de a_Q: pour tout nœud m descendant du nœud n dans a_Q, $h(m)$ est un descendant de $h(n)$ dans a_A;
- e_A est une fonction d'étiquetage qui associe à chaque nœud n de a_A, soit l'étiquette qui est associée par e_Q au nœud antécédent $h^{-1}(n)$, soit une valeur de renommage définie dans les coûts de transformation $c_Q(h^{-1}(n))$ du nœud antécédent $h^{-1}(n)$;
- pour tout nœud feuille n de a_A correspondant à un attribut interrogeable dans a_Q ($w_Q(h^{-1}(n))$ = ql), alors $w_A(n) = ql$;
- pour tout nœud feuille n de a_A correspondant à un attribut de projection dans a_Q ($p_Q(h^{-1}(n)) = pl$), alors $p_A(n) = pl$. n est dit *nœud de projection* de la requête. De plus, il doit exister au moins un nœud feuille n de a_A tel que $p_A(n) = pl$;
- pour tout nœud feuille n de a_A correspondant à un attribut de sélection dans a_Q ($s_Q(h^{-1}(n))$ = sl), alors $s_A(n) = sl$. n est dit *nœud de sélection* de la requête. De plus, il doit exister au moins un nœud feuille n de a_A tel que $s_A(n) = sl$;
- pour tout nœud feuille n de a_A tel que $ws_Q(h^{-1}(n))$ est défini, alors $ws_A(n) = ws_Q(h^{-1}(n))$;
- ct_A est le coût de transformation de la requête Q pour obtenir la requête approximative A: $ct_A = \sum_{i=1}^{l} cs_{n_s^i} + \sum_{i=1}^{q} cr_{n_r^i}$ où $\{n_s^1, \ldots, n_s^l\}$ représente la liste des nœuds supprimés[3] de a_Q dans a_A et $\{n_r^1, \ldots, n_r^q\}$ représente la liste des nœuds renommés[4] de a_Q dans a_A, $cs_{n_s^i}$ représente le coût de suppression du nœud n_s^i et $cr_{n_r^i}$ le coût de renommage du nœud n_i^r. De plus, ct_A doit être plus petit ou égal au coût de transformation maximum ct_{max} de la requête Q.

Exemple 43 *La figure 5.10 présente deux exemples de requêtes approximatives générées à partir de la requête Q de la figure 5.9. La première requête approximative (à gauche) a été générée par renommage du nœud* Amount *en* Attribute*, la seconde (à droite) par suppression du nœud* Amount*.*

Remarque 6 *Dans un arbre XML flou, la représentation d'une valeur floue n'est pas considérée comme faisant partie de la structure de l'arbre (cf définition 45). Cela permet de définir de manière homogène la suppression d'un nœud feuille qu'il soit précis ou flou. Dans les deux cas,*

[3] la liste des nœuds de a_Q tels que n_s^i n'a pas d'image dans a_A par h

[4] la liste des nœuds de a_Q tels que n_r^i a une image dans a_A par h et $e_Q(n_r^i) \neq e_A(h(n_r^i))$

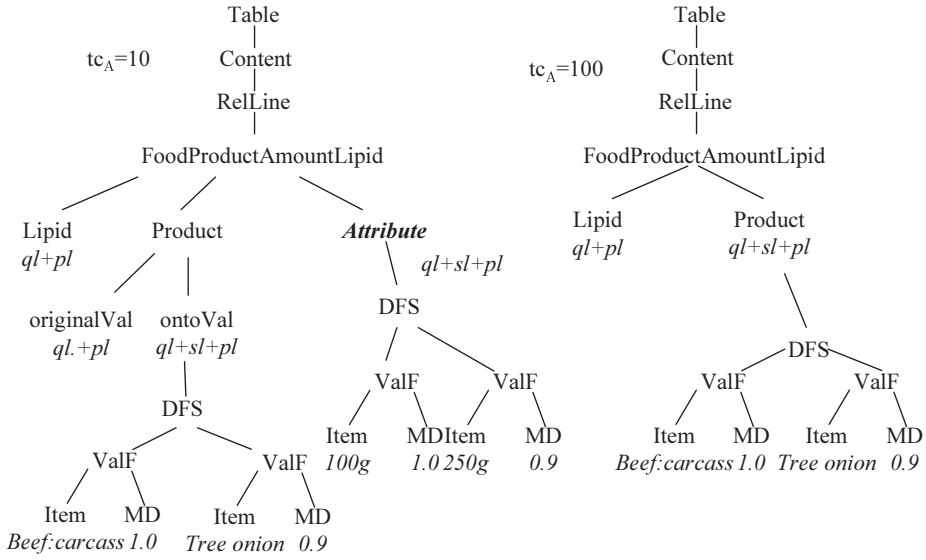


Fig. 5.10 – Deux exemples de requêtes approximatives générées à partir de l'arbre requête de la figure 5.9.

le nœud et sa valeur associée sont supprimés. Par exemple, dans la requête 5.9, le sous-arbre du nœud Amount *de racine DFS ne fait pas partie de la structure de l'arbre (définie par la fonction a_Q). Ainsi, la suppression du nœud sélection* Amount *pour obtenir la requête approximative de gauche dans la figure 5.10 entraîne automatiquement la suppression de la valeur floue qui lui est associée par la fonction ws_Q.*

5.3.3 Réponses à une requête dans l'adapteur du sous-système XML

Deux types de réponse à une requête Q sont définis dans l'adapteur du sous-système XML: une réponse exacte et une réponse approximative.

Réponse exacte La réponse exacte à une requête Q (i) satisfait tous les critères de sélection de Q, et (ii) associe une valeur à chaque nœud de projection de Q. La recherche de réponses exactes à une requête Q est effectuée par une valuation de Q à partir des arbres de données flous de la base.

Définition 53 Soit Q=(a_Q, e_Q, c_Q, ct_{max}, w_Q, p_Q, s_Q, ws_Q) une requête conforme à un arbre de type T=(a_t, e_t) et D=(a_D, e_D, v_D) un arbre de données flou instance de T. Une *valuation* de Q à partir de D est définie par une fonction σ_D de a_Q vers a_D telle que:

– σ_D est un homomorphisme de type strict[5] de (a_Q, e_Q) vers (a_D, e_D);

[5]cf définition 42 pour la définition d'un homomorphisme de type strict

– σ_D satisfait chaque nœud de sélection n_s^i, $i \in [1, m]$, de Q avec le degré de possibilité $\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i)))$ et avec le degré de nécessité $N(ws_Q(n_s^i), v_D(\sigma_D(n_s^i)))$.

L'adéquation de l'arbre de données flou D avec la requête Q selon la valuation σ_D est définie par le couple de valeurs $(ad_{\Pi(Q)}, ad_{N(Q)})$ tel que $ad_{\Pi(Q)}=min_{i\in[1,m]}(\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i))))$ et $ad_{N(Q)}=min_{i\in[1,m]}(N(ws_Q(n_s^i), v_D(\sigma_D(n_s^i))))$.

La réponse à une requête Q est un ensemble de tuples, chaque tuple correspond à une valuation de Q par rapport à un arbre de données flou D. Il est composé d'un couple de degrés d'adéquation de D avec la requête Q et d'un ensemble de valeurs associées aux nœuds de projection de la requête.

Définition 54 Une *réponse* à une requête Q=(a_Q, e_Q, c_Q, ct_{max}, w_Q, p_Q, s_Q, ws_Q) composée de m' nœuds de projection notés $n_p^1, \ldots, n_p^{m'}$ dans une base de données XML $\mathcal{W}$ est un ensemble de tuples, chaque tuple étant défini comme suit: $\{ \cup_{i=1}^{m'} v_D(\sigma_D(n_p^i)) \cup ad_{\Pi(Q)} \cup ad_{N(Q)} \mid D$ étant un arbre de données flou de $\mathcal{W}$ et σ_D une valuation de Q par rapport à $D\}$.

Exemple 44 *La réponse à la requête Q de la figure 5.9 suite à l'interrogation de l'arbre de données de la figure 5.11 est la suivante: { Product.ontoVal=Tree onion, Product.OriginalVal=Red Onion, Lipid=10g, Amount=100g, ad_Π=0.9, ad_N=0.9 }.*

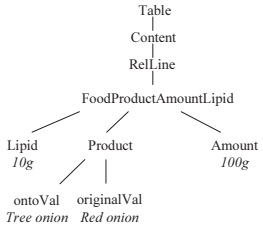


Fig. 5.11 – Un exemple d'arbre de données permettant d'obtenir une réponse exacte à la requête de la figure 5.9.

Réponse approximative La réponse approximative A à une requête Q est définie comme l'union des réponses aux requêtes approximatives générées à partir de Q. Nous avons vu que la génération des requêtes approximatives repose sur la suppression et le renommage de nœuds de Q. Dans cette section, nous allons voir que la recherche de réponses à une requête approximative A autorise l'insertion de nœuds dans A. Une réponse à une requête approximative A (i) satisfait tous les critères de sélection de A, et (ii) associe une valeur constante à chaque nœud de projection de A. La recherche de réponses à une requête approximative est effectuée par une valuation de la requête à partir des arbres de données flous de la base.

Définition 55 Soit A=(a_A, e_A, w_A, p_A, s_A, ws_A, ct_A) une requête approximative générée à partir de Q=(a_Q, e_Q, w_Q, c_Q, ct_{max}, p_Q, s_Q, ws_Q) conforme à un arbre de type T=(a_t, e_t) et D=(a_D, e_D, v_D) un arbre de données flou de la base. Une *valuation* de A par rapport à D est définie par une fonction σ_D de a_A vers a_D telle que:

- σ_D est un *homomorphisme de type faible* de (a_A, e_A) vers (a_D, e_D): c'est-à-dire un homomorphisme de structure faible (cf définition 52) préservant les étiquettes de a_A;
- σ_D satisfait chaque nœud de sélection n_s^i, $i \in [1, m]$, de A avec le degré de possibilité $\Pi(ws_A(n_s^i), v_D(\sigma_D(n_s^i)))$ et le degré de nécessité $N(ws_A(n_s^i), v_D(\sigma_D(n_s^i)))$.

L'adéquation de l'arbre de données flou D avec la requête approximative A selon la valuation σ_D est définie par le couple de valeurs ($ad_{\Pi(A)}$, $ad_{N(A)}$) tel que $ad_{\Pi(A)} = min_{i\in[1,m]}(\Pi(ws_A(n_s^i), v_D(\sigma_D(n_s^i))))$ et $ad_{N(A)} = min_{i\in[1,m]}(N(ws_A(n_s^i), v_D(\sigma_D(n_s^i))))$.

Une réponse à une requête approximative est un ensemble de tuples, chaque tuple étant composé: (i) d'un ensemble de valeurs associées à chaque nœud de projection, (ii) de l'adéquation de la réponse à la requête approximative, et (iii) du coût de transformation de la requête initiale pour obtenir la requête approximative.

Définition 56 Une *réponse* à une requête approximative A=(a_A, e_A, w_A, p_A, s_A, ws_A, ct_A) composée de m' nœuds de projection n_p^1, ..., $n_p^{m'}$ dans une base de données XML $\mathcal{W}$ est un ensemble de tuples, chaque tuple étant défini comme suit: { $\cup_{i=1}^{m'} v_D(\sigma_D(n_p^i)) \cup ad_{\Pi(A)} \cup ad_{N(A)} \cup ct_A \mid D$ étant un arbre de données flou de $\mathcal{W}$ et σ_D une valuation de A par D}.

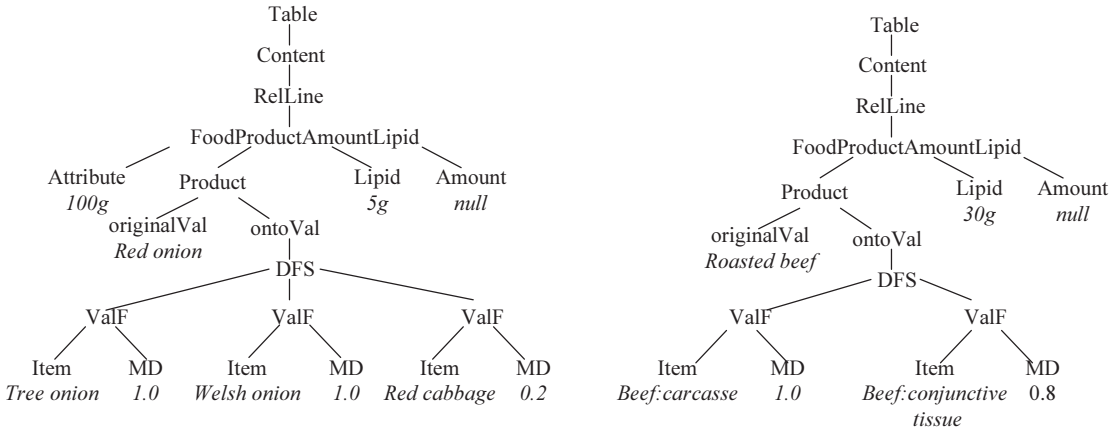


Fig. 5.12 – Exemples d'arbres de données flous XML contenant des éléments feuilles ($ontoVal$) dont la valeur associée est un arbre de données de racine DFS.

Exemple 45 *Pour faciliter l'interprétation des résultats, les réponses sont ordonnées d'abord par coûts de transformation croissants (tc_A) et ensuite par degrés d'adéquation décroissants ($ad_{\Pi(A)}$, $ad_{N(A)}$). La réponse à la requête approximative en partie gauche de la figure 5.10 après*

interrogation des arbres de données flous de la figure 5.12 est la suivante: {(Product.originalVal=Red onion, Product.ontoVal=1.0/Tree onion+1.0/Welsh onion+0.2/Red cabbage, Attribute=100g, Lipid=5g, ad_Π=0.9, ad_N=0, tc_A=10)}. La réponse à la requête approximative en partie droite de la figure 5.10 après interrogation des arbres de données flous de la figure 4.14 est la suivante: {(Product.originalVal= Roasted Beef, Product.ontoVal=1.0/Beef:carcass+0.8/Beef:conjunctive tissue, Lipid=30g, ad_Π=1.0, ad_N=0.2, ct_A=100), (Product.originalVal=Red onion, Product.ontoVal = 1.0/Tree onion+ 1.0/Welsh onion+ 0.2/Red cabbage, Lipid=5g, ad_Π=0.9, ad_N=0, ct_A=100) }.

5.4 Conclusion du chapitre

Dans ce chapitre, nous avons présenté les deux adapteurs de requête MIEL faisant partie respectivement du sous-système graphe conceptuel et du sous-système XML. Ils permettent d'interroger les données faiblement structurées, internes et externes de notre système d'intégration. Une originalité partagée par les deux adapteurs est leur capacité à effectuer une comparaison souple entre un critère de sélection flou et une donnée imprécise stockée en base. Une deuxième originalité spécifique au sous-système XML est sa capacité à combiner la flexibilité procurée par l'utilisation des critères de sélection flous et celle apportée par son système de dérivation de requêtes approximatives.

L'adapteur du sous-système graphe conceptuel a été implémenté en C++ sous Linux. Cette implémentation repose sur une extension de la plateforme CoGITaNT (Genest & Salvat (1998)) représentant environ 5000 lignes de code. Cette implémentation a été testée sur une base de 150 graphes conceptuels dans le domaine du risque alimentaire. Les utilisateurs ont été très impressionnés par la souplesse et la simplicité d'utilisation d'une part de l'interface graphique qui permet de saisir des graphes conceptuels sous CoGITaNT et d'autre part, par l'interface graphique du système d'interrogation MIEL qui restitue dans deux onglets séparés les données provenant du sous-système relationnel et du sous-système graphe conceptuel (voir figures 5.13 et 5.14). Nous proposerons dans un avenir proche des outils complémentaires pour aider l'utilisateur à construire un graphe conceptuel. Nous travaillerons à la conception d'une interface graphique proposant des graphes schémas pré-existants que les utilisateurs n'auront plus qu'à modifier et combiner.

L'adapteur du sous-système XML a été implémenté en Java en utilisant le processeur XQuery Saxon (www.saxonica.com). Cette implémentation représente environ 8000 lignes de code. Elle a été validée sur une base de 196 documents SML flous. Quelques tests préliminaires d'interrogation ont été réalisés. Ils ont porté sur trois requêtes différentes élaborées à partir de trois relations sémantiques de l'ontologie. Sur les 93 réponses obtenues pour l'ensemble des trois requêtes, 66 ont été jugées pertinentes. Parmi ces dernières, 62 sont obtenues à partir de requêtes approximatives, et seulement 4 à partir des requêtes initiales. Ces premiers résultats, bien que n'étant pas significatifs d'un point de vue statistique, illustrent bien la nécessité de

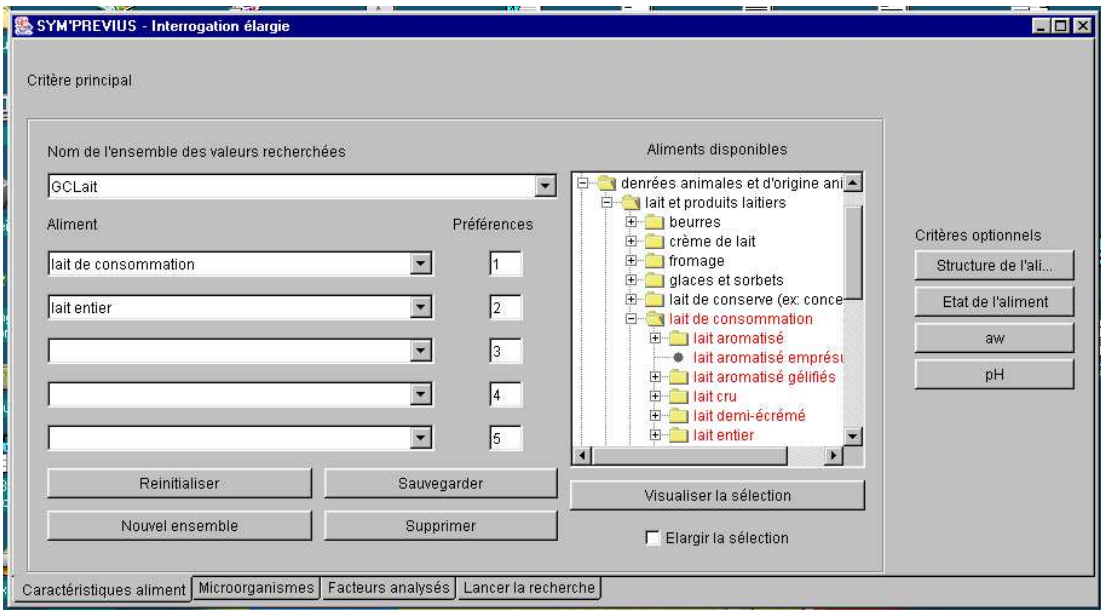


Fig. 5.13 – Requête MIEL interrogeant les sous-systèmes relationnels et graphes conceptuels sur le lait de consommation et le lait entier.

poursuivre notre effort de recherche concernant l'interrogation flexible de documents SML flous. Dans le cadre de la thèse de Gaëlle Hignette et du projet WebContent, nous serons amenés à étendre prochainement l'adapteur du sous-système XML pour prendre en compte les nouvelles annotations sémantiques floues que nous avons évoquées en perpective du chapitre 4.

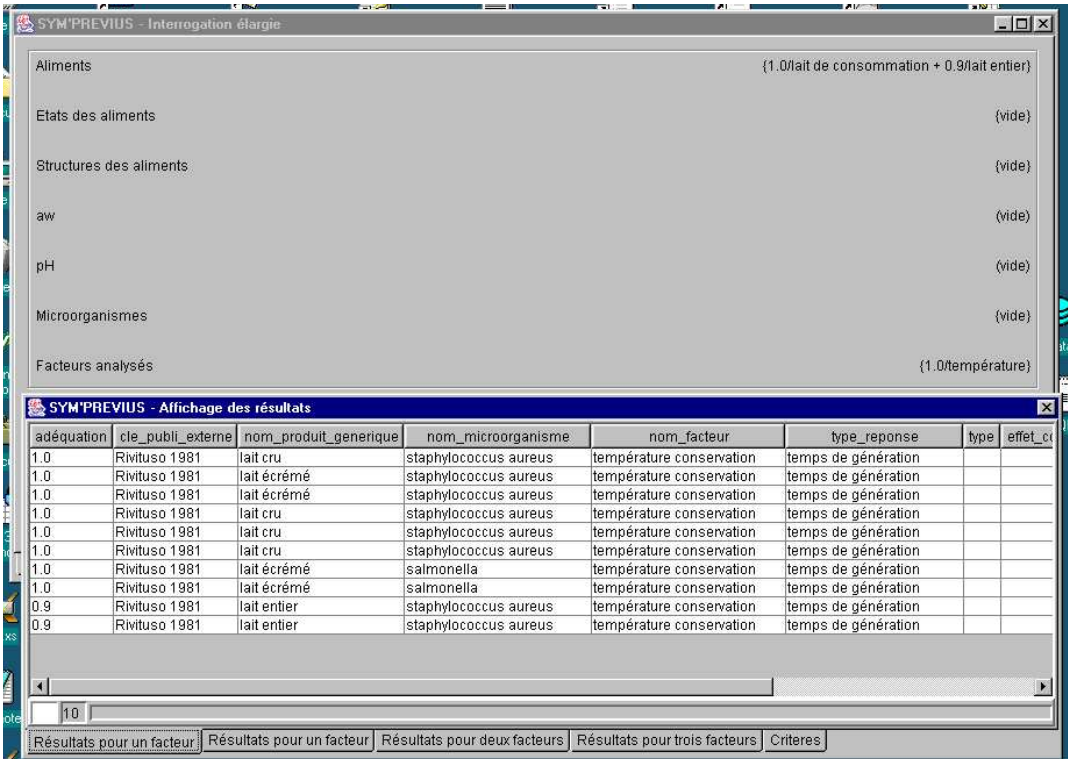


Fig. 5.14 – Réponses obtenues à la requête de la figure 5.13: le premier onglet présente les réponses obtenues par le sous-système graphes conceptuels. Les réponses sont triées par degrés d'adéquation décroissants (seul le degré de possibilité est affiché sur cet écran). Les onglets suivants contiennent les réponses provenant du sous-système relationnel.

Chapitre 6

Conclusion et perspectives

Nous avons présenté dans ce mémoire un système d'intégration de données internes et externes basé sur une architecture de type médiateur. La conception de ce système a été influencée par plusieurs caractéristiques des données à intégrer. Ces données peuvent être de format hétérogène, elles peuvent être imprécises et elles sont relativement rares. L'une de nos contributions dans ce travail a été d'étendre des modèles de représentation de données faiblement structurés comme le modèle des graphes conceptuels et le modèle XML en utilisant des concepts issus de la théorie de la logique floue et de la théorie des possibilités pour prendre en compte les caractéristiques des données. Cette contribution confère à notre système d'intégration de données son originalité: il supporte la représentation de données imprécises et il propose des mécanismes d'interrogation flexible. D'autre part, nous avons cherché à rendre ce travail générique en regroupant les connaissances spécifiques du domaine d'application dans l'ontologie du système d'intégration.

Une partie de cette ontologie est constituée d'une taxonomie de termes spécifiques au domaine. Ceci nous a amenés à étudier les spécifités des sous-ensembles flous dont le domaine de valeurs est organisé en taxonomie. Notre réflexion a d'abord été menée dans le cadre du modèle des graphes conceptuels car la taxonomie des termes est incluse dans le support terminologique, élément central de ce modèle de représentation de connaissances. Puis, nous avons défini le concept de sous-ensemble flou hiérarchique, indépendant d'un formalisme de représentation particulier, qui est à notre connaissance une contribution originale de ce travail. Nous avons ensuite utilisé ce concept dans le cadre du modèle relationnel et du modèle XML.

Nous avons étendu le modèle des graphes conceptuels pour pouvoir représenter des données imprécises. A la différence des travaux précédents qui ont privilégié la richesse de l'expressivité, nous avons cherché dans notre extension à préserver l'homogénéité du modèle en proposant une expressivité moins ambitieuse, mais non ambiguë et suffisante pour traiter nos applications-cibles. Cette extension a porté uniquement sur la représentation des sommets concepts dans lesquels des sous-ensembles flous définis sur le support terminologique du modèle peuvent être introduits. Nous avons proposé des mécanismes d'interrogation flexible basés sur le test de la relation de subsomption entre deux graphes conceptuels (représentant respectivement une

requête et une donnée). Ce test est exécuté en temps polynômial sous des restrictions acceptables pour nos applications-cibles.

Nous avons ensuite étendu le modèle XML pour représenter des données imprécises modélisées par des arbres de données flous. De nouveau, notre extension est moins expressive que celles proposées dans des travaux comparables. Mais, à la différence de ces travaux, notre originalité provient du fait que nous l'avons exploitée dans un mécanisme d'interrogation combinant la flexibilité apportée d'une part par l'utilisation des sous-ensembles flous dans les critères de sélection et d'autre part par la relaxation de la structure de la requête. Cette extension a été utilisée pour représenter des annotations sémantiques de tableaux de données issus du Web.

Le travail présenté dans ce mémoire se situe dans le domaine de la recherche en informatique appliquée. Les concepts proposés dans ce mémoire ont été réellement éprouvés sur de véritables applications. Notre système d'intégration a été utilisé dans deux applications concernant le risque alimentaire. L'implémentation du système d'intégration de données chimiques de l'unité Mét@risk étant actuellement en cours, l'implémentation la plus avancée concerne le système Sym'Previus dont l'objectif est l'évaluation de la contamination microbiologique des aliments. Dans cette implémentation, les trois sous-systèmes ont été complètement réalisés et le sous-sytème relationnel est actuellement mis en production sur le site du GIS Sym'Previus. Des expérimentations ont été menées, notamment concernant le sous-système relationnel et le sous-système XML. Cette confrontation de notre système d'intégration à des applications réelles n'est pas qu'une obligation institutionnelle, elle a guidé notre recherche et nous a permis d'en identifier les axes prioritaires. De ce point de vue, l'extraction d'information pertinente à partir de documents scientifiques complexes nous apparaît comme l'un des goulots d'étranglement importants dans les applications que nous avons traitées. Cette tâche requiert un volume important de main d'œuvre qualifiée pour effectuer un travail de saisie fastidieux. Ce constat conforte l'orientation que nous avons prise dans le projet e.dot: nous nous sommes intéressés d'une part à l'annotation sémantique semi-automatique de tableaux de données guidée par une ontologie du domaine et d'autre part à l'interrogation de ces documents annotés. Le degré d'automatisation de ce travail d'extraction doit être encore augmenté pour que l'on puisse l'utiliser dans des applications réelles.

Ce dernier point est l'une de nos perpectives immédiates. Dans le prolongement des travaux présentés dans ce mémoire, nous avons répondu en Juillet 2005 à un appel d'offre de l'ANR (Agence Nationale de la Recherche) dans le cadre du RNTL pour la création d'une plateforme technologique dans le domaine du Web sémantique. Ce projet, baptisé WebContent (http://www.webcontent.fr/), regroupe une vingtaine de partenaires d'origine académique (INRIA, LRI, LIP6, INRA, ...) et industrielle (CEA, EADS, Thales, Bongrain, Xyleme, Exalead, ...). Il a été labellisé par le RNTL fin 2005 et a démarré en 2006 pour une durée de 3 ans. L'objectif du projet est de proposer une plate-forme flexible et générique de gestion de contenus dont le but est d'intégrer des technologies du Web sémantique et de démontrer leur efficacité sur des applications réelles à fort enjeu économique ou sociétal. Plusieurs applications-cibles ont

été identifiées dans le domaine de la veille (veille économique en aéronautique, veille technologique dans le domaine du risque alimentaire, ...). Dans le cadre de ce projet et de la thèse de Gaëlle Hignette, nous allons proposer une amélioration de la méthode d'annotation sémantique de tableaux guidée par l'ontologie du domaine qui a été conçue dans le projet RNTL/e.dot. Plusieurs questions vont être abordées dans un futur proche. Nous entamerons une nouvelle réflexion sur la sémantique des sous-ensembles flous que nous manipulons dans les annotations sémantiques. Nous considérons pour l'instant que le terme de l'ontologie le plus représentatif d'un terme du web est une valeur qui existe, mais que l'on ne connaît pas de façon certaine : les degrés de similarité entre termes de l'ontologie et termes du Web représentent la possibilité qu'un terme de l'ontologie soit effectivement le terme le plus représentatif du terme du Web. Or, la valeur du degré de similarité est une indication en elle-même de la qualité de l'annotation, et on ne veut pas perdre cette information en normalisant le sous-ensemble flou. Il nous apparaît aujourd'hui mieux adapté de représenter la liste des termes de l'ontologie les plus proches d'un terme du tableau du Web comme un sous-ensemble flou représentant des similarités. La notion de similarité est d'ailleurs l'une des trois sémantiques les plus couramment associées aux sous-ensembles flous (cf Dubois & Prade (1997)). Nous serons donc amenés à réfléchir à toutes les conséquences qu'impliquera ce choix et en particulier en ce qui concerne l'interrogation des documents annotés. Par ailleurs, nous travaillerons sur l'enrichissement de l'ontologie afin de pouvoir catégoriser à la fois les colonnes symboliques et numériques des tableaux de données du Web. Nous étudierons également comment représenter l'incertitude associée à l'instanciation des relations sémantiques de l'ontologie, notamment lorsque cette instanciation est partielle. Puis, nous étendrons l'adapteur du sous-système XML pour prendre en compte ces nouvelles annotations sémantiques floues.

Enfin nous envisageons d'ouvrir notre architecture afin d'y intégrer d'autres sources de données construites autour d'ontologies différentes de celle de notre système d'intégration de données. Nous nous intéresserons pour cela à la problématique de la mise en correspondance d'ontologies. Ce type d'intégration est l'un des problèmes majeurs à résoudre afin de pouvoir exploiter au mieux les informations disponibles sur le Web.

Table des figures

Liste des tableaux

Bibliographie

Abiteboul, S. 1997 Querying Semi-Structured Data. *ICDT* (ed. F. N. Afrati & P. G. Kolaitis), *Lecture Notes in Computer Science*, vol. 1186, pp. 1–18. Springer.

Abiteboul, S., Agrawal, R., Bernstein, P. A., Carey, M. J., Ceri, S., Croft, W. B., DeWitt, D. J., Franklin, M. J., Garcia-Molina, H., Gawlick, D., Gray, J., Haas, L. M., Halevy, A. Y., Hellerstein, J. M., Ioannidis, Y. E., Kersten, M. L., Pazzani, M. J., Lesk, M., Maier, D., Naughton, J. F., Schek, H.-J., Sellis, T. K., Silberschatz, A., Stonebraker, M., Snodgrass, R. T., Ullman, J. D., Weikum, G., Widom, J. & Zdonik, S. B. 2005 The Lowell database research self-assessment. *Commun. ACM* **48** (5), 111–118.

Abiteboul, S., Quass, D., McHugh, J., Widom, J. & Wiener, J. L. 1997 The Lorel Query Language for Semistructured Data. *Int. J. on Digital Libraries* **1** (1), 68–88.

Aguiléra, V., Cluet, S., Vetri, P., Vodislav, D. & Wattez, F. 2000 Querying the XML Documents on the Web. *Proceedings of the ACM SIGIR Workshop on XML and I.R.*. Athens.

Amer-Yahia, S., Cho, S. & Srivastava, D. 2002 Tree Pattern Relaxation. *Proceedings of the 8th International Conference EDBT*.

Astrahan, M. M., Blasgen, M. W., Chamberlin, D. D., Eswaran, K. P., Gray, J., Griffiths, P. P., III, W. F. K., Lorie, R. A., McJones, P. R., Mehl, J. W., Putzolu, G. R., Traiger, I. L., Wade, B. W. & Watson, V. 1976 System R: Relational Approach to Database Management. *ACM Trans. Database Syst.* **1** (2), 97–137.

Baumgartner, R., Flesca, S. & Gottlob, G. 2001 Visual Web Information Extraction with Lixto. *VLDB* (ed. P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao & R. T. Snodgrass), pp. 119–128. Morgan Kaufmann.

Bidault, A., Froidevaux, C. & Safar, B. 2000 Repairing queries in a mediator approach. *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pp. 406–410.

Bosc, P., HadjAli, A. & Pivert, O. 2004*a* Fuzzy Closeness Relation as a Basis for Weakening Fuzzy Relational Queries. *FQAS* (ed. H. Christiansen, M.-S. Hacid, T. Andreasen & H. L. Larsen), *Lecture Notes in Computer Science*, vol. 3055, pp. 41–53. Springer.

Bosc, P., Lietard, L. & Pivert, O. 1994 Soft querying, a new feature for database management system. *Proceedings DEXA'94 (Database and EXpert system Application), Lecture Notes in Computer Science #856*, pp. 631–640. Springer-Verlag.

Bosc, P., Liétard, L., Pivert, O. & Rocacher, D. 2004*b* *Gradualité et imprécision dans les bases de données : ensembles flous, requêtes flexibles et interrogation de données mal connues*. Ellipses, Technosup.

Bosc, P. & Pivert, O. 1992 Some Approaches for Relational Databases Flexible Querying. *J. Intell. Inf. Syst.* **1** (3/4), 323–354.

Bouchon-Meunier, B. & Yao, J. 1992 Linguistic modifiers and imprecise categories. *International Journal of Intelligent Systems* **7**, 25–36.

Braga, D., Campi, A., Damiani, E., Pasi, G. & Lanzi, P. 2002 FXPath: Flexible Querying of XML Documents. *Proc. of EuroFuse 2002* .

Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E. & Yergeau, F. 2004 *Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation 04 February 2004. http://www.w3.org/TR/REC-xml/*.

Brouard, C. 2000 Construction et exploitation de Réseaux Sémantiques Flous pour l'Extraction d'Information Pertinente : le système RELIEFS. PhD thesis, Thèse de doctorat, Université Paris 6.

Buche, P., Dervin, C., Haemmerlé, O. & Thomopoulos, R. 2005*a* Fuzzy querying of incomplete, imprecise and heterogeneously structured data in the relational model using ontologies and rules. *IEEE Transactions on Fuzzy Systems* **13** (3), 373–383.

Buche, P., Dibie-Bartélemy, J., Haemmerlé, O. & Hignette, G. 2006*a* Fuzzy semantic tagging and flexible querying of XML documents extracted from the Web. *Journal of Intelligent Information Systems* **26** (1), 25–40.

Buche, P., Dibie-Bartélemy, J., Haemmerlé, O. & Thomopoulos, R. 2005*b* A Data Warehouse that Gathers Several Formalisms to Capture Data Heterogeneity and Incompleteness in the Field of Food Microbiological Safety. *1ère Journée Francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA 2005)*, pp. 109–121. Lyon, France: RNTI-B-1. Cépaduès Editions.

Buche, P., Dibie-Bartélemy, J., Haemmerlé, O. & Thomopoulos, R. 2006*b* Fuzzy concepts applied to the design of a database in predictive microbiology. *Fuzzy Sets and Systems* **157**, 1188–1200.

Buche, P., Dibie-Barthélemy, J., Haemmerlé, O. & Houhou, M. 2004 Towards flexible querying of XML imprecise data in a dataware house opened on the Web. *Proceedings of the 6th International Conference Flexible Querying Answering Systems, FQAS 2004*, pp. 28–40. Lyon, France: Lecture Notes in AI #3055, Springer.

Buche, P., Dibie-Barthélemy, J., Haemmerlé, O. & Thomopoulos, R. 2006*c* The MIEL++ architecture: when RDB, CGs and XML meet for the sake of risk assessment in food products. *Proceedings of the 14th International Conference on Conceptual Structures, ICCS'2006*, pp. 158–171. Aalborg, Danemark: LNCS 4068.

Buche, P., Dibie-Barthélemy, J. & Wattez, F. 2006*d* Approximate querying of XML fuzzy data. *Proceedings of the 7th International Conference Flexible Querying Answering Systems, FQAS 2006*, pp. 26–38. Milan, Italie: LNAI 4027.

Buche, P. & Haemmerlé, O. 2000*a* Towards a unified querying system of both structured and semi-structured imprecise data using fuzzy views. *Proceedings of the 8th International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence #1867*, pp. 207–220. Darmstadt, Germany: Springer-Verlag.

Buche, P. & Haemmerlé, O. 2000*b* Towards category-based fuzzy querying of both structured and semi-structured imprecise data. *Proceedings of the Fourth International Conference on Flexible Query Answering Systems (FQAS'2000)*, pp. 362–375. Warsaw, Poland: Springer-Verlag.

Buche, P. & Haemmerlé, O. 2000 Vers un système unifié d'interrogation de données imprécises structurées et semi-structurées utilisant des vues floues. *INFORSID*, pp. 174–189.

Buche, P., Haemmerlé, O. & Thomopoulos, R. 2001 Representation of weakly structured imprecise data for fuzzy querying. *Proceedings of the twentieth NAFIPS (North American Fuzzy Information Processing System)*, pp. 2126–2131. Vancouver, Canada: IEEE.

Buche, P., Haemmerlé, O. & Thomopoulos, R. 2003 Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk assessment. *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS'2003*, pp. 98–107. Maebashi, Japan: Lecture Notes in AI #2871, Springer.

Buneman, P., Davidson, S. B., Hillebrand, G. G. & Suciu, D. 1996 A Query Language and Optimization Techniques for Unstructured Data. *SIGMOD Conference* (ed. H. V. Jagadish & I. S. Mumick), pp. 505–516. ACM Press.

Bush, V. 1945 As we may think. *Atlantic Monthly* pp. 101–108.

Cao, T. 1999 Foundations of Order-Sorted Fuzzy Set Logic Programming in Predicate Logic and Conceptual Graphs. PhD thesis, University of Queensland, Australia.

Cao, T. & Creasy, P. 1998 Fuzzy order-sorted logic programming in conceptual graphs with a sound and complete proof procedure. *Proceedings of the 6th International Conference on Conceptual Structures, ICCSŠ98*, pp. 270–284. Montpellier, France: Lecture Notes in Artificial Intelligence # 1453, Springer.

Chaudhuri, S. & Dayal, U. 1997 An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record* **26** (1), 65–74.

Chein, M. & Mugnier, M. L. 1992 Conceptual graphs: fundamental notions. *Revue d'Intelligence Artificielle* **6** (4), 365–406.

Ciravegna, F. 2001 Adaptive Information Extraction from Text by Rule Induction and Generalisation. *IJCAI* (ed. B. Nebel), pp. 1251–1256. Morgan Kaufmann.

Codd, E. 1970 A relational model for large shared data banks. *Communications of the ACM* **13** (6), 377–387.

Codd, E. F. 1979 Extending the Database Relational Model to Capture More Meaning. *ACM Trans. Database Syst.* **4** (4), 397–434.

Coleman, M. 2003 Guest editorial : interactions of predictive microbiology and risk assessment. Risk Analysis. *Risk Analysis* **23**, 175–178.

Damiani, E. & Tanca, L. 2000 Blind queries to XML data. *DEXA'00*, pp. 345–356. Springer Verlag.

De Cock, M., S., G. & Nikravesh, M. 2004 Fuzzy thesauri for and from the www. *In Nikravesh, M., Zadeh, L., Kacprzyk, J., eds.: Soft Computing for Information Processing and Analysis* pp. 275–284.

Dubois, D. & Prade, H. 1980 *Fuzzy Sets and Systems-Theory and applications*. Academic Press.

Dubois, D. & Prade, H. 1985 *Théorie des possibilités: Application à la représentation des connaissances en informatique*. Masson.

Dubois, D. & Prade, H. 1986 Weighted minimum and maximum operations in fuzzy set theory. *Inf. Sci.* **39** (2), 205–210.

Dubois, D. & Prade, H. 1988 *Possibility Theory - An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press.

Dubois, D. & Prade, H. 1995 *Fuzziness in Database Management Systems, P. Bosc and J. Kacprzyk eds.*, chap. Tolerant fuzzy pattern matching : an introduction, pp. 42–58. Heidelberg: Physica Verlag.

Dubois, D. & Prade, H. 1997 The three semantics of fuzzy sets. *Fuzzy Sets and Systems* **90(2)**, 141–150.

Fargues, J. 1989 CG information retrieval using linear resolution, generalization and graph splitting. *Proceedings of the Fourth annual workshop on conceptual graphs*.

Freitag, D. & Kushmerick, N. 2000 Boosted Wrapper Induction. *AAAI/IAAI*, pp. 577–583. AAAI Press / The MIT Press.

Gagliardi, H., Haemmerlé, O., Pernelle, N. & Saïs, F. 2005 A Semantic Enrichment of Data Tables Applied to Food Risk Assessment. *Discovery Science* (ed. A. G. Hoffmann, H. Motoda & T. Scheffer), *Lecture Notes in Computer Science*, vol. 3735, pp. 374–376. Springer.

Garcia-Molina, H., Quass, D., Papakonstantinou, Y., Rajaraman, A., Sagiv, Y., Ullman, J. D. & Widom, J. 1995 The TSIMMIS Approach to Mediation: Data Models and Languages. *NGITS* (ed. A. Motro & M. Tennenholtz), pp. 0–.

Genest, D. & Salvat, E. 1998 A Platform Allowing Typed Nested Graphs: How CoGITo Became CoGITaNT (Research Note). *ICCS* (ed. M.-L. Mugnier & M. Chein), *Lecture Notes in Computer Science*, vol. 1453, pp. 154–164. Springer.

Ginsburg, S. & Hull, R. 1983 Order Dependency in the Relational Model. *Theor. Comput. Sci.* **26**, 149–195.

Guinaldo, O. & Haemmerlé, O. 1997 CoGITo : une plate-forme logicielle pour raisonner avec des graphes conceptuels. *INFORSID*, pp. 287–306.

Haemmerlé, O., Buche, P. & Thomopoulos, R. 2007 The MIEL system: uniform interrogation of structured and weakly-structured imprecise data (status on line, to appear). *Journal of Intelligent Information Systems* .

Halevy, A. Y. 2001 Answering queries using views: A survey. *VLDB J.* **10** (4), 270–294.

Hignette, G., Buche, P., Dibie-Barthélemy, J. & Haemmerlé, O. 2005 Fuzzy semantic annotation of XML documents. *Proceedings of the Conference on Advanced Information Systems Engineering Workshop DisWeb 2005*, pp. 319–332. Porto, Portugal.

Ichikawa, T. & Hirakawa, M. 1986 ARES: A Relational Database with the Capability of Performing Flexible Interpretation of Queries. *IEEE Trans. Software Eng.* **12** (5), 624–634.

Impe, J. V., Geeraerd, M., Legut'erinel, A. & P. Mafart, e. 2003 *Conference Proceedings of Predictive modelling in foods*. Katholieke Universiteit Leuven / BioTeC, Belgium.

Ireland, J. & Moller, A. 2000 Review of international food classification and description. *J. Food Comp. Anal.* **13** (4), 529–538.

Kacprzyk, J. & Ziolkowski, A. 1986 Database queries with fuzzy linguistic quantifiers. *IEEE Transactions on Systems, Man and Cybernetics SMC* **16** (3), 474–479.

Lacroix, M. & Lavency, P. 1987 Preferences; Putting More Knowledge into Queries. *VLDB* (ed. P. M. Stocker, W. Kent & P. Hammersley), pp. 217–225. Morgan Kaufmann.

Levy, A. Y., Rajaraman, A. & Ordille, J. J. 1996*a* Query-Answering Algorithms for Information Agents. *AAAI/IAAI, Vol. 1*, pp. 40–47.

Levy, A. Y., Rajaraman, A. & Ordille, J. J. 1996*b* Querying Heterogeneous Information Sources Using Source Descriptions. *VLDB* (ed. T. M. Vijayaraman, A. P. Buchmann, C. Mohan & N. L. Sarda), pp. 251–262. Morgan Kaufmann.

Levy, A. Y., Srivastava, D. & Kirk, T. 1995 Data Model and Query Evaluation in Global Information Systems. *J. Intell. Inf. Syst.* **5** (2), 121–143.

Leymann, F. & Roller, D. 2002 Using flows in Information integration. *IBM Systems Journal* **41**, 732–742.

Lin, D. 1998 An Information-Theoretic Definition of Similarity. *In proceedings of 15th International Conference on Machine Learning*, pp. 296–304.

Lipski, W. 1979 On Semantic Issues Connected with Incomplete Information Databases. *ACM Trans. Database Syst.* **4** (3), 262–296.

Lipski, W. 1981 On Databases with Incomplete Information. *J. ACM* **28** (1), 41–70.

Loiseau, Y., Boughanem, M. & Prade, H. 2005 *Soft computing for Information Retrieval on the Web*, springer-verlag edn., chap. Evaluation of term-based queries using possibilistic ontologies. E. Herrera-Viedma and G. Pasi and F. Crestani.

McMeekin, T., Olley, J., Ross, T. & Ratkowsky, D. 1993 *Predictive Microbiology : Theory and Application*. Research Studies Press, Taunton.

McMeekin, T. & Ross, T. 2002 Predictive microbiology : providing a knowledge-based framework for change management. *Int. J. Food Microbiology* **78**, 133–153.

Miyamoto, S. & Nakayama, K. 1986 Fuzzy information retrieval based on a fuzzy pseudothesaurus. *IEEE Transactions on Systems, Man and Cybernetics* **16**, 278–282.

Morton, S. 1987 Conceptual graphs and fuzziness in artificial intelligence. PhD thesis, University of Bristol.

Motro, A. 1984 Query Generalization: A Method for Interpreting Null Answers. *Expert Database Workshop*, pp. 597–616.

Motro, A. 1988 VAGUE: A User Interface to Relational Databases that Permits Vague Queries. *ACM Trans. Inf. Syst.* **6** (3), 187–214.

Mugnier, M. & Chein, M. 1996 Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle* **10** (1), 7–56.

Nauta, M. 2002 Modelling bacterial growth in quantitative microbiological risk assessment : is it possible ? *Int. J. Food Microbiology* **73**, 297–304.

Nierman, A. & Jagadish, H. V. 2002 ProTDB: Probabilistic Data in XML. *VLDB*, pp. 646–657.

Omri, M. & Chouigui, N. 2001 Measure of similarity between fuzzy concepts for identification of fuzzy user's requests in fuzzy semantic networks. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **9** (6), 743–748.

Papakonstantinou, Y., Garcia-Molina, H. & Widom, J. 1995*a* Object Exchange Across Heterogeneous Information Sources. *ICDE* (ed. P. S. Yu & A. L. P. Chen), pp. 251–260. IEEE Computer Society.

Papakonstantinou, Y., Gupta, A., Garcia-Molina, H. & Ullman, J. D. 1995*b* A Query Translation Scheme for Rapid Implementation of Wrappers. *DOOD* (ed. T. W. Ling, A. O. Mendelzon & L. Vieille), *Lecture Notes in Computer Science*, vol. 1013, pp. 161–186. Springer.

Pivk, A., Cimiano, P. & Sure, Y. 2004 From Tables to Frames. *International Semantic Web Conference* (ed. S. A. McIlraith, D. Plexousakis & F. van Harmelen), *Lecture Notes in Computer Science*, vol. 3298, pp. 166–181. Springer.

Prade, H. 1984 Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory. *Information Systems* **9** (1), 27–42.

Prade, H. & Testemale, C. 1984 Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences* **34**, 115–143.

Rabitti, F. & Savino, P. 1990 Retrieval of Multimedia Documents by Imprecise Query Specification. *EDBT* (ed. F. Bancilhon, C. Thanos & D. Tsichritzis), *Lecture Notes in Computer Science*, vol. 416, pp. 203–218. Springer.

van Rijsbergen, C. 1979 *Information retrieval*. University of Glasgow, book online at http://www.dcs.gla.ac.uk/Keith/Preface.html.

Rossazza, J., Dubois, D. & Prade, H. 1998 *Fuzzy and uncertain object-oriented databases: concepts and models*, r. de caluwe edn., *Advances in Fuzzy systems - Applications and Theory*, vol. 13, chap. A hierarchical model of fuzzy classes, pp. 21–61. World Scientific.

Roth, M. A., Wolfson, D. C., Kleewein, J. C. & Nelin, C. J. 2002 Information integration: a new generation of information technology. *IBM Systems Journal* **41**, 563–577.

Rundensteiner, E. & Bandler, W. 1986 The equivalence of knowledge representation schemata : 'semantic networks' and 'fuzzy relational products'. *Proceedings of the North-American Fuzzy Information Processing Society Conference (NAFIPSŠ86)*, pp. 477–501. New Orleans.

Schlieder, T. 2002 Schema-Driven evaluation of Approximate Tree-Pattern Queries. *Proceedings of the 8th International Conference on Extending Database Technology (EDBT 2002)*.

Sowa, J. 1984 *Conceptual structures - Information processing in Mind and Machine*. Addison-Welsey.

Tahani, V. 1977 A conceptual framework for fuzzy query processing - A step toward very intelligent database systems. *Inf. Process. Manage.* **13** (5), 289–303.

Thomopoulos, R. 2003 Représentation et interrogation élargie de données imprécises et faiblement structurées. PhD thesis, Institut National Agronomique Paris-Grignon.

Thomopoulos, R., Bosc, P., Buche, P. & Haemmerlé, O. 2003*a* Logical interpretation of fuzzy conceptual graphs. *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS'03)*, pp. 173–178. Chicago, USA.

Thomopoulos, R., Buche, P. & Haemmerlé, O. 2002 Extension du modèle des graphes conceptuels à la représentation de données floues. *Actes de la conférence RFIA (Reconnaissances des Formes et Intelligence Artificielle)*, pp. 318–328.

Thomopoulos, R., Buche, P. & Haemmerlé, O. 2003*b* Different kinds of comparisons between Fuzzy Conceptual Graphs. *Proceedings of the 11th International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence 2746*, pp. 54–68. Dresden, Germany: Springer.

Thomopoulos, R., Buche, P. & Haemmerlé, O. 2003*c* Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy sets and System* **140**, 111–128.

Thomopoulos, R., Buche, P. & Haemmerlé, O. 2004 Sous-ensembles flous définis sur une ontologie. *4èmes journées d'Extraction et de Gestion des Connaissances*, pp. 147–158.

Thomopoulos, R., Buche, P. & Haemmerlé, O. 2006 Fuzzy sets defined on a hierarchical domain. *IEEE Transactions on Knowledge and Data Engineering* **18** (10), 1397–1410.

Turowski, K. & Weng, U. 2002 Representing and processing fuzzy information - an XML-based approach. *Knowl.-Based Syst.* **15** (1-2), 67–75.

Ullman, J. D. 1988 *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press.

Ullman, J. D. 2000 Information integration using logical views. *Theor. Comput. Sci.* **239** (2), 189–210.

Vassiliadis, P. 2000 Gulliver in the land of data warehousing: practical experiences and observations of a researcher. *DMDW* (ed. M. A. Jeusfeld, H. Shu, M. Staudt & G. Vossen), *CEUR Workshop Proceedings*, vol. 28, p. 12. CEUR-WS.org.

Widom, J. 1995 Research Problems in Data Warehousing. *CIKM*, pp. 25–30. ACM.

Wiederhold, G. 1992 Mediators in the Architecture of Future Information Systems. *IEEE Computer* **25** (3), 38–49.

Wu, M.-C. & Buchmann, A. P. 1997 Research Issues in Data Warehousing. *BTW*, pp. 61–82.

Wuwongse, V. & Manzano, M. 1993 Fuzzy conceptual graphs. *Proceedings of the First International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence #699*, pp. 430–449. Quebec City, Canada: Springer-Verlag.

Wuwongse, V. & Tru, C. H. 1996 Towards Fuzzy Conceptual Graph Programs. *ICCS* (ed. P. W. Eklund, G. Ellis & G. Mann), *Lecture Notes in Computer Science*, vol. 1115, pp. 263–276. Springer.

Xyleme, L. 2001 A dynamic warehouse for XML Data of the Web. *IEEE Data Eng. Bull.* **24** (2), 40–47.

Yager, R. 1988 On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics* **18**, 183–190.

Yangarber, R., Lin, W. & Grishman, R. 2002 Unsupervised Learning of Generalized Names. *COLING*.

Zadeh, L. 1965 Fuzzy sets. *Information and Control* **8**, 338–353.

Zadeh, L. 1978 Fuzzy sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems* **1**, 3–28.

Zongmin, M. 2005 *Fuzzy database modeling with XML*. Springer.