Coopération de connaissances hétérogènes pour la construction et la validation de l'expertise d'un domaine

Rallou Thomopoulos[*,**], Jean-François Baget[***,**], Ollivier Haemmerlé[****]

[*]INRA, UMR IATE (bâtiment 31), 2 place P. Viala, 34060 Montpellier cedex 1
rallou.thomopoulos@ensam.inra.fr
[**]LIRMM (CNRS & Université Montpellier II), 161 rue Ada, 34392 Montpellier cedex 5
[***]LIG/INRIA Rhône-Alpes, 655 av. de l'Europe, Montbonnot St-Martin, 38334 St-Ismier cedex
jean-francois.baget@inrialpes.fr
[****]GRIMM-ISYCOM, Univ. Toulouse le Mirail, 5 allées Antonio Machado, 31058 Toulouse cedex
ollivier.haemmerle@univ-tlse2.fr

Résumé. Ce travail se situe dans le contexte général de la construction et de la validation de l'expertise d'un domaine. Il vise la coopération de deux types de connaissances, hétérogènes par leur niveau de granularité et par leur formalisme : des dires d'experts représentés dans le modèle des graphes conceptuels et des données expérimentales représentées dans le modèle relationnel. Nous proposons d'automatiser deux étapes : d'une part, la génération d'une ontologie simple (partie terminologique du modèle des graphes conceptuels) guidée à la fois par le schéma relationnel et par les données qu'il contient ; d'autre part, l'évaluation de la validité des dires d'experts au sein des données expérimentales. La méthode que nous introduisons pour cela est fondée sur l'utilisation de graphes conceptuels patrons annotés. Ces résultats ont été implémentés au sein d'une application concrète concernant le contrôle de la qualité alimentaire.

1 Introduction

La coopération de connaissances hétérogènes a été – et continue à être – très étudiée sous un aspect particulier : l'intégration de sources hétérogènes, coopérant pour répondre à une requête de l'utilisateur, chaque source étant en mesure de fournir une partie des réponses ou encore des réponses partielles. Elle continue à être une problématique essentielle, notamment dans le cadre de la mise en correspondance d'ontologies, du fait du nombre croissant de sources d'informations disponibles via le Web. La problématique qui nous intéresse ici est toutefois différente. En effet, alors qu'en intégration de sources hétérogènes les différentes sources d'information ont le même rôle (la mise à disposition d'information en vue de répondre à une requête), ici les différents types de connaissances n'ont pas le même statut : une des sources contient des connaissances synthétiques, d'un niveau de granularité général et considérées comme appréhendables par l'humain, elle fournit des règles génériques sans couvrir tous les cas particuliers possibles ; les autres sources, au contraire, sont d'un niveau de granularité très fin, précises et fiables, mais trop circonstanciées pour être directement exploitables par l'humain.

Dans cette étude, les formalismes utilisés pour les différentes sources sont eux aussi hétérogènes, adaptés au type de connaissance représenté :

1. des dires d'experts, connaissances à caractère générique, découlant de l'expérience des spécialistes du domaine et décrivant les mécanismes communément admis régissant ce domaine. Ces connaissances sont représentées sous la forme de règles dans le modèle des graphes conceptuels. Nous développons dans ce papier la justification du choix de ce modèle de représentation des connaissances ;
2. des données expérimentales, issues de la littérature internationale du domaine. Elles sont représentées dans le modèle relationnel. Ces données nombreuses décrivent avec précision et de façon chiffrée des expériences réalisées pour approfondir la connaissance du domaine et leurs résultats. Ces résultats peuvent – ou non – vérifier les connaissances apportées par les dires d'experts.

La coopération des deux types de connaissances permet de tester la validité des dires d'experts sur les données expérimentales, et à plus long terme de consolider l'expertise du domaine.

Deux différences importantes entre les deux formalismes – ayant des répercussions sur les vocabulaires utilisés – sont, d'une part, que les graphes conceptuels représentent des connaissances d'un caractère beaucoup plus générique que celles de la base de données relationnelle, d'autre part, que le modèle des graphes conceptuels comporte une partie ontologique (vocabulaire hiérarchisé constituant le support du modèle) contrairement au modèle relationnel. Nous proposons dans un premier temps la génération d'une ontologie, guidée par les informations de structure et les données du modèle relationnel, qui en l'occurrence préexistent aux connaissances exprimées sous forme de graphes conceptuels. Les difficultés rencontrées sont les suivantes : comment identifier, au sein du schéma relationnel et/ou des données qu'il contient, les concepts que l'on peut considérer comme pertinents pour un niveau de granularité plus général, celui des dires d'experts ? Comment hiérarchiser les différents concepts identifiés, alors que le modèle relationnel ne prend pas explicitement en compte la relation "sorte de" ? Peut-on aller plus loin dans la suggestion de concepts complémentaires pertinents ? La méthodologie proposée est semi-automatique, elle nécessite une validation experte.

Dans un deuxième temps, nous introduisons un processus permettant de tester la validité des dires d'experts au sein des données expérimentales, c'est-à-dire de réaliser l'interrogation d'une base de données relationnelle par un système dans le formalisme des graphes conceptuels. Cette étape est automatique. Outre la définition de l'évaluation de la validité des dires d'experts, le problème posé est celui de l'automatisation de la construction de requêtes SQL à partir de graphes conceptuels dont la forme et le contenu peuvent varier. Le processus que nous proposons s'appuie sur l'utilisation de graphes conceptuels patrons annotés.

Ce travail est illustré par une application concrète dans le domaine de la qualité alimentaire mené par l'INRA (Institut National de la Recherche Agronomique) de Montpellier.

Le papier est organisé de la façon suivante. La section 2 rappelle un certain nombre de notions préliminaires concernant le modèle des graphes conceptuels. La section 3 décrit la génération d'une ontologie, guidée par les informations de structure et les données du modèle relationnel. La section 4 présente la méthode d'évaluation de la validité des dires d'experts au sein des données expérimentales. La section 5 est consacrée à l'application des résultats au sein d'un projet concernant le contrôle de la qualité alimentaire. Enfin, nous concluons et présentons les perspectives de ce travail.

2 Notions préliminaires

Nous rappelons ici la syntaxe et la sémantique de deux formalismes de la famille des graphes conceptuels (Sowa, 1984) : les graphes conceptuels simples et leur extension aux règles. La formalisation adoptée ici est proche de celle de Mugnier (2000), que le lecteur pourra consulter pour plus de précision.

Le choix de ce formalisme pour modéliser les connaissances d'experts est justifié par les considérations suivantes, développées dans (Bos et al., 1997) et (Genest, 2000) :

- l'aspect graphique (diagrammatique) des connaissances représentées rend la modélisation plus simple par l'expert, et son apprentissage du langage plus rapide ;
- les raisonnements sont calculés par des opérations de graphes et sont donc, eux aussi, représentables graphiquement, ce qui permet à l'expert d'affiner sa modélisation en visualisant de façon intuitive les conséquences de celle-ci.

2.1 Les graphes conceptuels simples

Les graphes conceptuels simples forment un langage correspondant au fragment positif, conjonctif, existentiel de la logique du premier ordre. Il a été introduit (Sowa, 1976) comme une interface graphique pour les bases de données relationnelles.

Syntaxe Dans ce langage, un vocabulaire encode les connaissances du niveau ontologique (des noms de classes et leur hiérarchie), tandis que les graphes encodent des connaissances factuelles (les instances et les relations entre elles).

Définition 1 *Un* vocabulaire *est un n-uplet $\mathcal{V} = ((T_C, \leq_C), (T_1, \leq_1), \ldots, (T_k, \leq_k))$ d'ensemble finis, partiellement ordonnés et deux à deux disjoints où les éléments de T_C sont des* types de concepts *et les éléments de T_i sont des* types de relations *d'arité i. Nous nous donnons également deux ensembles disjoints M et V de* marqueurs individuels *et de* noms de variables.

Définition 2 *Un graphe (conceptuel) simple défini sur un vocabulaire $\mathcal{V}$ est un quintuplet $G = (C, R, \gamma, \tau, \mu)$ où C est un ensemble de* concepts*; R est un ensemble de* relations*, $\gamma : R \rightarrow C^+$ associe à chaque relation un tuple de concepts (ses* arguments*), dont la taille est le* degré *de la relation ; τ associe à chaque concept de C un élément de T_C et à chaque relation de degré i de R un élément de T_i (leur type) ; et μ associe à chaque concept c de C un marqueur individuel de M (c est dit individuel) ou un nom de variable de V (c est dit générique).*

Un vocabulaire est représenté par les diagrammes de Hasse de ses ordres partiels. Nous représentons un graphe simple de la façon suivante : chaque concept c est représenté par un rectangle à l'intérieur duquel est inscrit la chaîne $\tau(c) : \mu(c)$; chaque relation r est représentée par un ovale contenant la chaîne $\tau(r)$; si c est le i-ième argument de la relation r, on dessine un trait entre les représentations de c et r, et on inscrit i à côté de ce trait. Ainsi, la Fig. 1 représente le graphe simple défini par : $G = (C, R, \gamma, \epsilon_C, \epsilon_R)$ où : $C = \{c_1, c_2, c_3, c_4\}$; $R = \{r_1, r_2, r_3\}$; $\gamma(r_1) = (c_1, c_2)$, $\gamma(r_2) = (c_1, c_3)$, $\gamma(r_3) = (c_3, c_4)$; $\tau(c_1) =$ Aliment, $\tau(c_2) =$ Cuisson à l'eau, $\tau(c_3) =$ Vitamine, $\tau(c_4) =$ Teneur, $\tau(r_1) =$ subit, $\tau(r_2) =$ contient, $\tau(r_3) =$ caractérisé ; $\mu(c_1) =$ Frekeh, $\mu(c_2) = x1$, $\mu(c_3) = x2$, $\mu(c_4) = x3$ (où Frekeh est un marqueur individuel, et $x1, x2, x3$ sont des noms de variables).

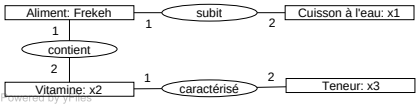

FIG. 1 – *Un graphe conceptuel simple G.*

Sémantique L'opérateur Φ associe une formule logique à un vocabulaire ou à un graphe simple. Le problème de déduction entre graphes simples peut ainsi être défini par le problème de déduction des formules logiques associées. Ces formules sont obtenues de la façon suivante :

Interprétation d'un vocabulaire Soient t et t' deux types de relations d'arité i. On note $\phi(t, t') = \forall x_1 \ldots \forall x_i (t(x_1, \ldots, x_i) \rightarrow t'(x_1, \ldots, x_i))$. L'interprétation du vocabulaire $\Phi(\mathcal{V})$ est la conjonction, pour toute paire de types telle que $t \leq t'$, des formules $\phi(t, t')$. Notons que les types de concepts sont interprétés comme des types de relations d'arité 1.

Interprétation d'un graphe A chaque concept c nous associons l'atome $\phi(c) = \tau(c)(\mu(c))$ ($\mu(c)$ est une constante si c est individuel, une variable sinon) ; et à chaque relation r t.q. $\gamma(r) = (c_1, \ldots, c_i)$ l'atome $\tau(r)(\mu(c_1), \ldots, \mu(c_i))$. Notons $\phi(G)$ la conjonction des $\phi(x)$ pour tous les concepts et relations de G. Alors $\Phi(G)$ est la fermeture existentielle de $\phi(G)$.

Par exemple, la traduction par Φ du graphe de la Fig. 1, représentant l'information "le frekeh subit une cuisson à l'eau et contient une vitamine caractérisée par une certaine teneur", est : $\exists x1\ \exists x2\ \exists x3$ (Aliment(Frekeh) $\wedge$ Cuisson à l'eau($x1$) $\wedge$ Vitamine($x2$) $\wedge$ Teneur($x3$) $\wedge$ subit(Frekeh, $x1$) $\wedge$ contient(Frekeh, $x2$) $\wedge$ caractérisé($x2$, $x3$)).

Le problème d'inférence dans les graphes conceptuels simples consiste à savoir si on peut déduire un graphe Q (répondre à la requête Q) à partir d'une base de connaissances constituée d'un graphe G, ou d'un ensemble de graphes, (la base de faits) et d'un vocabulaire. Il s'agit d'un problème NP-difficile.

Définition 3 *Soient G et Q deux graphes simples définis sur un vocabulaire $\mathcal{V}$. On dit que Q est conséquence de G et on note $G \models Q$ ssi $\Phi(\mathcal{V}), \Phi(G) \models \Phi(Q)$.*

Inférences Le calcul de conséquence entre graphes simples est efficacement réalisé par une sorte d'homomorphisme de graphes étiquetés appelé projection.

Définition 4 *Soient $G = (C_G, R_G, \gamma_G, \tau_G, \mu_G))$ et $Q = (C_Q, R_Q, \gamma_Q, \tau_Q, \mu_Q)$ deux graphes simples définis sur un vocabulaire$\mathcal{V}$. Une* projection *de H dans G est une application π de C_Q dans C_G telle que :*

- $\forall c, c' \in C_Q, \mu_Q(c) = \mu_Q(c') \Rightarrow \pi(c) = \pi(c')$;
- $\forall c \in C_Q$, *c est individuel* $\Rightarrow \mu_G(\pi(c)) = \mu_Q(c)$;
- $\forall c \in C_Q, \tau_G(\pi(c)) \leq_C \tau_Q(c)$;
- $\forall r \in R_Q$, *avec* $\gamma(r) = (c_1, \ldots, c_k)$, $\exists r' \in R_G$ *tq* $\gamma(r') = (\pi(c_1), \ldots, \pi(c_k))$ *et* $\tau(r') \leq_k \tau(r)$.

Théorème 1 *Soient G et Q deux graphes simples définis sur un vocabulaire $\mathcal{V}$, où G est sous forme normale*[1]*. Alors il existe une projection de Q dans G si et seulement si $G \models Q$.*

[1] Un graphe simple est sous forme normale quand tous ses concepts ont un marqueur distinct. Tout graphe peut être transformé en un graphe normal équivalent en temps linéaire.

2.2 Règles de graphes conceptuels

Syntaxe Les règles (Salvat, 1998) forment une extension des graphes conceptuels dans laquelle on ajoute à une base de connaissance des règles de la forme "si A alors B" où A et B sont deux graphes simples. L'ajout de règles augmente fortement l'expressivité du langage, puisque l'on obtient un problème de déduction semi-décidable (c'est un modèle de calcul).

Définition 5 *Une règle (de graphe conceptuel simple) définie sur un vocabulaire $\mathcal{V}$ est une paire $R = (H, C)$ de graphes simples définis sur $\mathcal{V}$. H est l'hypothèse de R et C sa conclusion.*

Une règle est représentée graphiquement par les deux graphes qui la composent, séparés par un symbole d'implication allant de l'hypothèse vers la conclusion, comme dans la Fig. 2 qui représente la règle : "si un aliment subit une cuisson à l'eau et contient une vitamine caractérisée par une certaine teneur, alors cette teneur montre une diminution".

FIG. 2 – *Une règle de graphe conceptuel simple R.*

Sémantique L'opérateur Φ est étendu afin de traduire les règles. Si $R = (H, C)$ est une règle, alors $\Phi(R) = \forall x_1 \ldots \forall x_i(\phi(H) \rightarrow (\exists y_1 \ldots \exists y_j \phi(C)))$, où les x_p sont les noms de variables de H et les y_q sont les noms de variables de C qui ne sont pas dans H.

La formule logique associée à la règle R de la Fig. 2 est $\Phi(R) = \forall x1\ \forall x2\ \forall x3\ \forall x4$ ((Aliment(x1) $\wedge$ Cuisson à l'eau($x2$) $\wedge$ Vitamine($x3$) $\wedge$ Teneur($x4$) $\wedge$ subit(x1, $x2$) $\wedge$ contient(x1, $x3$) $\wedge$ caractérisé($x3$, $x4$)) $\rightarrow$ ($\exists$ $x5$ (Teneur($x4$) $\wedge$ Diminution($x5$) $\wedge$ montre($x4$, $x5$)))).

Définition 6 *Soient G et Q deux graphes simples définis sur un vocabulaire $\mathcal{V}$, et $\mathcal{R} = \{R_1, \ldots, R_k\}$ un ensemble de règles définies sur $\mathcal{V}$. On dit que Q est conséquence de G et $\mathcal{R}$ et on note $G, \mathcal{R} \models Q$ ssi $\Phi(\mathcal{V}), \Phi(G), \Phi(R_1), \ldots, \Phi(R_k) \models \Phi(Q)$.*

Inférences Le calcul de déduction en présence de règles peut se faire en marche avant ou en marche arrière (voir (Baget et Salvat, 2006) pour une présentation de ces deux méthodes). Nous présentons ici brièvement la marche avant, plus intuitive.

Définition 7 *Une règle $R = (H, C)$ est dite applicable à un graphe simple G s'il existe une projection π de H dans G. Dans ce cas, appliquer R à G suivant π consiste à faire l'union disjointe [2] de G et de sp(π, C), où sp(π, C) est obtenu en remplaçant le marqueur de tout concept c de C identique à celui d'un sommet c' de H par le marqueur de $\pi(c')$;puis à mettre sous forme normale le graphe obtenu.*

Théorème 2 *Soient G et Q deux graphes simples définis sur un vocabulaire $\mathcal{V}$, et $\mathcal{R}$ un ensemble de règles définies sur $\mathcal{V}$. Alors il existe une séquence finie d'applications de règles de $\mathcal{R}$ qui transforme G en un graphe simple G' tq $G' \models Q$ ssi $G, \mathcal{R} \models Q$.*

[2] L'union disjointe de deux graphes est le graphe dont le dessin est la juxtaposition de leur dessin.

3 Génération d'une ontologie

Nous nous situons dans le cas où un recueil de données expérimentales détaillées représentées dans le modèle relationnel préexiste à l'expression de connaissances expertes d'un niveau de granularité plus général. L'objectif est d'automatiser autant que possible la génération d'une ontologie simple, constituant l'ensemble des types de concepts de la partie terminologique du modèle des graphes conceptuels, à l'aide du schéma et des données relationnels existants.

Dans cette partie, après une présentation de travaux proches, nous décrivons trois étapes de la génération de l'ontologie : l'identification de types de concepts de haut niveau, la hiérarchisation de ces types de concepts, la proposition de types de concepts complémentaires.

3.1 Travaux proches

Cette problématique nécessitant une expertise importante, une méthode totalement automatisée pour la génération d'une ontologie (Pernelle et al., 2001) est exclue. Notre objectif est différent de l'apprentissage de concepts telle qu'abordée par les approches FCA – Formal Concept Analysis (Tilley et al., 2005) – ou ILP – Inductive Logic Programming (Muggleton et Raedt, 1994) –, qui s'appuient sur l'existence de propriétés communes à des sous-ensembles de données pour les regrouper en de nouveaux concepts. Ici l'objectif premier est d'identifier et de hiérarchiser des concepts pertinents pour l'expression de connaissances expertes, parmi ceux déjà présents dans les données de façon peu explicite et avec une structure inappropriée.

La recherche d'une structure hiérarchique, en particulier d'une structure arborescente, dans des données semi-structurées (Termier et al., 2002) ou non structurées (Kietz et al., 2000; Folch et al., 2004) a été étudiée, notamment dans le cadre relativement récent de l'échange et de l'interrogation de données sur le Web. En revanche, la recherche d'une nouvelle structure pour des objectifs spécifiques dans des données déjà structurées, qui est le but visé ici, est peu courante. Des travaux proches sont ceux qui touchent la question de la cohabitation entre vocabulaires hétérogènes, tels que la transformation de modèles (Sendall et Kozaczynski, 2003) et l'alignement d'ontologies (Euzenat et al., 2004). En alignement d'ontologies, des correspondances sont établies entre des vocabulaires préexistants, conçus indépendamment les uns des autres, tandis que dans cette étude l'ontologie est dérivée des données.

Des graphes conceptuels aux bases de données La correspondance entre graphes conceptuels simples et requêtes conjonctives en bases de données est bien connue (Kolaitis et Vardi, 1998; Mugnier, 2000). Soit $\mathcal{V}$ un vocabulaire, et G et Q deux graphes simples sur $\mathcal{V}$. G et Q sont transformés (en G' et Q') de la façon suivante : les types de concepts sont transformés en relations unaires, et chaque concept de type t devient un concept sans type, incident à une relation unaire typée t. Pour chaque relation r de type t, pour chaque supertype t' de t, nous rajoutons une nouvelle relation r' de type t' telle que $\gamma(r) = \gamma(r')$. Il s'ensuit que $G \models_{\mathcal{V}} Q$ ssi $\Phi(G') \models \Phi(Q')$ (nous n'avons plus besoin des formules traduisant le support, toutes leurs conséquences ont été traduites dans les graphes). Puisque $\Phi(G')$ et $\Phi(Q')$ sont des formules positives conjonctives, nous pouvons définir $\mathcal{B}$ comme les tables ayant comme formule logique associée $\Phi(G')$ et A comme la requête ayant comme formule logique associée $\Phi(Q')$. Nous avons ainsi $G \models_{\mathcal{V}} Q$ ssi il existe une réponse à A dans $\mathcal{B}$. Cependant, cette correspondance repose sur une identification entre le vocabulaire des graphes conceptuels et le schéma de bases de données, hypothèse trop forte comme nous le verrons par la suite.

Le système Sym'Previus Haemmerlé et Carbonneill (1996) proposent d'ajouter une couche "graphes conceptuels" à une base de données relationnelle (BDR) préexistante. Cette couche sert d'interface en vue de permettre une complétion des requêtes se fondant sur la sémantique des attributs présents dans la BDR. Chaque attribut de la BDR est intégré à l'ensemble des types de concepts (sous des types de concepts génériques qui spécifient le type de la donnée au sens SQL du terme). D'autres types de concepts sont ajoutés manuellement à cet ensemble afin de disposer de connaissances supplémentaires qui sont exploitées par spécialisation ou généralisation au moment de l'expression des requêtes dans le modèle des graphes conceptuels.

Le système Sym'Previus a été développé dans le cadre d'un projet de recherche français sur un outil de prévention du risque microbiologique dans les aliments (Haemmerlé et al., 2006). Cet outil repose sur trois bases distinctes, ajoutées au système successivement au fur et à mesure du développement du projet : une base de données relationnelle, une base de graphes conceptuels et une base de données XML. Les trois bases sont interrogées simultanément et uniformément par le biais d'une interface unique, qui se fonde sur une même ontologie.

Cette ontologie a été construite manuellement, au moment de l'ajout de la base de graphes conceptuels au système. Un schéma de base de données relationnelle ainsi que ses données préexistaient. L'ensemble des attributs correspondant à des entités significatives de l'application a été partitionné en deux : les attributs dont les valeurs pouvaient être hiérarchisées selon la relation "sorte de" (substrat, germe pathogène...) et les attributs dont les valeurs étaient des ensembles intrinsèquement " plats " (les noms d'auteurs de publications, par exemple). Tous les noms d'attributs significatifs ont été ajoutés à l'ontologie Sym'Previus en tant que types de concepts. Les valeurs apparaissant dans les colonnes correspondant à des attributs à valeurs hiérarchisées ont été insérés en tant que sous-types de concepts dans l'ontologie. Leur positionnement précis dans la hiérarchie a été réalisé manuellement par les experts.

3.2 Identification de types de concepts de haut niveau

Dans cette étape, l'objectif est d'identifier des types de concepts de haut niveau (niveau de granularité général). Nous identifions deux types d'entités, que nous considérons comme susceptibles de correspondre à des types de concepts de haut niveau pertinents :

- celles dont les occurrences portent un nom, c'est-à-dire qui ont un attribut "nom" (ou encore "intitulé", "libellé", contenant la chaîne "nom", etc.). Nous supposons en effet que ces entités sont de caractère plus général, par opposition aux entités secondaires correspondant à des informations plus détaillées, dont les occurrences ne sont pas nommées mais identifiées uniquement par des identifiants numériques. Ce sont les premières qui sont utiles pour l'expression des dires d'experts : ceux-ci manipulent des notions désignées par un nom et non des informations circonstantielles détaillées ;
- celles qui peuvent être subdivisées en sous-catégories. Nous cherchons pour cela les entités qui ont un attribut "catégorie" (ou encore "famille", "type", etc.). Nous supposons en effet que ces entités, du fait de la classification engendrée par leurs sous-catégories, fournissent des types de concepts pertinents pour l'ontologie.

La frontière entre les deux cas n'est pas absolue et est très dépendante du type de modélisation. Par exemple, l'attribut "nom" d'une entité peut parfaitement avoir pour valeurs des sous-catégories de l'entité considérée. Ainsi dans le cas de notre application, l'entité *Constituant nutritionnel* a un attribut "nom" destiné à prendre des valeurs telles que "Vitamine", "Lipide", etc., qui ne désignent pas à proprement parler des instances, mais des familles de constituants

nutritionnels. Si l'on prend un exemple très courant sortant du cadre de notre application, une entité *Personne* ayant un attribut "nom" peut cacher des utilisations différentes : le plus souvent, le nom d'une personne ("Dupont" par exemple) désigne un individu particulier (même si elle n'en est pas l'identifiant) ; mais si l'on se situe dans le contexte d'une application en généalogie, "Dupont" peut désigner une branche d'individus.

Du fait de cette proximité, les deux cas seront traités de façon homogène par la suite. Par souci de simplification, nous n'indiquerons pas systématiquement la liste complète des attributs considérés ("nom", "catégorie", "famille", etc.) mais nous les désignerons sous le terme d'*attributs indicateurs*. Ces attributs sont de type chaîne de caractères.

Définition 8 *On appelle* attribut indicateur *tout attribut dont le nom figure dans une liste prédéfinie de termes déclarés propres à exprimer la dénomination ou la classification. Un tel attribut est considéré comme appartenant à une entité d'un niveau de granularité général.*

Remarque 1 *Ce processus permet de proposer des types de concepts de haut niveau pertinents. Etant donnée la variabilité des modélisations, il nécessite une vérification experte.*

Utilisation du schéma relationnel Dans un premier temps, nous nous appuyons sur le schéma de la base de données relationnelle. D'un point de vue ingénierie des bases de données, après une modélisation à l'aide par exemple du modèle entité-association, on sait qu'une relation (ou table) du schéma de la base de données relationnelle correspond :

- soit à une entité du domaine représenté – elle en comporte alors les attributs. Elle peut également comporter les identifiants d'autres entités (avec lesquelles elle était liée par une association), plus rarement des attributs d'association ;
- soit à une association (de type plusieurs à plusieurs) entre entités – elle comporte alors comme attributs leurs identifiants et les attributs d'association.

La table obtenue porte généralement le nom de l'entité ou de l'association correspondante.

Afin d'identifier les types de concepts de haut niveau, nous faisons les hypothèses simplificatrices suivantes :

1. les entités – plutôt que les relations – véhiculent les principaux concepts du domaine représenté. Les types de concepts de haut niveau sont donc à rechercher dans les noms d'entités, autrement dit parmi les noms de tables du schéma relationnel ;
2. le cas d'une association ayant un attribut indicateur est considéré comme exceptionnel.

Remarque 2 *Ce dernier cas peut introduire une ambiguïté dans les trois situations suivantes :*

- *une association de type plusieurs à plusieurs ayant un attribut indicateur produit dans le schéma relationnel une table ayant ce même attribut. Le nom de cette table est alors considéré comme un type de concept de haut niveau, alors qu'il ne s'agit pas d'un nom d'entité mais d'un nom d'association. Ce cas n'est pas réellement problématique car le type de concept identifié peut être pertinent ; mais il introduit une ambiguïté sur l'origine des types de concepts identifiés ;*
- *une table représentant une entité peut avoir un attribut indicateur sans que celui-ci soit un attribut de l'entité, mais d'une association (de type un à plusieurs) liant cette entité à une autre. Dans ce cas, le nom de l'entité est abusivement considéré comme un type de concept de haut niveau, alors que c'est éventuellement le nom de l'association qui pourrait être un candidat pertinent ;*

– *une table représentant une entité peut avoir plusieurs attributs indicateurs, certains provenant d'associations liant cette entité à d'autres. Dans ce cas, le nom de l'entité est correctement identifié comme un type de concept de haut niveau, mais les noms des associations, qui pourraient être des candidats pertinents, ne sont pas considérés.*

Ces cas, s'ils survenaient, devraient être levés par l'expert lors de la vérification des types de concepts de haut niveau identifiés. Notons que ces ambiguïtés disparaissent s'il est possible de s'appuyer directement sur le modèle conceptuel des données.

Remarque 3 *Les clés étrangères ne sont pas examinées lors de la recherche des attributs indicateurs, afin d'éviter de prendre en compte plusieurs fois la même information.*

Définition 9 *Sont considérés comme* types de concepts de haut niveau issus du schéma relationnel *les noms des tables qui comportent (au moins) un attribut indicateur. Les types des concepts de haut niveau ainsi identifiés sont ajoutés à l'ontologie.*

Exemple 1 *Dans le cas de notre application, des exemples de types de concepts de haut niveau issus du schéma sont les suivants :* Aliment, Changement, Constituant, Méthode, Opération, Propriété, Variable, ... *En revanche,* Expérience, Valeur par défaut, Valeur expérimentale*, par exemple, n'ont pas été considérés comme des types de concepts de haut niveau.*

Utilisation des données relationnelles Dans un second temps, nous nous intéressons aux valeurs prises par les attributs indicateurs. Nous avons fait l'hypothèse que les attributs indicateurs sont susceptibles de prendre pour valeurs des sous-catégories de l'entité à laquelle ils appartiennent. La prise en compte des données relationnelles permet par conséquent de proposer comme types de concepts de haut niveau les valeurs des attributs indicateurs. Leur organisation hiérarchique est précisée dans la partie 3.3.

Définition 10 *Sont considérés comme* types de concepts de haut niveau issus des données *les valeurs prises par les attributs indicateurs de la base de données. Les types des concepts de haut niveau ainsi identifiés sont ajoutés à l'ontologie.*

Exemple 2 *Dans le cas de notre application, ont par exemple été définis comme types de concepts de haut niveau issus des données les types de concepts suivants :* Augmentation, Diminution, Protéine, Lipide, Vitamine, Vitamine B, Qualité, Teneur, ...

3.3 Hiérarchisation des types de concepts

Deux niveaux de hiérarchisation sont proposés :

– la hiérarchisation des types de concepts de haut niveau issus des données par rapport à ceux issus du schéma : la valeur prise par un attribut indicateur d'une table (type de concept de haut niveau issu des données) est considérée comme une spécialisation du type de concept portant le nom de cette table (type de concept de haut niveau issu du schéma). Par exemple, *Vitamine* est une spécialisation de *Constituant nutritionnel* ;
– la hiérarchisation des types de concepts de haut niveau issus des données entre eux : elle s'appuie sur l'inclusion des labels des types de concepts. Par exemple, *Vitamine B* (désignant la famille des vitamines B) est une spécialisation de *Vitamine*.

La définition 11 résume les étapes 3.2 et 3.3, leur résultat est soumis à vérification experte.

Définition 11 *La* génération d'une ontologie simple *O à partir de la base de données relationnelle est réalisée de la façon suivante. Pour chaque table, de nom noté* T*, de la base de données, si la table* T *comporte au moins un attribut indicateur, alors :*

- *le type de concept (de haut niveau issu du schéma relationnel)* T *est ajouté à* O *;*
- *pour chaque attribut indicateur de* T*, prenant un ensemble de valeurs* $v_1, \ldots, v_n$ *:*
  - *le type de concept (de haut niveau issu des données)* v_i*, sous-type de* T*, est ajouté ;*
  - *si* v_i *est inclus dans* v_j *(*$i, j \in [1, n]$*), alors* v_j *est un sous-type de* v_i*.*

Exemple 3 *Par exemple dans le cas de notre application la table* Constituant *comporte l'attribut indicateur* nom_constituant*, prenant pour valeurs* Protéine*,* Lipide*,* Vitamine*, etc.*

Le type de concept (de haut niveau issu du schéma relationnel) Constituant *et les types de concepts (de haut niveau issus des données)* Protéine*,* Lipide*,* Vitamine*,* Vitamine B *sont ajoutés à* O *comme sous-types de* Constituant*. "Vitamine" étant inclus dans "Vitamine B", le type de concept* Vitamine B *est sous-type de* Vitamine *(voir Fig. 3).*

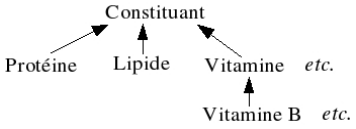

FIG. 3 – *Exemple de hiérarchisation des types de concepts*

3.4 Proposition de types de concepts complémentaires

La méthode proposée dans cette partie afin de compléter l'ontologie par la suggestion de types de concepts supplémentaires pertinents, est spécifique à la forme des connaissances expertes considérée dans l'application. Nous nous situons dans le cas suivant. Les connaissances expertes sont exprimées par des règles de la forme "si (hypothèse) alors (conclusion)". Plus précisément, il s'agit de règles de causalité exprimant une relation de cause à effet entre (i) un ensemble de conditions, décrit par l'hypothèse, interagissant entre elles pour produire (ii) l'effet qui en résulte, décrit par la conclusion.

Par exemple, une règle experte simple issue de l'application est la suivante : "si un aliment, caractérisé par une teneur en vitamines, subit une cuisson à l'eau, alors cette teneur diminue". Elle est représenté par la règle de graphes conceptuels de la figure 2.

La nature des interactions existant entre les concepts apparaissant dans l'hypothèse n'est pas toujours bien connue des experts. En particulier, ces interactions peuvent être dues à l'interférence d'autres concepts qui ne sont pas nécessairement identifiés et explicités. L'objectif de cette partie est de mettre en évidence certains de ces concepts. La méthode proposée est fondée sur la comparaison de descriptions textuelles des concepts apparaissant dans l'hypothèse.

En effet, les tables de la base de données relationnelle qui ont permis d'obtenir les types de concepts apparaissant dans l'hypothèse (cf. definition 11) fournissent parfois des descriptions textuelles, contenues dans la valeur d'un attribut nommé par exemple "description", "commentaires", etc. Pour chaque paire de types de concepts apparaissant dans une même hypothèse de règle experte et pour lesquelles une telle description est disponible, la démarche proposée consiste à rechercher dans ces descriptions l'existence de termes communs.

Exemple 4 *La comparaison des decriptions textuelles de certaines opérations (Cuisson à l'eau, Cuisson vapeur, Hydratation, Séchage) avec les descriptions textuelles de certains constituants (Son de blé, Fibre, Lipide, Vitamine, Polyphénol) ont en commun le terme "eau". En effet, ces opérations unitaires ont toutes un effet sur la teneur en eau (apport ou retrait d'eau) et ces constituants possèdent tous des sous-catégories ayant une affinité particulière avec l'eau (solubilité ou absorption particulières). La mise en évidence du terme commun "eau" a conduit les experts à compléter, d'une part, par l'ajout du type de concept* Eau*, d'autre part, par la spécialisation de types de concepts existants pour faire apparaître des catégories ayant une interaction particulière avec l'eau : ainsi* Vitamine *est spécialisé en* Vitamine hydrosoluble *(surtype, entre autres, de* Vitamine B*, qui est soluble dans l'eau) et* Vitamine liposoluble*.*

Les résultats obtenus sont nombreux et doivent être triés manuellement par l'expert.

La recherche de termes communs fait appel à des techniques de traitement de la langue naturelle, en particulier la suppression des mots creux ("stopwords"), l'homogénéisation des variations syntaxiques (tokenisation, lemmatisation).

4 Evaluation de la validité des dires d'experts

Contrairement à la partie précédente (section 3) qui nécessite une intervention experte, la méthode présentée dans cette partie est automatique. L'objectif est de tester si les connaissances expertes exprimées sous forme de règles de graphes conceptuels sont valides au sein des données expérimentales de la base relationnelle. Un taux de validité de la règle testée est calculé et les données faisant exception à la règle sont identifiées et visualisées par l'utilisateur.

Dans cette partie, après une présentation des travaux existants, nous définissons ce que nous entendons par l'évaluation de la validité d'une règle, introduisons les notions de patron et d'instance de règle, enfin exposons le déroulement de la validation d'une instance de règle.

4.1 Problématiques proches

On peut distinguer deux formes de cohabitation entre une base de données relationnelle et une base de connaissances dans le modèle des graphes conceptuels :

- il n'y a pas d'échange de données entre les deux modèles, en revanche ceux-ci sont exploités en utilisant un formalisme commun (pivot) pour l'expression des requêtes et/ou de l'ontologie du domaine. Le projet Sym'Previus (Haemmerlé et al., 2006) est un exemple d'application où le formalisme pivot est un langage de requêtes inspiré du formalisme relationnel. Le cas inverse (interrogation d'une BD par des requêtes graphes conceptuels) est celui qui nous intéresse ici ;
- il y a échange de données entre les deux modèles. Ce cas se rencontre par exemple : (i) s'il y a nécessité de migration de données vers l'un des deux formalismes jouant le rôle d'entrepôt. Ce cas a été envisagé – mais pas exploré – comme perspective au projet Sym'Previus, où le modèle des graphes conceptuels est utilisé comme formalisme de stockage provisoire et souple de données non prévues par le schéma relationnel ; (ii) si l'un des deux formalismes paraît plus adapté pour la résolution de certains types de problèmes, et que l'on fait le choix d'utiliser le formalisme le plus adapté à les traiter. Ce cas n'a pas fait l'objet de travaux à notre connaissance.

4.2 Calcul du taux de validité

Evaluer la validité d'une règle experte au sein des données expérimentales consiste à calculer la proportion de données satisfaisant à la fois l'hypothèse et la conclusion de cette règle, parmi celles qui en satisfont l'hypothèse. Si l'on note n_H le nombre de données satisfaisant l'hypothèse et $n_{H\wedge C}$ le nombre de données satisfaisant à la fois l'hypothèse et la conclusion, le taux de validité V d'une règle est $V = \frac{n_{H\wedge C}}{n_H} \times 100$, où n_H et $n_{H\wedge C}$ sont le résultat de requêtes SQL effectuant un comptage (*select count*) des données remplissant respectivement les critères de satisfaction de l'hypothèse et les critères de satisfaction de l'hypothèse et de la conclusion. Le problème qui se pose est celui de l'automatisation de la construction de ces requêtes.

4.3 Notions de patron de règle, d'instance de règle et propriétés associées

Bien que les règles expertes puissent prendre des formes variables, il est possible de les regrouper en ensembles de règles qui suivent la même forme générale.

Exemple 5 *Les règles expertes représentées par les figures 2 et 4 sont de la même forme.*

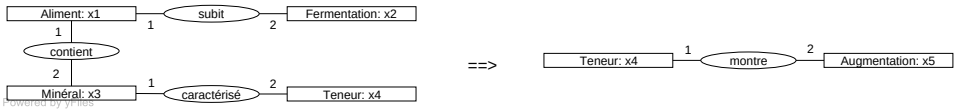

FIG. 4 – *Exemple de règle experte de même forme que celle de la figure 2*

La "forme générale" d'un ensemble de règles expertes peut elle-même être représentée par une règle, appelée patron de règle. Sa structure est identique à celle des règles expertes de cet ensemble, mais ses sommets concepts sont plus généraux que ceux des règles expertes de l'ensemble. Autrement dit, chacune des règles expertes de l'ensemble a un graphe hypothèse et un graphe conclusion qui sont des spécialisations (par restriction des étiquettes) de ceux du patron de règle. Ces règles sont appelées instances de règle. Les graphes hypothèse et conclusion du patron de règle se projettent donc dans ceux de chacune de ses instances.

Exemple 6 *Les règles des figures 2 et 4 sont des instances du patron de règle de la figure 5.*

FIG. 5 – *Exemple de patron de règle*

Le niveau de généralité des types de concepts utilisés dans un patron de règle n'est pas quelconque : il s'agit de concepts de haut niveau issus du schéma relationnel. Au contraire, les types de concepts utilisés dans une instance de règle peuvent être des concepts de haut niveau issus des données (les marqueurs peuvent de plus être individuels). Cette particularité est essentielle pour le déroulement de la validation d'une instance de règle.

Définition 12 *Un* patron de règle *est une règle, dans le formalisme des graphes conceptuels, dont les concepts ont pour types des types de concepts de haut niveau issus du schéma relationnel et dont les marqueurs sont génériques. Une* instance de règle *est une règle, dans le formalisme des graphes conceptuels, obtenue par restriction des étiquettes des sommets concepts d'un patron de règle donné. L'instance de règle est dite* conforme *à ce patron.*

En conséquence, les types de concepts apparaissant dans un patron de règle fournissent une liste de noms de tables de la base de données (les concepts de haut niveau issus du schéma). L'hypothèse (respectivement, la conclusion) d'un patron de règle peut être interprétée, au sein de la base de données, comme la formule d'une requête permettant de sélectionner les données satisfaisant l'hypothèse (respectivement, la conclusion). Cette formule fait intervenir les tables apparaissant comme types de concepts dans l'hypothèse (respectivement, la conclusion) du patron de règle. Cette formule ne fait que spécifier un schéma de requête. Elle n'est pas contrainte pas des critères de sélection particuliers. De tels critères n'apparaîtront que lors du traitement des instances de règles, présenté en 4.4.

Définition 13 *Soit H l'hypothèse d'un patron de règle. Soit Q une requête sur la base de données relationnelle permettant de sélectionner les données satisfaisant H. Q s'écrit en termes de calcul relationnel sous la forme $\{T|F(T)\}$, où F est une formule, T une variable n-uplet de F et $F(T)$ une évaluation de F. La réponse à la requête Q sera un ensemble de n-uplets $\{t|F(t)vraie\}$. F est construite par la conjonction des formules suivantes.*

- Formules atomiques associées aux concepts de H : *Soit $s_{c_1}, \ldots, s_{c_n}$ les concepts de H, de types $c_1, \ldots, c_n$ (ce sont des types de concepts de haut niveau issus du schéma relationnel et donc des tables de la base de données relationnelle). Les concepts de H étant génériques, chaque concept s_{c_i} fournit la formule atomique : $\exists x_i, c_i(x_i)$.*
- Formules associées aux relations de H : *Soit s_r un sommet relation de H avec $\gamma(s_r) = (s_{c_k}, \ldots, s_{c_l})$. Deux cas de figure peuvent se présenter :*
  - *le schéma de Q ne fait pas intervenir d'autres tables que celles présentes dans H pour joindre les tables $c_k, \ldots, c_l$. Chaque concept $s_{c_k}, \ldots, s_{c_l}$ de $\gamma(s_r)$ fournit au moins une formule atomique*[3] *de la forme : $x_i.a_i = X_i$, où a_i désigne un attribut de la table c_i et X_i une constante ou une expression $x_j.a_j$ ($j \in [k, l]$, a_j attribut de c_j).*
  - *le schéma de la requête Q fait intervenir d'autres tables que celles présentes dans H pour joindre les tables $c_k, \ldots, c_l$. Soit $t_m, \ldots, t_p$ ces tables. Chacune d'entre elles fournit une formule atomique $\exists x_i, t_i(x_i)$ et au moins une formule atomique $x_i.a_i = X_i$. Le sommet relation s_r fournit alors une formule (non-atomique) de la forme : $\exists x_m, \ldots, x_p, t_m(x_m) \wedge \ldots \wedge t_p(x_p) \wedge x_k.a_k = X_k \wedge \ldots \wedge x_l.a_l = X_l \wedge x_m.a_m = X_m \wedge \ldots x_p.a_p = X_p$.*
- Attributs recherchés : *Soit $attr_1, \ldots, attr_q$ les attributs recherchés, issus respectivement des tables $tbl_1, \ldots, tbl_q$ ($attr_i$ non nécessairement distinct de a_j, $j \in [k, l] \cup [m, p]$ et tbl_i dans $\{c_1, \ldots, c_n, t_m, \ldots, t_p\}$). $F(t)$ est contrainte par : $t.attr_i = tbl_i.attr_i$ ($i \in [1, q]$).*

Dans le cas général, $F(t)$ est donc de la forme : $\exists x_1, \ldots, x_n, x_m, \ldots, x_p, c_1(x_1) \wedge \ldots \wedge c_n(x_n) \wedge t_m(x_m) \wedge \ldots \wedge t_p(x_p) \wedge x_k.a_k = X_k \wedge \ldots \wedge x_l.a_l = X_l \wedge x_m.a_m = X_m \wedge \ldots x_p.a_p = X_p \wedge t.attr_1 = tbl_1.attr_1 \ldots t.attr_z = tbl_q.attr_q$.

[3] Ces formules atomiques ne sont pas nécessairement distinctes de celles fournies par les autres voisins de s_r, par exemple un voisin peut fournir $x_i.a_i = x_j.a_j$ et un autre $x_j.a_j = x_i.a_i$.

A ce stade (patrons de règle), cette formule ne peut être qu'en partie générée de façon automatique. En effet, les tables $t_m, \ldots, t_p$, les attributs a_i et les termes X_i ne peuvent pas toujours être calculés. Les limites de l'automatisation sont dues à l'ambiguïté des jointures entre tables et du fait des possibilités multiples que l'on peut rencontrer en cas de jointures intermédiaires à réaliser. La formule F doit donc être définie par le concepteur pour l'hypothèse de chaque patron de règle.

La formule de la requête permettant de sélectionner les données satisfaisant la conclusion d'un patron de règle est construite de la même façon. Enfin, la formule de la requête permettant de sélectionner les données satisfaisant à la fois l'hypothèse et la conclusion d'un patron de règle est obtenue par conjonction des formules associées à l'hypothèse et à la conclusion.

Pour permettre l'évaluation d'une règle experte (voir partie 4.2), les deux requêtes nécessaires sont celle comptant les données satisfaisant l'hypothèse et celle comptant les données satisfaisant à la fois l'hypothèse et la conclusion d'un patron de règle. Ces deux requêtes sont associées par le concepteur à chaque patron de règle.

Exemple 7 *La formule associée à l'hypothèse du patron de règle de la figure 5 est*
$\exists x_1, x_2, x_3, x_4, x_5, x_6, aliment(x_1) \wedge operation(x_2) \wedge constituant(x_3) \wedge propriete(x_4) \wedge resultat(x_5) \wedge etude(x_6) \wedge x_1.id\_aliment = x_5.id\_aliment \wedge x_2.id\_operation = x_6.id\_operation \wedge x_3.id\_constituant = x_5.id\_sous\_constituant \wedge x_4.id\_propriete = x_5.id\_propriete \wedge x6.id\_etude = x5.id\_etude \wedge t.x_4.id\_resultat = x_5.id\_resultat$

La requête SQL associée à l'hypothèse du patron de la figure 5 est

```
SELECT COUNT(resultat.id_resultat)
FROM resultat, aliment, constituant, etude, operation
WHERE resultat.id_aliment = aliment.id_aliment
AND etude.id_operation = operation.id_ operation
AND resultat.id_sous_constituant = constituant.id_constituant
AND resultat.id_ propriete = propriete.id_propriete
AND resultat.id_etude = etude.id_etude
```

A chaque concept d'un patron de règle est associé une information, destinée à indiquer la spécialisation de ce sommet concept au sein des instances de règle conformes à ce patron :

- si le type de concept de ce sommet a des sous-types (concepts de haut niveau issus des données), de quel attribut de la table sont-ils des valeurs ? Cet attribut est supposé le même pour toutes les instances d'un patron de règle donné ;
- si le marqueur de ce sommet est susceptible d'être individuel au sein des instances de règle, de quel attribut de la table sont-ils des valeurs ? On suppose l'existence d'un tel attribut et, là encore, cet attribut est supposé le même pour toutes les instances.

Exemple 8 *Dans le patron de règle de la figure 5, le type* Constituant *peut être spécialisé par des sous-types qui sont aussi des valeurs de l'attribut* nom_constituant *de la table* Constituant*.*

Ainsi dans les figures 2 et 4, Vitamine *et* Minéral*, qui sont des spécialisations du type de concept* Constituant*, sont aussi des valeurs de l'attribut* Constituant.nom_constituant*.*

Définition 14 *Un patron* annoté *est un patron de règle P auquel sont associés :*

- *une requête d'hypothèse, dénombrant le nombre de n-uplets de la base de données satisfaisant l'hypothèse de P ;*

- *une requête d'hypothèse et conclusion, dénombrant le nombre de n-uplets de la base de données satisfaisant à la fois l'hypothèse et la conclusion de P ;*
- *pour chacun de ses sommets concepts s_c (de type c), deux attributs :*
  - *un attribut de type, indiquant l'attribut de la table c contenant les spécialisations (notées c'_i) du type de concept c attendues (le cas échéant), dans les instances de règle conformes à P, pour un sommet image de s_c (par l'opération de projection) ;*
  - *un attribut de marqueur, indiquant l'attribut de la table c contenant les marqueurs des types de concept c ou c'_i attendus (le cas échéant), dans les instances de règle conformes à P, pour un sommet image de s_c (par l'opération de projection).*

Remarque 4 *Les formules des requêtes associées à un patron de règle ne faisant que spécifier un schéma de requête, le résultat des deux requêtes est par définition identique, c'est-à-dire égal au nombre de données de la base. Un patron de règle a donc une validité de 100 %.*

4.4 Déroulement de la validation d'une instance de règle

Afin de tester la validité d'une règle experte, c'est-à-dire d'une instance de règle, ce qui est l'objectif recherché, deux nouvelles requêtes vont être construites automatiquement : une requête dénombrant les données satisfaisant l'hypothèse de l'instance de règle (appelée requête d'hypothèse) et une requête dénombrant les données satisfaisant à la fois l'hypothèse et la conclusion de l'instance de règle (appelée requête d'hypothèse et conclusion).

Ces requêtes sont composées de deux parties :

- leurs premières parties respectives décrivent le schéma de la requête à exécuter : il s'agit des requêtes associées au patron de règle auquel se conforme l'instance de règle à évaluer. Ces parties sont donc fournies par les annotations du patron de règle ;
- leurs secondes parties permettent de sélectionner les seuls n-uplets qui prennent les valeurs d'attributs correspondant aux spécialisations réalisées dans l'instance de règle. Ces parties spécifient donc des critères de sélection, qui vont être construits automatiquement en utilisant comme attributs de sélection les annotations du patron de règle (attributs de type et attributs de marqueur) et comme valeurs de sélection les types de concepts et les marqueurs présents dans l'instance de règle à évaluer.

Définition 15 *Soit P un patron de règle et I une instance de règle à valider, conforme à P.*

La requête d'hypothèse (respectivement d'hypothèse et conclusion) de I, notée Q_H (resp. $Q_{H\wedge C}$, est la conjonction de :

- *la requête d'hypothèse (resp. d'hypothèse et conclusion) associée à P ;*
- *l'ensemble des critères de sélection de la forme $attribut = valeur$ obtenus comme suit. Soit π une projection de P dans I. Soit $sc = [c, m]$ un sommet concept de l'hypothèse de P (resp. de P entier) et $sc' = [c', m']$ son image dans I par π.*
  - *Si $c' < c$ (au sens de la relation de spécialisation) alors un critère de sélection est créé, ayant pour attribut l'attribut de type associé à sc et pour valeur c' (type de concept de haut niveau issu des données, qui correspond à une valeur prise par l'attribut de type associé à sc). Si de plus c' a des sous-types, dans l'ensemble des types de concepts, alors pour chacun de ces sous-types c'' un critère de sélection est créé, ayant pour attribut l'attribut de type associé à sc et pour valeur c''.*
  - *Si $m' < m$ (au sens de la relation de spécialisation) alors un critère de sélection est créé, ayant pour attribut l'attribut de marqueur associé à sc et pour valeur m'.*

Remarque 5 *S'il existe plusieurs projections de P dans I, une requête d'hypothèse (resp. d'hypothèse et conclusion) de I est obtenue pour chaque projection. Seule la requête d'hypothèse (resp. d'hypothèse et conclusion) donnant le plus grand résultat (nombre de données) est retenue : on estime qu'elle correspond à la spécialisation escomptée du patron de règle.*

Exemple 9 *La requête SQL associée à l'hypothèse de l'instance de règle de la Fig. 2.*

```
SELECT COUNT(resultat.id_resultat)
FROM resultat, aliment, constituant, etude, operation
WHERE resultat.id_aliment = aliment.id_aliment
AND etude.id_operation = operation.id_ operation
AND resultat.id_sous_constituant = constituant.id_constituant
AND resultat.id_ propriete = propriete.id_propriete
AND resultat.id_etude = etude.id_etude
// Partie de la requete ajoutée à celle du patron (voir Exemple 7)
AND operation.nom_ operation = 'Cuisson à l'eau' propriete.nom_propriete = 'Teneur'
AND (constituant.nom_constituant = 'Vitamine'
// Partie correspondant à la prise en compte des sous-types de Vitamine (les types
Cuisson à l'eau et Teneur n'ont pas de sous-type dans cet exemple)
OR constituant.nom_constituant = 'Vitamine liposoluble'
OR constituant.nom_constituant = 'Vitamine E' ...)
```

Les requêtes Q_H et $Q_{H \wedge C}$ ont respectivement pour résultat n_H et $n_{H \wedge C}$, qui permettent de calculer le taux de validité de l'instance de règle. Les règles dont le taux de validité est strictement inférieur à 100 % ont des exceptions au sein de la base de données . Ces exceptions peuvent être sélectionnées et affichées à l'utilisateur.

Exemple 10 *Pour la règle de la Fig. 2, on a un taux de validité V de 97.5 % (voir Fig. 7 et 8).*

5 Application

Les méthodes présentées ont été mises en œuvre au sein d'une application concernant le contrôle de la qualité alimentaire. L'enjeu est d'améliorer la maîtrise des facteurs de contrôle de la qualité nutritionnelle. Après une présentation de l'environnement de travail, nous décrirons les données expérimentales et les connaissances expertes de l'application, puis leur validation.

5.1 Environnement de travail

Les données expérimentales sont regroupées au sein d'une base de données MySQL. La consultation et la saisie des données par des spécialistes du domaine se fait via un navigateur, au travers de formulaires PHP. La base de données contient à l'heure actuelle une trentaine de tables et les résultats détaillés d'environ 600 expériences.

Les règles expertes sont représentées à l'aide de l'interface graphique CoGUI (http ://www.lirmm.fr/ gutierre/cogui/). Environ 150 règles expertes sont disponibles, une vingtaine est utilisée pour tester la méthodologie proposée, à commencer par les cas les plus simples.

La communication entre les deux systèmes est établie à l'aide d'une connexion JDBC.

5.2 Description des données expérimentales

Destiné à des scientifiques et à des industriels de l'agroalimentaire, l'outil (en langue anglaise) de consultation et de saisie des données expérimentales intègre des données scientifiques, aussi exhaustives que possible, issues de la littérature traitant des qualités nutritionnelles des aliments à base de blé dur, et décrivant l'impact des procédés de transformation sur ces qualités. Un tel article scientifique comporte généralement les informations suivantes :

- des mesures expérimentales, notamment sur l'analyse des constituants nutritionnels (par exemple : dosage de la vitamine B1 dans les pâtes) ;
- des résultats sur l'impact d'une ou plusieurs opérations unitaires sur une ou plusieurs qualités nutritionnelles (effet de la cuisson-extrusion sur la teneur en minéraux) ;
- des données sur l'influence de certains paramètres de l'opération unitaire (par exemple : l'effet de la température de stockage sur la rétention des vitamines) ;
- des données sur l'influence d'autres opérations unitaires (par exemple : l'effet du type de séchage des pâtes sur la rétention des vitamines pendant la cuisson dans l'eau) ;
- des modèles décrivant l'évolution des qualités nutritionnelles (par exemple : la cinétique de dégradation thermique de la vitamine B1) ;
- des références bibliographiques.

5.3 Description des connaissances expertes

Les types de concepts du vocabulaire utilisé pour exprimer les connaissances expertes ont été obtenus comme présenté dans la partie 3. Dans la modélisation du support, l'essentiel de la sémantique est porté par les types de concepts. Les types de relations constituent quant à eux des connecteurs généraux aussi stables que possible. La figure 6 montre une partie de ce vocabulaire, créé à l'aide de CoGUI.

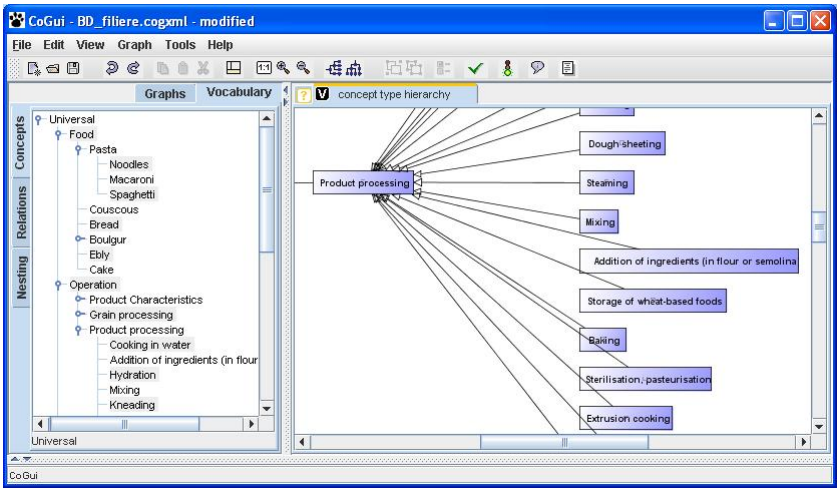

FIG. 6 – *Une partie du vocabulaire utilisé pour exprimer les connaissances expertes*

Les connaissances expertes, représentées par des règles de graphes conceptuels, expriment, pour chaque opération unitaire intervenant dans le process de fabrication d'un aliment à base de blé dur, et pour chaque constituant nutritionnel répertorié, l'impact ou les impacts connu(s) de cette opération sur ce constituant, en explicitant les conditions dans lesquelles cet impact semble se produire, pouvant faire intervenir des interactions avec d'autres opérations unitaires.

L'impact peut concerner la variation de la teneur du constituant (augmentation, diminution, stagnation) mais aussi la modification de propriétés qualitatives du constituant, telles que la digestibilité, l'allergénicité, etc.

5.4 Validation des connaissances expertes

L'évaluation des connaissances expertes peut être visualisée de deux façons par l'utilisateur : elle peut être effectuée individuellement règle par règle, et permet alors à l'utilisateur d'obtenir l'affichage des données expérimentales faisant exception à cette règle ; elle peut également être effectuée sous la forme d'un tableau récapitulatif de l'ensemble des règles déclarées dans l'application et de leurs taux de validité respectifs. Les figures 7 et 8 illustrent le premier cas de figure.

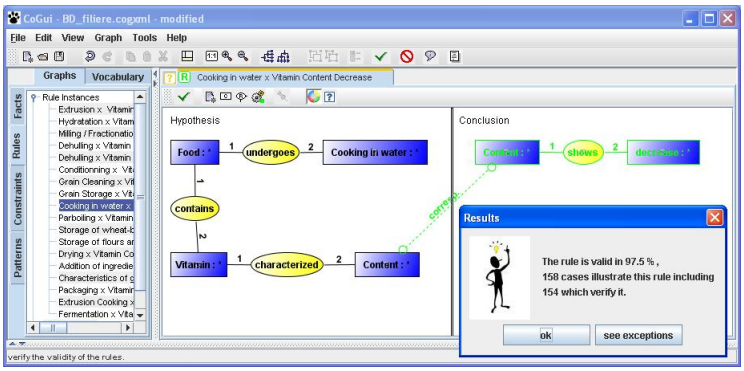

FIG. 7 – *Evaluation de la validité d'une règle experte*

Cooking in water x Vitamin Content Decrease
RULE EXCEPTIONS

Food	Parameters of the unit operation				Interactions with prior unit operations	Results				Reference
	Temperature(°C)	% of salt in water(%)	Time(mn)	Kind of water()		Name of the component	Percentage Value	Percentage Standard deviation	Kind of result	
Pasta	100	Undefined	20	Undefined	Yes	Vitamin A	4%		increase	Parrish, D. B. et al.,1980
Pasta	100	Undefined	20	Undefined	Yes	Vitamin A	8%		increase	Parrish, D. B. et al.,1980
Pasta	100	Undefined	20	Undefined	Yes	Vitamin A	8%		increase	Parrish, D. B. et al.,1980

FIG. 8 – *Affichage des exceptions d'une règle experte*

6 Conclusion et perspectives

Etant donnés deux types d'information hétérogènes disponibles pour un domaine (des connaissances expertes génériques exprimées par des règles de causalité d'une part, des résultats expérimentaux détaillés d'autre part) représentés dans deux formalismes distincts (respectivement le modèle des graphes conceptuels et le modèle relationnel), nous avons proposé dans cet article deux étapes pour la construction d'une expertise du domaine : (i) la génération d'une ontologie par l'identification de types de concepts de haut niveau au sein du schéma relationnel et au sein des données relationnelles et la hiérarchisation de ces types de concepts. Cette étape est automatique mais soumise à vérification experte ; (ii) l'évaluation de la validité des connaissances expertes au sein des données expérimentales. Cette étape est fondée sur la notion de patron de règle dans le formalisme graphes conceptuels, auquel est associé un "squelette" de requête SQL correspondant dans le formalisme relationnel. L'évaluation d'une instance de règle donnée conforme à un patron, se fait alors en complétant le "squelette" associé à ce patron par des critères de sélection spécifiques à l'instance de règle considérée. Cette étape est automatique, ce qui est permis par des annotations effectuées sur les patrons de règle.

La méthodologie proposée est ainsi fondée sur la coopération des deux types d'information et des deux formalismes hétérogènes. Elle est illustrée par un cas d'application concret.

L'objectif à plus long terme des règles de causalité est une utilisation à des fins d'aide à la décision : étant donnée une requête de l'utilisateur, exprimant un but souhaité, la question est de déterminer quelles conditions permettre d'atteindre ce but, en recherchant des règles dont la conclusion satisferait le but souhaité, et dont l'hypothèse fournirait des conditions suffisantes à son obtention.

Références

Baget, J.-F. et E. Salvat (2006). Rules dependencies in backward chaining of conceptual graphs rules. In *Proc. of ICCS'06*, pp. 102–116. Springer.

Bos, C., B. Botella, et P. Vanheeghe (1997). Modeling and Simulating Human Behaviors with Conceptual Graphs. In *Proc. of ICCS'97*, Volume 1257 of *LNAI*, pp. 275–289. Springer.

Euzenat, J., T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng-Kuntz, M. Ehrig, M. Hauswirth, M. Jarrar, R. Lara, D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris, S. Van Acker, et I. Zaihrayeu (2004). State of the art on ontology alignment. deliverable 2.2.3, Knowledge web NoE.

Folch, H., B. Habert, M. Jardino, N. Pernelle, M.-C. Rousset, et A. Termier (2004). Highlighting latent structure in documents. In *International Conference on Language Resources and Evaluation (LREC)*, Volume 4, pp. 1131,1334.

Genest, D. (2000). *Extension du modèle des graphes conceptuels pour la recherche d'informations*. Ph. D. thesis, Université Montpellier II.

Haemmerlé, O., P. Buche, et R. Thomopoulos (2006). The MIEL system : uniform interrogation of structured and weakly structured imprecise data. *Journal of Intelligent Information Systems*.

Haemmerlé, O. et B. Carbonneill (1996). Interfacing a relational database using conceptual graphs. In *DEXA '96 : Proceedings of the 7th International Workshop on Database and Expert Systems Applications*, Washington, DC, USA, pp. 499. IEEE Computer Society.

Kietz, J.-U., A. Maedche, et R. Volz (2000). A method for semi-automatic ontology acquisition from a corporate intranet. In *Proceedings of EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, October 2000*, Number 1937 in Springer Lecture Notes in Artificial Intelligence (LNAI).

Kolaitis, P. G. et M. Y. Vardi (1998). Conjunctive-Query Containment and Constraint Satisfaction. In *Proceedings of PODS'98*.

Muggleton, S. et L. D. Raedt (1994). Inductive logic programming : Theory and methods. *Journal of Logic Programming 19/20*, 629–679.

Mugnier, M.-L. (2000). Knowledge Representation and Reasoning based on Graph Homomorphism. In *Proc. ICCS'00*, Volume 1867 of *LNAI*, pp. 172–192. Springer.

Pernelle, N., M.-C. Rousset, et V. Ventos (2001). Automatic construction and refinement of a class hierarchy over multi-valued data. In *Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp. 386–398.

Salvat, E. (1998). Theorem proving using graph operations in the conceptual graphs formalism. In *Proc. of ECAI'98*, pp. 356–360.

Sendall, S. et W. Kozaczynski (2003). Model transformation : The heart and soul of model-driven software development. *IEEE Software 20*(5), 42–45.

Sowa, J. F. (1976). Conceptual Graphs. *IBM Journal of Research and Development*.

Sowa, J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley.

Termier, A., M.-C. Rousset, et M. Sebag (2002). TreeFinder : a First Step towards XML Data Mining . In *International Conference on Data Mining ICDM02*.

Tilley, T. A., R. J. Cole, P. Becker, et P. W. Eklund (2005). *A Survey of Formal Concept Analysis Support for Software Engineering Activities*. LNAI3626. Springer-Verlag.

Summary

This work takes place in the general context of the construction and validation of a domain expertise. It aims at the cooperation of two kinds of knowledge, heterogeneous by their levels of granularity and their formalisms: expert statements represented in the conceptual graph model and experimental data represented in the relational model. We propose to automate two stages: firstly, the generation of a simple ontology (terminological part of the conceptual graph model) guided both by the relational schema and by the data it contains; secondly, the evaluation of the validity of the expert statements within the experimental data. The method we introduce for that is based on the use of annotated conceptual graph patterns. These results were implemented within a real-life application concerning food quality control.