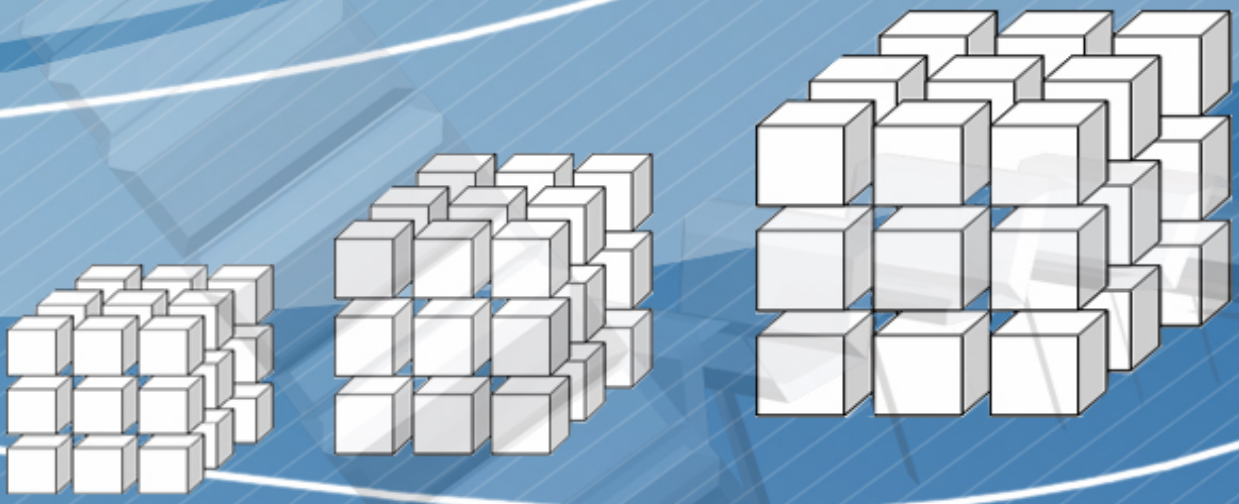
LES SYSTEMES DECISIONNELS

APPLICATIONS ET PERSPECTIVES

ASD 2008



Editeurs

Azedine BOULMAKOUL
Omar BOUSSAID
Jamel FEKI
Faïez GARGOURI

LES SYSTEMES DECISIONNELS

APPLICATIONS ET PERSPECTIVES

ASD 2008

Atelier des **S**ystèmes **D**écisionnels

Editeurs

Azedine BOULMAKOUL
Omar BOUSSAID
Jamel FEKI
Faiez GARGOURI

Préface

Les technologies des entrepôts de données et analyses en lignes sont au cœur des systèmes décisionnels modernes. Consciente du rôle des recherches dans cette thématique, la communauté scientifique internationale ne cesse d'accorder une importance de plus en plus grandissante, voire privilégiée, à ce domaine ce qui s'est traduit par l'apparition de nouvelles manifestations scientifiques venant enrichir le panorama des rencontres entre chercheurs et industriels.

Dans le prolongement des deux éditions précédentes (Agadir–Maroc, 2006 et Sousse–Tunisie, 2007), ASD 2008 (Atelier sur les Systèmes Décisionnels) ambitionne de consolider les expériences conduites par les chercheurs, industriels et utilisateurs issus de communautés travaillant avec les systèmes décisionnels. L'objectif de cette troisième édition de l'atelier, en particulier après le succès des deux premières éditions, est de contribuer à dynamiser davantage la recherche dans ce domaine et à créer une synergie entre les chercheurs, essentiellement mais non exclusivement maghrébins, travaillant dans leur pays ou dans des laboratoires de recherche à l'étranger. D'autre part, de renforcer les liens existants et de tisser de nouvelles relations afin de faire émerger une communauté thématisée systèmes décisionnels au niveau du Maghreb.

Ces actes regroupent les articles acceptés et présentés à cette troisième édition ASD. ASD 2008 a reçu 31 soumissions d'articles de nombreux pays (Algérie, Canada, France, Maroc, Suisse, Tunisie, &&&). Après évaluation par les membres du comité scientifique, composé par &&& experts internationaux du domaine, 21 articles ont été retenus. Ces derniers couvrent différents thèmes de recherche et d'application sur les systèmes décisionnels.

ASD 2008 a accueilli deux conférenciers invités : Karine Zeitouni, enseignant-chercheur, membre de l'équipe Systèmes de Bases de Données du laboratoire PRISM de l'Université de Versailles-Saint-Quentin (France), et Ahmed Zaidaoui consultant expert des solutions décisionnelles & Business intelligence chez Linkware (Maroc). Leurs exposés ont porté respectivement sur «l'exploitation des données spatio-temporelles pour l'aide à la décision» et «Business intelligence et pilotage de l'entreprise».

ASD 2008 a reçu le soutien de différentes institutions publiques d'enseignement et de recherche : le laboratoire ERIC de l'université Lumière Lyon2 (France), le laboratoire MIRACL de l'Université de Sfax (Tunisie), l'université HASSAN II Mohammedia, la Faculté des Sciences et Techniques de Mohammedia, le Centre National pour la Recherche Scientifique et Technique (CNRST), l'opérateur Maroc Telecom, le groupe ITISSAL Technologies, le groupe SIS Consultants, la société Open Visio. Nous sommes reconnaissants de leur soutien.

Ce travail a été réalisé avec l'aide du Ministère Marocain de l'Education Nationale, de l'Enseignement Supérieur, du Centre National pour la Recherche Scientifique et Technique et de la Formation des Cadres et de la Recherche Scientifique. Nous en sommes infiniment reconnaissants.

Le succès de cette troisième édition de ASD n'aurait pas été réalisé sans la coopération étroite du comité scientifique et des membres du comité d'organisation, que nous tenons également à remercier très chaleureusement.

travail publié avec le soutien du Ministère de l'Education Nationale, de l'Enseignement Supérieur, de la Formation des Cadres et de la Recherche Scientifique, sur les fonds gérés par le Centre National pour la Recherche Scientifique et Technique

Les éditeurs
A. BOULMAKOUL, O. BOUSSAID, J. FEKI, F. GARGOURI

Comité de pilotage

- Azedine BOULMAKOUL (FST, Université Hassan II, Mohammedia, Maroc)
- Omar BOUSSAID (ERIC, Université Lumière Lyon 2, Lyon, France)
- Jamel FEKI (MIRACL, Université de Sfax, Sfax, Tunisie)
- Faïez GARGOURI (MIRACL, Université de Sfax, Sfax, Tunisie)

Comité de programme

- Frederic ADAM (Business Information Sys., Univ. College Cork, Cork, Irlande)
- Zahia ALIMAZIGHI (USTHB, Alger, Algérie)
- Marie-Aude AUFAURE (SUPELEC, Paris, France)
- Thierry BADARD (CRG, Université de Laval, Laval, Canada)
- Abdelmajid BADRI (FST, Université Hassan II, Mohammedia, Maroc)
- Yvan BEDARD (CRG, Université de Laval, Laval, Canada)
- Bouziane BELDJILALI (Université Sénia, Oran, Algérie)
- Mostafa BELLAFKIH (Institut National des Postes et Télécom., Rabat, Maroc)
- Ladjel BELLATRECHE (ENSMA, Université de Poitier, Poitier, France)
- Hanêne BEN ABDALLAH (MIRACL, Université de Sfax, Sfax, Tunisie)
- Abdelmajid BEN HAMADOU (MIRACL, Université de Sfax, Sfax, Tunisie)
- Riadh BEN MESSAOUD (Université 7 novembre à Carthage, Tunis, Tunisie)
- Nabila BENHARKAT (LIRIS, Université Lyon 1, Lyon, France)
- Djamel BENSLIMANE (LIRIS, Université Lyon 1, Lyon, France)
- Fadila BENTAYEB (ERIC, Université Lumière Lyon 2, Lyon, France)
- Ilham BERRADA (ENSIAS, Rabat, Maroc)
- Maurizio BIELLI (NRC, Institute of Syst. Analysis and Informatics, Rome, Italie)
- Rafik BOUAZIZ (MIRACL, Université de Sfax, Sfax, Tunisie)
- Omar BOUCELMA (LSIS, Université d'Aix-Marseille III, Marseille, France)
- Azedine BOULMAKOUL (FST, Université Hassan II, Mohammedia, Maroc)
- Omar BOUSSAID (ERIC, Université Lumière Lyon 2, Lyon, France)
- Jérôme DARMONT (ERIC, Université Lumière Lyon 2, Lyon, France)
- Farid EL HEBIL (Institut National des Postes et Télécom., Rabat, Maroc)
- Cécile FAVRE (Laboratoire ERIC, Université Lumière Lyon 2, Bron, France)
- Jamel FEKI (MIRACL, Université de Sfax, Sfax, Tunisie)
- Faïez GARGOURI (MIRACL, Université de Sfax, Sfax, Tunisie)
- Yasser HACHAICHI (MIRACL, Université de Sfax, Sfax, Tunisie)
- Mohand-Saïd HACID (LIRIS, Université de Claude Bernard, Lyon, France)
- Nouria HARBI (ERIC, Université Lumière Lyon 2, Lyon, France)
- Mohamed JANATI (ENSIAS, Rabat, Maroc)
- Daniel LEMIRE (Université du Québec à Montréal, Montréal, Canada)
- Sabine LOUDCHER (ERIC, Université Lumière Lyon 2, Lyon, France)
- Mimoune MALKI (Université de Sidi Bel-Abbès, Sidi Bel Abbes, Algérie)
- Rokia MISSAOUI (LARIM, Univ. du Québec en Outaouais, Outaouais, Canada)

- Rachid OULAD HAJ THAMI (ENSIAS, Univ. Mohammed V, Rabat, Maroc)
- Thurasamy RAMAYAH (Lab, Univ. Sains Malaysia, Pulau Pinang, Malaisie)
- Frank RAVAT (IRIT, Université de Toulouse III, Toulouse, France)
- Sahbi SIDHOM (LORIA, Université de Nancy 2, Nancy, France)
- Olivier TESTE (IRIT, Université de Toulouse III, Toulouse, France)

Comité d'organisation

- Mohammed ASSOU (FST, Université Hassan II, Mohammedia, Maroc)
- Riadh BEN MESSAOUD (Université 7 novembre à Carthage, Tunis, Tunisie)
- Azedine BOULMAKOUL (FST, Université Hassan II, Mohammedia, Maroc)
- Yasser HACHAICHI (MIRACL, Université de Sfax, Sfax, Tunisie)
- Farid EL HEBIL (Institut National des Postes et Télécom., Rabat, Maroc)
- Abdelfatah IDRI (FST, Université Hassan II, Mohammedia, Maroc)
- Rabia MARGHOUBI (Institut National des Postes et Télécom., Rabat, Maroc)
- Meriem MANDAR (FST, Université Hassan II, Mohammedia, Maroc)
- Mohamed RAMDANI (FST, Université Hassan II, Mohammedia, Maroc)

Tables matières

Vers une expression des besoins décisionnels en langage naturel

Fahmi Bargui, Jamel Feki, Hanene Ben-Abdallah
Laboratoire MIRACL
Département d'Informatique, Faculté des Sciences Economiques et de Gestion de Sfax, Route de l'Aéroport Km 4 – 3018 Sfax, BP. 1088
{fahmi.bargui, feki.jamel, hanene.benabdallah}@fsegs.rnu.tn

Résumé. Dans cet article, nous nous intéressons à l'ingénierie des besoins décisionnels. Particulièrement, c'est l'expression de ces besoins en langage naturel qui est étudiée. Afin de limiter les difficultés émergentes de la spécification des besoins analytiques en langage naturel, et au moment de la validation et du traitement automatique de ces besoins, nous proposons un modèle et une grammaire simplifiée, élaborés suite à une étude empirique.

1 Introduction

Dans le cadre de l'automatisation des étapes de développement d'un système décisionnel, la plupart des recherches se focalisent sur les transformations du MPIM (« Multidimensionnel Platform Independent Model ») vers MPSM (« Multidimensionnel Platform Specific Model ») (MDA, 2003) et réservent peu d'efforts d'investigation à la modélisation du MCIM (« Multidimensionnel Computation Independent Model ») et à sa transformation vers le MPIM. Ce phénomène peut être expliqué essentiellement par la présence des technologies mûres pour la représentation et la formalisation des modèles MPIM et MPSM tels que : UML (UML, 2007), schéma en étoile (Kimball, 1996), modèle relationnel (Codd,1970), XML... Par ailleurs, nous constatons, qu'en général, la modélisation du niveau CIM manque de formalisme et en particulier dans le domaine décisionnel.

L'objectif de cet article est de proposer un modèle MCIM pour les besoins analytiques. Pour l'élaboration de ce modèle, nous nous sommes fixés trois recommandations importantes : *i)* situer ce modèle à un haut niveau d'abstraction afin de garantir sa stabilité ; *ii)* utiliser des concepts faciles à comprendre non seulement par les décideurs, experts dans le domaine de prise de décisions, mais aussi par l'équipe de développement du SID. Ceci permet d'éviter les coûts de formation des experts sur les nouveaux concepts, et facilite la communication entre les différents intervenants ; et *iii)* assurer la traçabilité entre les modèles MCIM et le MPIM.

Tenant compte de ces trois recommandations, nous nous sommes orientés vers le langage naturel comme moyen de spécification des besoins analytiques vu qu'il est susceptible de jouer un rôle pivot entre les différents acteurs d'un SID. Dans ce cadre, nous avons examiné deux alternatives. La première consiste à utiliser un gabarit (« Template ») qui sera rempli par le décideur avec des termes métiers correspondant à des concepts multidimensionnels : ce qu'il souhaite analyser, suivant quels axes… Cette solution faciliterait l'analyse et la dérivation du MPIM, mais limiterait le champ d'application du CIMM à quelques requêtes types et restreindrait le champ d'expression du décideur. La seconde alternative est à base d'une syntaxe en langage naturel libre ; c'est-à-dire, sans fixer un style d'écriture ce qui reste un grand défi à lever. Dans ce travail, nous avons opté pour une solution de compromis entre ces deux extrêmes. En fait, elle consiste à définir une grammaire en procédant à une étude

empirique d'un ensemble de besoins analytiques. Cette grammaire est constituée d'un ensemble de patrons linguistiques formalisant les styles d'écritures communs et fréquemment utilisés dans la spécification de ces besoins.

Cet article traite de cette problématique. Il est organisé comme suit : la section 2 étudie l'état de l'art des spécifications des besoins analytiques et introduit nos motivations ; la section 3 propose un modèle pour la spécification des besoins analytiques ; la section 4 définit une grammaire formalisant la spécification des besoins ; la section 5 illustre, à travers un exemple, le passage vers le schéma multidimensionnel à partir de besoins exprimés. Finalement, la section 6 récapitule cet article et enchaîne sur ses perspectives.

2 Etat de l'art et motivations

Cette section examine les travaux les plus pertinents en conception de SID et que nous classifions en trois catégories.

Approches dirigées par les données (« data-driven approaches »)

Ces approches partent du modèle du système d'information opérationnel et appliquent un ensemble de règles de transformation pour construire des schémas multidimensionnels, *cf.,* Cabibbo et al. (1998), Golfarelli et al. (1998), Hüsemann et al. (2000), Moody et al. (2000), etc.

Toutefois, ces approches exigent des décideurs une bonne expertise dans les modèles des systèmes opérationnels et une parfaite connaissance des sources de données. De plus, malgré que ces approches puissent atteindre un bon niveau d'automatisation, il se pose le problème de marginalisation des besoins analytiques durant la conception du SID. Ce problème constitue souvent la principale cause d'échec d'un projet décisionnel Giorgini et al. (2007).

Approches dirigées par les besoins (« requirement-driven approaches »)

Ces approches dites descendantes, sont initiées par les travaux de (Kimball, 1996). De même, Bruckner et al. (2001), Schiefer et al. (2002) ont proposé une approche guidée par les processus, limitée à l'analyse et la documentation des besoins analytiques. Ces derniers sont analysés selon trois perspectives associées à des niveaux d'abstractions différents dont chacun est spécifié par un gabarit (*e.g.,* besoins métiers, besoins utilisateurs et besoins techniques). Bien que les auteurs se sont concentrés sur la documentation des besoins analytiques, considérée comme une phase très importante dans le processus de développement du SID, ils se sont limités à indiquer les constituants de chaque gabarit, sans donner des détails sur la manière dont il faut les présenter ni des exemples illustrant leur approche. De plus, les auteurs ne proposent pas une démarche pour assister les concepteurs.

Par ailleurs, Paim et Castro (2003) ont proposé le processus DWARF (DataWarehouse Requirements deFinition) pour l'identification, la spécification et la validation des besoins analytiques. Pour guider le concepteur dans la réalisation de chaque phase de DWARF, les auteurs ont adopté des techniques connues du génie logiciel comme les interviews, le prototypage et les scénarii pour l'identification des besoins ; les cas d'utilisation avec une description textuelle pour la documentation des exigences, etc. Bien que ces techniques étaient de grand usage dans de développement des systèmes d'information, de nos jours d'autres techniques et approches (*e.g.,* approches dirigées par les buts) ont montré leur efficacité. Néanmoins, aucun formalisme n'a été proposé pour la spécification des exigences des décideurs.

D'autre part, Feki et al. (2008) ont proposé une approche automatisant la génération d'un schéma multidimensionnel à partir des besoins OLAP. Malgré que les auteurs ont étudié des algorithmes pour cette génération, ils présument que le décideur est assez compétent pour, d'une part, identifier ses besoins OLAP et, d'autre part, les exprimer suivant un format tabulaire prédéfini.

Approches mixtes (« data and requirement driven approaches»)

Ces approches résultent d'une combinaison des deux précédentes afin de profiter de leurs avantages. Parmi ces approches, Winter et al. (2003, 2004) ont proposé une démarche permettant l'analyse des exigences d'information (« requirement information ») des décideurs et leurs associations avec les sources de données. Les auteurs décrivent textuellement ce qui doit être réalisé dans chacune des étapes de la démarche, sans détailler comment le réaliser. Ainsi, ils n'ont pas suggéré des techniques, des directives ou des modèles permettant d'identifier, de documenter ou de valider ces exigences. De plus, ils n'ont pas précisé comment associer les sources de données avec les besoins identifiés afin de créer l'entrepôt de données.

D'autre part, Giorgini et al. (2005, 2007) ont proposé une approche mixte, dirigée par les buts, pour dériver un schéma multidimensionnel à partir de deux modèles élaborés par le concepteur décisionnel selon deux perspectives différentes : *modélisation organisationnelle* centrée sur les stackeholders et *modélisation décisionnelle* focalisée sur les décideurs. La génération du schéma multidimensionnel tient compte aussi bien des besoins analytiques spécifiés dans le modèle décisionnel que des sources de données spécifiées dans le modèle organisationnel. Le modèle décisionnel est obtenu en décomposant les buts métiers des décideurs d'un niveau d'abstraction très élevé en sous buts de niveaux d'abstractions intermédiaires. Ce processus de décomposition est itéré jusqu'à la construction d'un arbre de buts dont les feuilles indiquent les exigences fonctionnelles des décideurs. Ensuite, chaque besoin fonctionnel est enrichi par des détails techniques spécifiant les dimensions, les mesures et les faits. Malgré que le modèle décisionnel permette de capturer les exigences des décideurs, l'arbre des buts peut contenir un nombre important de nœuds ce qui le rend parfois illisible. De plus, les concepts utilisés pour spécifier les besoins fonctionnels concernent des détails techniques et relatifs au contexte des entrepôts de données, ce qui complique leurs validations par les décideurs non experts en entreposage de données. Ainsi, la validation des besoins risque de ne pas aboutir à son objectif ce qui augmente le risque d'échec du projet.

Cette étude de l'état de l'art montre que : (*i*) il existe un manque de formalismes appropriés pour les besoins analytiques, qui devrait être pratique pour le concepteur et facile à comprendre et à valider par le décideur, et (*ii*) les modèles proposés n'adhèrent pas à la démarche MDA, c'est-à-dire qu'ils assurent trop peu ou pas du tout, la traçabilité des besoins.

La suite de cet article introduit notre démarche d'élaboration d'un modèle basé sur le langage naturel pour la spécification des besoins analytiques.

3 Un modèle pour les besoins analytiques

D'après List et al. (2000) un modèle de spécification des besoins analytiques doit posséder les caractéristiques suivantes :

1. Haut niveau d'abstraction : le modèle des besoins doit capturer directement les concepts métiers des décideurs. Les détails techniques (e.g., dimensions, mesures, faits) sont conçus pour spécifier la solution conceptuelle, et non pas les besoins des décideurs.
2. Complétude : toutes les informations nécessaires à la prise de décisions doivent être capturées.
3. Lisibilité : le langage naturel utilisé pour la documentation des besoins est facilement compréhensible par le décideur. Ce qui facilite la communication entre tous les intervenants, notamment en validation des besoins analytiques.
4. Précision : l'utilisation des formules mathématiques pour la spécification des indicateurs et d'une grammaire pour la description des requêtes types apportent une meilleure précision aux besoins.
5. Traçabilité : Elle désigne en quoi un élément du modèle des besoins correspond dans chaque modèle produit lors du cycle de développement d'un logiciel Spanoudakis et al. (2004).

Chaque modèle produit lors du développement d'un SID doit être guidé par le processus de prise de décisions Prakash et al. (2008). Dans la suite de cette section nous décrivons le processus de pilotage, les concepts identifiés sont écrits en style gras. Ensuite, nous proposons un modèle, décrit par le tableau TAB.1, permettant de capturer ces concepts.

Selon (thomson, 2002) une décision est intentionnelle ; chaque décision justifiable est basée sur au moins un objectif et une prédiction. D'après Mard et al. (2004, p132), **un objectif** doit être mesurable par une valeur prévisionnelle **cible** (*e.g.,* un taux, une quantité, un montant) qu'un **processus** doit atteindre dans une période donnée. La valeur réalisée d'un processus est mesurée par un **indicateur de mesure de performance**. L'analyse de l'écart entre les valeurs (réalisée et cible) permet au **décideur** de juger la performance du processus. En cas d'écart négatif, le décideur constate une anomalie et procède à l'identification de ses causes. Pour cela, il a besoin d'analyser des informations détaillées du processus analysé disponibles dans l'ED et accessibles par des **requêtes types** formulées par les décideurs.

Le tableau TAB.1 montre les concepts métiers utilisés par le décideur dans le processus de prise de décisions. Ces concepts permettent la description d'un besoin analytique.

Processus	<identification d'un processus>
Acteur	<nom d'un responsable chargé de contrôler un processus>
Objectif	< résultat qu'un processus doit atteindre dans une période donnée >
Indicateur	**libellé** : <identification d'un indicateur>
	formules : < expressions de calcul >
	cible : <une valeur indiquant le niveau acceptable d'un indicateur >
Scénario d'informations	<description des requêtes types>

TAB. 1- Modèle pour la spécification des besoins analytiques

Expliquons chaque concept de notre modèle :

- **Processus :** ensemble d'activités reliées entre elles par des échanges d'informations et contribuant à la fourniture d'une même prestation à un client interne ou externe à l'entreprise.
- **Acteur** : nom d'un responsable chargé de contrôler la performance d'un processus.

- **Objectif** : décrit un objectif mesurable qu'un processus doit atteindre dans une période donnée.
- **Indicateur** : fournit une valeur réalisée par un processus. Pour chaque indicateur, les informations suivantes doivent systématiquement être explicitées.
  - **libellé :** nom d'un indicateur
  - **Formules :** un indicateur possède une formule principale et des formules secondaires, introduites par le symbole \, pour le calcul des opérandes (par exemple *CA=prix de vente * quantité* pour l'indicateur chiffre d'affaires). Chaque formule est une expression de calcul utilisant des : opérateurs arithmétiques, fonctions d'agrégation, signes de ponctuations, barres verticales, valeurs numériques, caractères, opérandes et symboles.
  - **cible :** une valeur prévisionnelle, désignant le niveau acceptable de l'indicateur.
- **Scénario d'informations** : ensemble de requêtes types, exprimées en langage naturel, permettant d'extraire les informations pertinentes pour l'analyse et la prise de décisions.

Les éléments identifiés dans la ligne 4 du tableau TAB1 (*i.e.,* libellé et formule) seront présents dans les requêtes analytiques. Ces requêtes seront formalisées par une grammaire que nous présentons dans la section 4.

4 Langage naturel pour l'expression des besoins analytiques

D'après (Kimball, 1996) et Paim et al. (2003), le langage naturel est le meilleur moyen d'expression des besoins analytiques car il facilite la communication avec les décideurs. Cependant, si plusieurs styles d'écriture sont employés alors des ambigüités peuvent être rencontrées. Afin de limiter les difficultés inhérentes, nous fixons un style d'expression de ces besoins tout en préservant les avantages du langage naturel (*i.e.,* expressivité, simplicité…). Pour ce faire, nous définissons une grammaire simplifiée dont l'élaboration a nécessité la collecte et l'étude d'un ensemble de requêtes types rédigées en langage naturel du style suivant :

1. Analyser le chiffre d'affaires par catégorie de produit par jour et année.
2. Afficher le nombre total d'heures supplémentaires par enseignant et par semestre.
3. Etudier l'évolution du nombre de mortalité des volailles par poulailler d'élevage et par date.
4. Comparer le taux annuel de factures non réglées par catégorie et code postale d'un client durant un exercice comptable avec le taux annuel acceptable des factures non réglées.

Par ailleurs, nous avons constaté que les décideurs utilisent souvent, dans la description de leurs besoins, des expressions très marquantes. Par exemple, pour introduire les *axes d'analyses*, ils emploient des prépositions (*e.g.,* par, pour, selon, durant…). De même, pour spécifier les *propriétés* décrivant les axes d'analyses, ils utilisent des groupes nominaux ayant des structures grammaticales récurrentes et simples comme un *nom* (*e.g.,* enseignant, date, semestre…), *nom-préposition-[déterminant]-nom* (*e.g.,* catégorie de produit, poulailler d'élevage, catégorie d'un client…), etc.

Dans le reste de cet article, nous appelons *patrons linguistiques* l'ensemble des structures grammaticales employées pour la description des axes d'analyses. Nous fixerons les patrons

linguistiques pour la spécification de ces axes d'analyses (*cf.* 4.1) et nous définirons une grammaire pour la spécification des requêtes types (*cf.* 4.2).

4.1 Patrons linguistiques pour les axes d'analyses

Dans la littérature de la conception multidimensionnelle, les axes d'analyses proviennent des entités du SI opérationnel. En réalité, ces entités sont décrites textuellement dans le Dictionnaire de Données (DD) du SI opérationnel. Nous avons alors recueilli et étudié plusieurs de ces descriptions et nous avons identifié leurs structures grammaticales, récurrentes. Pour couvrir un grand nombre de structures, nous avons examiné 4000 groupes nominaux décrivant des propriétés provenant d'une centaine de DD élaborés dans le cadre de projets de fin d'études et appartenant à neuf domaines distincts. Suite à l'examen de ces groupes nominaux, nous avons pu dégager neuf patrons linguistiques assez fréquemment utilisés. Le tableau TAB.2 donne les fréquences d'apparitions de ces différents patrons par domaine.

Structure Grammaticale \ domaine	Commerciale	Médical	Commerce électronique	Comptabilité	Enseignement	Assurance	Finance	Ressource Humaine	Production	Pourcentage
1. [dét] NOM Exemple : client	190	60	87	8	83	18	25	36	35	**13.55**
2. [dét] NOM prép dét NOM Exemple : nom d'un client	584	140	237	9	227	36	36	107	102	**36.95**
3. [dét] NOM prép NOM prép dét NOM Exemple : adresse de livraison d'un client	73	21	43	1	23	6	7	14	13	**5.03**
4. [dét] NOM adj prép dét NOM Exemple : code postale d'un client	48	4	15	4	12	5	4	7	12	**2.78**
5. [dét] NOM prép dét NOM adj Exemple : désignation d'un acte médical	19	29	14	6	5	6	9	5	2	**2.38**
6. [dét] NOM prép dét NOM prép NOM Exemple : numéro d'un bon de sortie	62	10	6	7	14	11	11	6	9	**3.40**
7. [dét] NOM prép dét NOM pp Exemple : ancienneté d'un ouvrier qualifié	18	11	13	10	13	10	14	18	17	**3.10**
8. [dét] NOM pp prép dét NOM Exemple : quantité entrée d'un article	30	20	15	4	23	8	1	10	1	**2.80**
9. [dét] NOM prép NOM adj prép dét NOM Exemple : carte d'identité nationale d'un client	3	2	1	0	1	0	0	2	2	**0.28**
Autres structures complexes	260	150	300	10	210	0	50	150	60	**29.75**

*TAB. 2- Statistiques des structures grammaticales identifiées dans l'échantillon étudié (**adj**ectif qualificatif, **dét**erminant, **p**articipe **p**assé et **prép**osition).*

Dans l'échantillon étudié, environ 70% (lignes 1 à 9 de TAB.2) des groupes nominaux sont concis et précis. De plus, ils adhèrent à des structures grammaticales simples constituées de :

- *Adjectif qualificatif* : Dans notre contexte, nous avons remarqué que ces adjectifs écartent les ambiguïtés dues à l'usage du langage naturel. Par exemple, dans le groupe nominal *carte d'identité nationale*, si nous supprimons l'adjectif *nationale* le sens devient ambigüe : la *carte d'identité* peut être *bancaire* ou *scolaire.*
- *Participe passé* : forme verbale utilisée avec l'auxiliaire dans la formation des temps composés et de la forme passive ; sans auxiliaire le participe passé peut fonctionner comme adjectif. Dans notre contexte, nous avons remarqué que l'emploie du participe passé s'est limité au rôle d'un adjectif qualificatif. Dans les groupes nominaux, *quantité entrée, ouvrier qualifiée, quantité commandée*…les mots *entrée, qualifiée et commandée* jouent le rôle d'un adjectif qualificatif. Leur suppression entraîne une ambiguïté.
- *Nom* : désigne une entité concrète ou abstraite comme *client*, *adresse.*
- *Déterminant* : caractérise un nom pouvant être un article défini ou indéfini (*e.g.,* un, une, le, la,…).
- Préposition : mot invariable qui introduit, selon un certain rapport de sens, un complément de verbe, de nom, d'adjectif ou d'adverbe.

De plus, nous avons constaté que les neuf patrons linguistiques identifiés (lignes 1 à 9 de TAB.2) désignent des concepts multidimensionnels. Dans la ligne 1 du tableau TAB.2, la structure grammaticale *déterminant nom* décrit une entité qui désigne une dimension. Par exemple le groupe nominal *un client* indique la dimension client. Dans les lignes 2 à 9 de TAB.2, la structure *préposition déterminant* est utilisée pour la séparation entre deux groupes nominaux. Le premier désigne un attribut dimensionnel et le deuxième désigne une dimension. A titre d'exemple, le groupe nominal *code postale d'un client* ayant la structure *nom-adjectif qualificatif-préposition-déterminant-nom* (ligne 3 de TAB.2), génère une dimension *client* (ayant la structure *nom*) et un attribut dimensionnel *code postale* (ayant la structure *nom-adjectif qualificatif*).

En outre, dans les neuf patrons identifiés, un attribut dimensionnel ou une dimension est spécifié par un groupe nominal selon l'une des structures suivantes : *déterminant nom*, *nom préposition nom*, *nom adjectif qualificatif*, *nom participe passé* et *nom préposition nom adjectif qualificatif*. Dans ces cinq structures, nous remarquons qu'un *nom* est toujours suivi d'une *préposition,* d'un *adjectif qualificatif* ou d'un *participe passé*. Les neuf patrons linguistiques identifiés peuvent alors être fusionnés en vue de définir un groupe nominal générique noté *GN1* (cf. 4.2).

Les 30% restants des groupes nominaux (dernière ligne de TAB.2) sont des écritures assez longues et peu utiles, composées entre autres de mots ayant les quatre catégories grammaticales : adjectifs non qualificatifs (démonstratifs, indéfinis, interrogatifs, etc.), adverbe, pronom (possessifs, démonstratifs, indéfinis, etc.) et les entités nommées. De plus, nous avons remarqué que les mots de ces quatre catégories grammaticales ne correspondent à aucun concept multidimensionnel. Par exemple, dans le groupe nominal *les quatre dernières années*, ayant comme structure grammaticale *déterminant-adjectif numéral cardinal-adjectif numéral ordinal-nom*, seul le mot *années*, correspond à un attribut dimensionnel, les autres mots sont inutiles. Ainsi, les groupes nominaux peuvent être réécrits d'une manière plus concise et simple en utilisant les neuf patrons linguistiques retenus et décrits par GN1 (cf. 4.2).

4.2 Proposition d'une grammaire pour la description des requêtes types

La fusion des neuf patrons linguistiques identifiés dans la section (4.1) produit la structure grammaticale **GN1** :

GN1 ::= NOM (NOM| adjectif qualificatif| participe passé| préposition)*

Un axe d'analyse est décrit par la structure grammaticale GN utilisant GN1 selon la syntaxe suivante :

GN ::= [déterminant] GN1 [,[déterminant] GN1,… et [déterminant] GN1] [préposition déterminant GN1]

Rappelons que la structure `préposition déterminant` est utile pour la séparation entre les groupes nominaux désignant des attributs dimensionnels et les groupes nominaux désignant des dimensions. Cette structure facilitera la segmentation de la phrase en vue de l'identification des différents groupes nominaux. Nous présentons ci-dessous des exemples décrivant des axes d'analyses conformes à la syntaxe GN :

- *un client, numéro de carte d'identité national et nom d'un client...*

L'étude des structures grammaticales des groupes nominaux utilisés pour la spécification de la dimension temps (*e.g., jour et année*.) a montré l'absence de la structure `préposition déterminant` comme séparateur entre le groupe nominal désignant la dimension temps et le groupe nominal désignant l'attribut dimensionnel temps.

L'absence du groupe nominal désignant explicitement le nom de la dimension temps, cause des problèmes au niveau de la segmentation de la requête posée par le décideur et cause des erreurs d'analyse et de génération du schéma multidimensionnel. Ceci est dû au fait que la dimension temps est implicitement annoncée par : jour, année, semestre, etc. Afin de contourner cette difficulté, nous avons opté d'exprimer la dimension temporelle conformément à GN comme suit : *jour et année d'une date* ayant la structure grammaticale `nom et nom préposition déterminant nom`.

Chaque phrase décrivant une requête type (*cf.* 4) peut être rédigée suivant la grammaire de la figure FIG.1 où les crochets indiquent les éléments optionnels et le caractère * désigne zéro ou plusieurs occurrences d'un élément.

Besoin ::= Verbe [GM] indicateur [GM] (marqueur_dimensionel GN)* [marqueur_comparatif GM]
GN ::= [déterminant] GN1 [,[déterminant] GN1,… et [déterminant] GN1] [préposition déterminant GN1]
GN1 ::= [déterminant] NOM (NOM| adjectif |participe passé| préposition)*
Verbe ::= analyser | comparer | étudier | suivre |…
GM ::= chaine de caractères
Indicateur ::= chaîne de caractères de mots clés
Marqueur_dimensionel ::= durant | en fonction de | selon | suivant | par | pour
Marqueur_comparatif ::= avec | par rapport à
Préposition ::= de | à | dans | en | chez | concernant | sur | depuis
Adjectif ::= adjectif qualificatif
Déterminant ::= un | une | des| du | de la| de l'| le | la | les

FIG.1- Grammaire pour la spécification des besoins analytiques

Dans la figure FIG.1, le symbole GM désigne une chaine de caractères, employée pour compléter la sémantique de la requête type, mais ne désignant pas des concepts multidimensionnels. L'exemple ci-dessous montre une requête écrite en langage naturel libre *R* et sa requête transformée R^T conforme à notre grammaire.

R : *Etudier l'évolution du nombre de mortalité des volailles par poulailler d'élevage et par date*.

R^T : *Etudier* (Verbe) *l'évolution du* (GM) *nombre de mortalité des volailles* (indicateur) *par* (marqueur_dimensionel) *poulailler d'un élevage* (GN) *par* (marqueur_dimensionel) *date* (GN).

De cette requête, nous pouvons identifier une dimension *élevage* et son attribut dimensionnel *poulailler*, et la dimension *date.*

Dans la section suivante nous proposons des directives pour la génération du schéma multidimensionnel à partir des besoins analytiques.

5 Génération du schéma multidimensionnel

Dans cette section, nous illustrons à travers un exemple la correspondance entre les éléments de notre modèle des besoins et ceux du schéma multidimensionnel. Cette correspondance assure, d'une part, la traçabilité du modèle de besoins. D'autre part, elle constitue un premier pas vers l'identification des règles de génération du schéma multidimensionnel à partir du modèle des besoins.

Le tableau TAB.3 décrit un exemple d'un besoin analytique exprimé par un directeur commercial qui vise à maximiser les ventes. Pour ce faire, il se fixe un ensemble de stratégies (*e.g.,* recrutement de représentants commerciaux, octroi de remises pour les factures dépassant un montant de 5000 DT ou des facilités de payement…). Pour mesurer la performance de ce processus, le décideur choisit l'indicateur taux mensuel de croissance du chiffre d'affaires. (Le tableau TAB.3 montre toutes les informations concernant cet indicateur.) De plus, le décideur choisit les informations nécessaires à cette analyse. La partie scénarii de l'exemple montre les exigences en informations exprimées sous forme de requêtes types.

Processus	Vente
Acteur	Directeur commercial
Objectif	Maximiser les ventes
Indicateur	**libellé** : taux de croissance mensuel du chiffre d'affaires
	Formule : $CA_m - CA_{m-1} \; / \; CA_{m-1}$ \ CA=prix de vente * quantité de vente mensuelle
	Référence : nous fixons un taux de croissance acceptable de 5 %
Scénarii	(1) Analyser le taux de croissance mensuel du chiffre d'affaires par ville et pays d'une succursale par mois d'une date. (2) Etudier le taux de croissance mensuel du chiffre d'affaires par nom, prénom et adresse d'un responsable de vente durant le mois d'une date. (3) Evaluer le taux de croissance mensuel du chiffre d'affaires par sous-catégorie et catégorie d'un produit selon le mois d'une date.

TAB. 3- Exemple d'un besoin analytique (CA_m désigne le chiffre d'affaires du mois m)

Par ailleurs, un schéma en étoile permet l'analyse détaillée d'un processus métier, (Kimball, 1996) et (Adamson, 2006). Généralement, ce schéma est constitué d'un fait central composé de mesures enregistrées par rapport à des dimensions. Selon notre démarche le nom d'un fait correspond au nom du processus à évaluer. Dans notre exemple, le nom du *processus vente* correspond au nom du *fait vente*.

Généralement, les mesures sont des attributs numériques qui servent au calcul des indicateurs agrégés. Ainsi, nous pouvons associer les mesures à la formule de calcul de l'indicateur. De plus, selon (Kimball, 1996), les données atomiques permettent une flexibilité maximale des analyses. Par conséquent, nous associons les opérandes des formules secondaires (dénotées par le caractère \) aux mesures. Dans notre exemple (TAB.3), les mesures *prix de vente* et *quantité de vente mensuelle* proviennent de la *formule secondaire* du CA.

Les dimensions et les attributs dimensionnels sont associés aux requêtes types et plus particulièrement aux groupes nominaux rédigés conformément à la syntaxe de GN (cf. 4.2). La `préposition déterminant` divise GN en deux groupes nominaux, le premier désigne des attributs dimensionnels et le deuxième indique une dimension. A titre d'exemple, la première requête de l'exemple génère deux groupes nominaux :

1. *ville et pays d'une succursale* génère les attributs dimensionnels *ville* et *pays* et la dimension *succursale*.
2. *mois d'une date* génère l'attribut dimensionnel *mois* et la dimension *date*.

En raisonnant de même pour les deux autres requêtes, nous obtenons le schéma en étoile de la figure FIG.2.

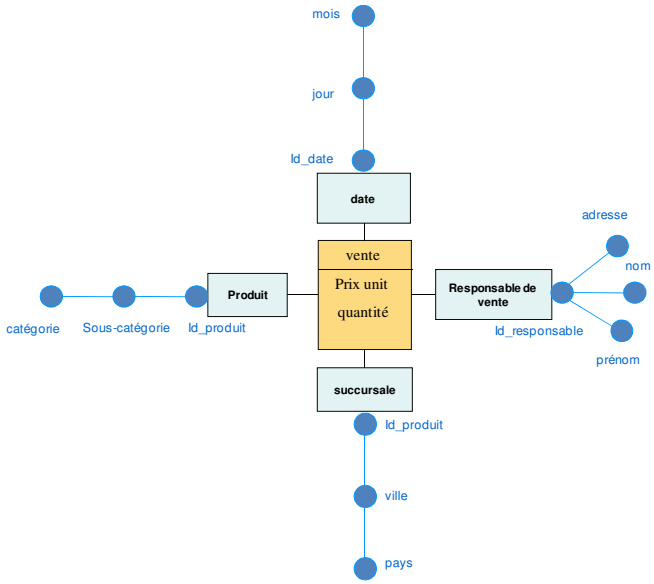


FIG. 2- Schéma en étoile construit pour l'exemple du tableau TAB.3.

6 Conclusion

Dans cet article nous avons proposé un modèle pour la spécification des besoins analytiques, approprié au contexte des systèmes d'information décisionnels. Ce modèle permet de capturer les concepts métiers du processus de prise de décisions et facilite la validation des besoins par les décideurs. De plus, nous avons formalisé les requêtes analytiques par une grammaire définie suite à l'étude empirique d'un corpus de phrases extraites à partir de plusieurs dictionnaires de données. Notre grammaire permet d'exprimer toute sorte de requête analytique. En outre, nous avons illustré, à travers un exemple, qu'elle assure la traçabilité des besoins à travers les relations de correspondance entre les éléments du modèle de besoins et le schéma multidimensionnel.

Ce travail est un maillon dans la chaîne de conception automatisée des entrepôts de données selon la démarche MDA. Ses perspectives se situent à plusieurs niveaux. En effet, étant donnés un ensemble d'exigences analytiques spécifiées conformément à notre modèle, nous envisageons automatiser la génération de schémas de magasins de données. Dans une étape ultérieure, nous comptons les valider par rapport aux sources de données.

Références

Adamson, C. (2006). *Mastering data warehouse aggregates solutions for star schema performance*. Indianapolis: Wiley Publishing, Inc.

Cabibbo, L. et R.Torlone (1998). A logical Approach to Multidimensional Databases. *EDBT*, 183-197.

Codd, E.F (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 377–387.

Feki, J., A.Nabli, H.Ben-Abdallah, et F.Gargouri (2008). An Automatic Data Warehouse Conceptual Design Approach, in 2nd edn of Encyclopedia of Data Warehousing and Mining, John Wang Edition.

Giorgini, P., S.Rizzi, et M.Garzetti (2007). GRAnd: A goal-oriented approach to requirement analysis in data warehouses. *Journal of decision Support Systems*, 45:4-21.

Giorgini, P., S. Rizzi, et M. Garzetti (2005). Goal-oriented requirement analysis for data warehouse design, DOLAP, 47-56.

Golfarelli, M., et S.Rizzi (1998). A Methodological framework for Data Warehouse Design. DOLAP.

Hüsemann, B., J.Lechtenbörger, et G.Vossen (2000). Conceptual Data Warehouse Design. *DMDW*, 6.1-6.11.

Mard, M., R.R.Dunne, E.obsborne et J.S.Rigby (2004). *Driving your company's value: strategic benchmarking for value.* New Jersey: Wiley Publishing, Inc.

Kimball, R. (1996). *The Data Warehouse Toolkit*. Indianapolis: Wiley Publishing, Inc.

List, B., J.Scheifer, et A.M. Tjoa (2000). Process-oriented requirement analysis supporting the data warehouse design process a use case driven approach. Dexa, LNCS 1873, 593-

603.

Bruckner, R., B.List, et J.Scheifer (2001). Developing Requirements for Data Warehouse Systems with use cases. Sventh Americas conference on information Systems.

MDA (Model Driven Architecture 1.0.1) (2003). http://www.omg.org/mda/.

Moody, L. D. et A.R.K.Mark (2000). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouses and Data Mart Design. *DMDW*, 5.1-5.12.

Paim, F.R.S. et J.B. Castro (2003). DWARF : an approach for requirements definition and management of data warehouse systems, *RE*,75-.

Prakash, N. et A. Gosain (2008). An approach to engineering the requirements of data warehouses. *Journal of Requirements Eng*, 13:49-72.

Prakash, N., Y.Singh, et A.Gosain (2004). Informational Scenarios for Data Warehouse Requirements Elicitation, *ER*, LNCS 3288, 205-216.

Schiefer, J., B.List, et R.M.Bruckner (2002). Aholistic approach for managing requirements of data warehouse systems, Proceedings of Americas Conference on Information Systems.

Spanoudakis, G., A. Zisman, E.Pérez-Minana, et P.Krause (2004). Rule-based generation of requirements traceability relations. *Journal of systems and software*, 72:105-127.

Thomson, E. (2002). OLAP solutions : building multidimensionnel information systems: 2nd edn. New York: Wiley.

UML (Unified Modeling Language 2.1.2) (2007). http://www.omg.org/spec/UML/2.1.2/.

Winter, R. et B. Strauch (2003). A method for demand-driven Information Requirements Analysis in Data Warehousing Projetcs. *HICSS*, 231.1.

Winter, R. et B. Strauch (2004). Information Requirements engineering for data warehouse systems. SAC, 1359-1365.

Summary

In this paper, we are interested in DSS requirements engineering. More precisely, we address the problem of expressing these requirements in natural language. In order to limit the emergent difficulties during the steps of i) specification of OLAP requirements in natural language, ii) their validation and iii) the automatic treatment of these needs, we propose a model and a simplified grammar, developed through an empirical study.

Un méta modèle multidimensionnel générique pour la conception des entrepôts de données

Nouria Harbi, Fadila Bentayeb, Marie Jeanne Meuke Fante, Omar Boussaid

Laboratoire ERIC, Université Lumière – Lyon 2
5 avenue Pierre Mendès-France
69676 Bron Cedex
{nouria.harbi, omar.boussaid}@univ-lyon2.fr,
bentayeb@eric.univ-lyon2.fr, meukefante@yahoo.fr

Résumé. La modélisation multidimensionnelle de données est souvent liée à l'implémentation. Ainsi le modèle conceptuel est souvent confondu avec le modèle logique. C'est le cas du schéma en étoile, qui depuis toujours est perçu comme un modèle conceptuel pour exprimer les besoins d'analyse et formuler en même temps dans des termes de modèle logique. Cette confusion biaise le discours de l'analyste et par conséquent met en évidence l'absence d'un modèle conceptuel multidimensionnel standard reconnu par tous Rizzi et al. (2006).
Actuellement, bien qu'aucun modèle n'est reconnu comme standard ; il se dégage néanmoins plusieurs concepts qui sont admis par la communauté des chercheurs et des industriels comme étant les bases d'une modélisation multidimensionnelle : cube de données, dimension, fait, hiérarchie, paramètre (membre), et attribut Golfarelli et al. (1998). L'ensemble de ces concepts possède des propriétés qui constituent une base pour la modélisation conceptuelle multidimensionnelle.
L'objectif de cet article est de présenter les limites des modèles d'entrepôts de données classiques qui ne prennent pas en compte les propriétés qu'un modèle multidimensionnel doit satisfaire, et de proposer un méta modèle (M^3Gen) conceptuel multidimensionnel générique qui peut prendre en compte les spécificités des données, éloigné des contraintes physiques liées à l'implémentation et ceci à l'aide d'un socle minimum de propriétés. Pour valider notre méta modèle, nous avons implémenté un prototype sous oracle 11g afin de générer de façon interactive tout modèle d'entrepôt de données.

1 Introduction

A l'heure actuelle, il est largement accepté par tous que la modélisation de référence pour les systèmes d'information décisionnels soit la modélisation multidimensionnelle Inmo (1996) Jarke et al. (2001). Les systèmes d'information décisionnels ont pour objectif d'assurer le support du processus de prise de décision. Autrement dit, ils doivent permettre l'analyse en ligne des données (OLAP**).** Historiquement, la préparation des données à l'analyse avait pour préalable d'extraire des données à partir des sources de données qui étaient en général relationnelles. Les modèles multidimensionnels proposés correspondaient à des modèles logiques basés sur des concepts relationnels (relation, clefs primaires, clefs étrangères…). D'autre part, au vue de l'importance de la volumétrie des données, la dénormalisation était tolérée afin de satisfaire des contraintes de performances. Dans la plus part des cas, la modélisation des applications multidimensionnelles est liée à l'implantation Torlone (2003). Cette démarche montre l'importance accordée aux caractéristiques de l'information manipulée afin d'avoir des temps de réponse acceptables par les décideurs. Ainsi le modèle conceptuel était confon-

du avec le modèle logique. C'est le cas du schéma en étoile par exemple, qui depuis toujours est perçu comme un modèle conceptuel pour exprimer les besoins d'analyse et formuler en même temps dans des termes de modèle logique.
L'avènement du décisionnel date maintenant de deux décennies et depuis, de nombreux travaux, sur la modélisation conceptuelle multidimensionnelle existent dans la littérature. Toutefois, aucun consensus n'a pu se dégager sur un modèle conceptuel multidimensionnel standard. Cependant, différentes propriétés concernant ce modèle émergent de différentes propositions déjà avancées dans ce domaine. Pour définir un modèle multidimensionnel qui représente uniquement les concepts du monde réel indépendamment de toute implantation physique et de tout formalisme logique, différentes listes de propriétés ont été proposées dans Vassiliadis et al. (1999), Blaschka et al. (1999), Abelló et al. (2001), Rafanelli (2003), Lujan-Mora (2005), Annoni (2007).
L'objectif de ce papier est de réunir l'ensemble de ces propriétés que nous avons recensé dans la littérature pour servir comme base de contraintes devant être satisfaite pour toute proposition de modèle conceptuel multidimensionnel. Ainsi, Nous proposons un Méta Modèle Multidimensionnel Générique (M^3Gen) permettant de générer d'une façon interactive tout modèle multidimensionnel selon les propriétés énoncées.
Cet article est organisé de la façon suivante. Nous présentons d'abord les notions et définitions de base de la modélisation multidimensionnelle en section 2, ensuite un panorama des différentes approches de conception des modèles multidimensionnels classiques et de leurs limites est exposé dans la section 3. Puis, dans la section 4 nous décrirons à travers un exemple les propriétés issues des modèles étudiés. Nous proposons ensuite dans la section 5 une nouvelle classification de ces propriétés. Dans la section 6 nous verrons la présentation, l'élaboration, l'instanciation et l'implémentation du méta modèle multidimensionnel générique (M^3Gen) qui prend en compte certaines propriétés exposées en section 4. Nous terminerons par une conclusion et des perspectives dans la section 7.

2 Modélisation multidimensionnelle

La modélisation multidimensionnelle des données vise à représenter les données en fonction de l'analyse prévue par les décideurs Chrisment et al. (2005). L'entrepôt de donnée est une structure informatique dans laquelle est centralisé un volume important de données consolidées à partir des différentes sources de renseignements d'une entreprise (notamment les bases de données internes). L'organisation des données est conçue pour que les personnes intéressées aient accès rapidement et sous forme synthétique à l'information stratégique dont elles ont besoin pour la prise de décision.
.

2.1 Exemple introductif

Dans cette section, nous présentons un entrepôt de données (FIG1) qui permet aux décideurs d'observer et d'analyser deux domaines d'activités : la production et la vente à travers des indicateurs : quantité produite, quantité vendue, bénéfices et montant, selon plusieurs axes : période, vendeur, client, lieu-magasin, produit, usine.

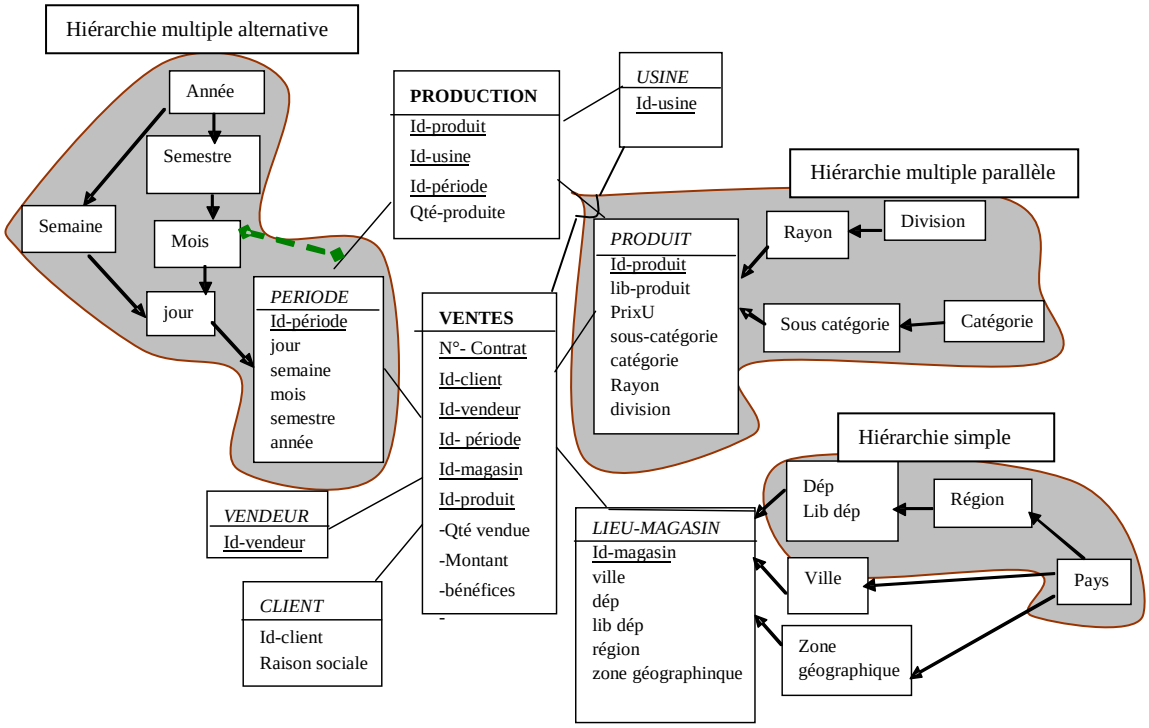


FIG 1 *– Entrepôts de données VENTES.*

Nous constatons que les nombreuses propositions sur la modélisation conceptuelle multidimensionnelle sont qu'elles ne reposent pas sur des bases théoriques standard Rizzi et al. (2006). Actuellement, bien qu'aucun modèle multidimensionnel n'est reconnu comme standard il se dégage néanmoins trois concepts qui sont admis par la communauté des chercheurs et des industriels comme étant les bases d'une modélisation multidimensionnelle, à savoir le concept de «cube de données», mais aussi ceux de «dimension» et de «fait». Nous définissons ces concepts comme suit:

- *Fait* : Un fait représente un sujet d'analyse, appelé aussi donnée factuelle ou centre d'intérêt sur lequel porte l'analyse. Il est exprimé en fonction d'indicateurs appelés mesures et d'axes d'observation appelés dimensions.
- *Dimension* : Une dimension représente un axe suivant lequel l'analyse des données est faite. Elle se compose de descripteurs des faits et peut être organisées sous forme d'hiérarchies.
- *Mesure* : Elle représente un indicateur de performance de l'activité à analyser. C'est la propriété qui caractérise un fait. Généralement c'est une donnée numérique, additive permettant des opérations d'agrégation.
- *Cube de données* : Un cube de données représente une structure multidimensionnelle dont les cellules contiennent des données agrégées (*mesures*) et dont les arêtes (*dimensions*) contiennent les axes d'analyse naturels des données Kimball et al. (1998).

Les premiers modèles proposés sont basés sur des structures (*Cube*) ne représentant pas tous les concepts exprimés par les utilisateurs mais répondant aux caractéristiques de l'implantation. Pour définir un modèle multidimensionnel qui représente les concepts du monde réel indépendamment de tout aspect physique, les applications d'analyse multidimen-

sionnelle requièrent des concepts qui se rapprochent de la vision des données par les décideurs et de la sémantique du décisionnel. Les concepts introduits sont : les différents types de schémas en étoile, en flocon de neige et en constellation Kimball (1996), les hiérarchies, les paramètres, les attributs faibles, Golfarelli et al. (1998).
- *Hiérarchie* : Elle définit plusieurs paliers d'observations des faits pour une dimension donnée. Cette dernière contient des paramètres et/ou des attributs faibles (voir plus loin). Une hiérarchie est composée de sous ensemble de paramètres organisés en plusieurs niveaux représentant des granularités différentes.
Exemple : Les paramètres *Jour*, *Mois Semestre* et *Année* sont organisés en hiérarchie. C'est à dire un ensemble de paliers d'observation représentant des niveaux de granularité différents.
- *Paramètre* (membre) : C'est un membre (propriété d'une dimension) qui représente également un descripteur de faits. Celui-ci a la particularité de représenter également un descripteur pouvant définir un niveau de granularité de la hiérarchie.
Exemple : Les paramètres *Ville*, *Département*, *Pays* de la dimension «*LIEU-MAGASIN*».
- *Attribut faible* : est un membre d'une dimension ou d'une hiérarchie qui ne peut pas être transformé en niveau hiérarchique.
Exemple : Le libellé du département (*Lib Dép*) décrit le paramètre *Département.*
- *Attribut forte* : C'est un membre (propriété d'une dimension) qui représente également un descripteur de faits. Celui-ci a la particularité de représenter également un descripteur pouvant définir un niveau de granularité de la hiérarchie.
- *Schéma en étoile* : est une représentation multidimensionnelle des données proposée par Kimball (1996) qui représente les faits au centre et les dimensions qui rayonnent autour des faits. Cette représentation est un standard reconnu.
- *Schéma en flocon de neige* : est un schéma en étoile dont certaines (ou toutes) dimensions sont organisées en hiérarchie.
Exemple : les dimensions «*PERIODE*», «*LIEU-MAGASIN*» organisées en hiérarchie permettent une représentation en flocon de neige (FIG 1)
- *Constellation (schéma en constellation)* : est un regroupement de schémas en étoile qui partagent des dimensions. Il met en avant la corrélation entre les faits et il évite de définir plusieurs fois la même dimension au sein d'une même organisation.
L'ensemble de ces concepts qui sont spécifiques aux analyses multidimensionnelles ont des propriétés liées Annoni (2007) et Trujillo (2002).

3 Etat de l'art

Il existe de nombreuses recherches menées depuis la fin des années 90 au sujet de la modélisation multidimensionnelle de données. Le modèle conceptuel qui offre la possibilité de représenter des constructions sémantiques dans le but de «capturer» les objectifs d'analyse des utilisateurs, les différents formalismes (niveau intermédiaire entre le niveau conceptuel et niveau logique) qui offrent moins de constructions conceptuelles, le modèle logique qui est le deuxième niveau fondamental de toute modélisation.
Dans cette section, nous nous intéresserons à la modélisation conceptuelle et à la modélisation logique des données multidimensionnelles. D'une façon générale l'évolution de la modélisation multidimensionnelle montre qu'au fil des années, on tend de plus en plus à favoriser une modélisation conceptuelle orientée vers la capture des sémantiques mais il n'existe toujours pas de consensus ni de standard pour représenter ces modèles. Nous décrirons brève-

ment les différents modèles conceptuels, logiques et autres formalismes présentés dans la littérature.

Au niveau des modèles logiques, le travail de recherche le plus conséquent est celui proposé par Kimball (1996). Il décrit l'implémentation du modèle multidimensionnel sous forme de base de données relationnelles ne pouvant pas être considéré comme un modèle physique car il reste indépendant des outils. Dans le même esprit Moody (2000) propose une méthodologie basée sur un modèle Entité-Association(E-A). Différentes sortes de schémas sont obtenus (« star », « constellation », « galaxy »etc. …). Tous ces schémas contiennent des tables relationnelles et sont basés sur la dualité faits-dimensions.

Au niveau conceptuel, Golfarelli, Maio et al. (1998) proposent un modèle conceptuel pour les entrepôts de données construit à partir des schémas opérationnels (soit E/A soit relationnel). Trujillo (1998, 2002) quant à lui propose un modèle conceptuel orienté objet basé sur le langage UML et présenté sous forme de requêtes. Sapia (1999) part sur une spécialisation du modèle E/A qui représente des sémantiques Multidimensionnelles. Dans la même optique Abello et al. (2002, 2006) proposent un modèle multidimensionnel basé sur la spécialisation du méta-modèle du diagramme de classes UML. Pour Sánchez et al. (1999) ce qui est important est de présenter un modèle conceptuel multidimensionnel permettant de représenter des bases de données multidimensionnelles indépendamment des outils spécifiques utilisés pour leur implémentation, le schéma multidimensionnel est défini selon des domaines (domaine des dimensions, domaine de synthèse et domaine de description), une hiérarchie est alors une collection de domaines d'agrégations. Un schéma de fait est une collection d'attributs de dimensions, collection de dimension. Tryfona et al. (1999) proposent un modèle multidimensionnel starER qui a pour concepts de base des faits, des entités, des relations et des attributs.

Par ailleurs d'autres auteurs proposent des formalismes différents pour la modélisation multidimensionnelle des données comme Hypercube pour Agrawal et al. (1996) qui utilise des fonctions pour introduire les hiérarchies. Gyssens et al. (1997) proposent des Tables Dimensionnelles comme formalisme et utilisent aussi des fonctions pour introduire les hiérarchies. Contrairement à eux Thomas et al. (1997), Cabibbo et al. (1998) utilisent l'ordre partiel entre les attributs pour introduire les hiérarchies et proposent le Cube et F-table comme formalisme. Seuls Li et Wang. (1996) utilisent des relations pour l'introduction des hiérarchies et proposent le Cube multidimensionnel comme formalisme.
Pour synthétiser les différentes approches en terme de modélisation multidimensionnelle, nous dirons que les modèles reposent sur des paradigmes qui peuvent être le paradigme entité-association, le paradigme objet ou des paradigmes totalement nouveaux et spécifiques tels que la combinaison de deux paradigmes. A la suite de cette étude bibliographique il en ressort qu'aucun modèle n'est individuellement complet. Par exemple : Tryfona et al. (1999) travaille sur des concepts de base tel que fait, dimension, hiérarchie non stricte, hiérarchie multiple alternative. Les modèles comme celui de Prat et Akoka. (2002) ne prennent pas en compte la propriété des faits multiples.
De ce fait nous établissons un tableau (Tab1) comparatif des différentes approches selon plusieurs critères que nous avons identifié à savoir : type d'approche (descendant ; ascendant et mixte) et le niveau d'abstraction (conceptuel, logique et formalisme)

Approche & Niveaux d'abstraction / Auteurs (Modèle)	Approche	Niveau Conceptuel	Niveau Logique	Niveau inter-médiaire (Formalisme)
				x
				x
Li and Wang(MDD) 1996	Descendant		Objet	
Agrawal, Gupta et sarawagi 1997		Spécifique		
Kimball 1997	Ascendant	Spécifique		
Cabibbo and Torlone(MD) 1998	Ascendant	Spécifique		
Golfarelli et al. (DFM) 1998		E-A		
Sapia et al. (MERM) 1999		E-A		
Tryfona et al. (starER) 1999		E-A		
Sanchez et al. (IDEA) 1999		Objet		
Abello et al. (2000,2006)				Spécifique
Pedersen(EMDM)2000	Ascendant		E-A	
Moody et Kortink 2000	Ascendant	Spécifique		
Hüsemann et al. 2000	Mixte	Objet		
Bonifati et al.2001	Descendant	Objet		
Prat et akoka 2002	Mixte	Objet		
Trujillo et al. (GOLD) 2002		E-A		
Phipps et Davis 2002				

Tab1 *: Tableau comparatif des approches de modélisation et de données multidimensionnelles.*

Cette étude bibliographique et cette comparaison nous amènent à nous interroger sur la proposition d'un méta modèle générique qui pourrait prendre en compte toutes les propriétés qui existent sur la modélisation multidimensionnelle et les concepts de base. Pour cela il nous semble plus judicieux de recenser ces propriétés.

4 Les propriétés de la modélisation multidimensionnelle

Nous présentons ici, 14 propriétés nécessaires à la modélisation multidimensionnelle des données Anonni (2007) liées à : (1) la structuration des dimensions afin d'exprimer les liens qui existent entre les paramètres des dimensions ; (2) la structuration des faits et des mesures, car les faits peuvent être liés. Un fait peut être composé de plusieurs mesures et une mesure peut dériver d'une ou plusieurs autres mesures ; (3) la cohérence de l'interrogation des données car une requête peut être construite à partir du résultat d'une précédente avec des opérations d'augmentation ou de réduction du niveau de détail des données suivant la hiérarchie ; (4) le traitement symétrique des faits et des dimensions au cours des manipulations des données. Cette propriété est liée aux algèbres et aux langages de définition et de manipulation des données.

Pour illustrer les propriétés ci-dessous, nous utilisons l'entrepôt de données VENTES (FIG1).

4.1 Propriétés liées à la structure des dimensions

Propriété 1 : Les hiérarchies simples d'une dimension : elles explicitent le chemin entre les niveaux d'une dimension. Exemple : Fig 1 Les paramètres *Ville*, *Département*, *Pays* de la dimension «*LIEU-MAGASIN*» c'est une hiérarchie simple à trois niveaux.
Cette hiérarchie est appelée également hiérarchie stricte si elle présente une relation 1-N entre ces niveaux : un département appartient à une seule région et une région appartient à un seul pays. D'autre part, un pays a plusieurs régions ; une région a plusieurs départements.
Propriété 2 : Les hiérarchies multiples (ou parallèles) et alternatives : elles partagent plusieurs niveaux d'une dimension.
Si une dimension a au moins deux hiérarchies simples dont les niveaux fins ne sont pas les mêmes, elle possède une hiérarchie multiple (parallèle). Si les niveaux fins sont les mêmes, ces hiérarchies sont dites alternatives. Exemple *:* Fig1 Les paramètres *Jour*, *Mois Semestre* et *Année* sont organisés en hiérarchie multiple alternative
Propriété 3 : Les hiérarchies non strictes : elles sont définies quand une instance d'un niveau fin peut appartenir à plusieurs niveaux supérieurs associés. Exemples : 1 jour (quantième) peut appartenir à plusieurs mois.
Il s'agit d'une dimension qui partage entre plusieurs niveaux les mêmes membres. C'est une relation N-N entre membres.
Propriété 4 *:* Les éléments terminaux multiples : ils indiquent qu'une dimension peut être reliée aux faits par plus d'un de ces paramètres (éléments).
Ce concept est important car tous les faits d'un schéma en constellation ne s'expriment pas en fonction du même niveau de granularité d'une dimension.
Exemple : le fait «*VENTES*» est au niveau granularité *Jour* alors que le fait «*PRODUCTION*» est au niveau granularité *Mois*. Concept est représenté graphiquement par ◆– –◆
Propriété 5 **:** Les rôles d'une dimension : ils permettent de représenter qu'un fait peut s'exprimer plusieurs fois en fonction d'une même dimension.
Ces rôles sont pertinents car dans de nombreux projets une dimension intervient plus d'une fois. La conception qui en résulte est la duplication de la dimension selon les différents rôles. C'est un abus de conception car c'est le même concept qui participe à la relation avec des rôles différents. Exemple la dimension « USINE » intervient deux fois : une fois en tant que « USINE DE PRODUCTION » et une fois en tant que « USINE, VENTE USINE).
Propriété 6 **:** Une dimension dégénérée : elle désigne un attribut d'un fait qui permet de l'identifier de manière unique et qui n'est pas une donnée qui caractérise le fait. Autrement dit l'attribut n'est pas une mesure. Il agit comme une clef seulement.
Exemple : Dans le fait «*VENTES*», le *N° de contrat* identifie de manière unique le fait «*VENTES*». Alors que le fait peut aussi être identifiable de manière unique par les clefs étrangères des dimensions en fonction des quelles s'exprime une vente : «*PRODUIT*», «*VENDEUR*», «*LIEU-MAGASIN*», «*CLIENT*», «*PERIODE*».

4.2 Propriétés liées à la structure des faits et des mesures

Propriété 7 **:** Les dimensions multiples : elles caractérisent un fait composé de plusieurs dimensions.
Exemple : le fait «*VENTES*» est composé des dimensions : «*PRODUIT*», «*VENDEUR*», «*LIEU-MAGASIN*», «*CLIENT*», «*PERIODE*».
Propriété 8 **:** Les faits multiples : ils constituent le schéma en constellation. Ils permettent de représenter plusieurs domaines d'activité d'un système décisionnel complexe.

Exemple : Dans FIG. 1 les faits : «*VENTES*» et «*PRODUCTION*»
***Propriété 9* :** Les mesures dérivées : elles définissent les mesures obtenues par calcul à partir d'une ou plusieurs mesures d'un fait et d'autres données. Ce calcul peut être arithmétique une fonction analytique ou une fonction d'agrégation
Exemple : Dans le domaine «*VENTES*», la mesure *Chiffre d'affaire* peut être déduite par un calcul arithmétique simple à partir de la mesure *Qté vendue* et une donnée source qui est le prix unitaire *PrixU.*
***Propriété 10* :** Les liens entre mesures : Ils explicitent les règles de calcul des mesures dérivées. C'est une des tâches des décideurs pour préciser les causes et les tendances. C'est aussi un moyen pour contrôler la fiabilité des données. Le concepteur déroule tout le calcul à l'origine d'une mesure. La connaissance des liens entre les mesures permet de procéder à cette validation plus efficace et par la même occasion connaître les mesures concernées par la modification d'une mesure donnée.
Exemple : Evaluation des commissions des vendeurs, Calcul des chiffres d'affaires par produit par client, par vendeur…,
***Propriété 11* :** Un fait dégénéré : il est une mesure telle qu'à une valeur de celle-ci sont associées plusieurs valeurs d'une dimension pour un même rôle
Exemple : Dans le cas où une vente est réalisée par plusieurs vendeurs, la mesure *Commission* du fait «*VENTES*» est transformée en fait dégénéré associé au lien entre le fait «*VENTES*» et la dimension «*VENDEUR*».

4.3 Cohérence de l'interrogation

La cohérence de l'interrogation est liée à la fiabilité des données. En effet, une requête construite via des opérations de manipulation sur les données suivant une hiérarchie permet de réduire ou d'augmenter le niveau de détail des données. Elle se caractérise par la pertinence des agrégations.
***Propriété 12* :** La pertinence des agrégations : elle définit les fonctions d'agrégation valides pour le passage d'un niveau de détails *N* à un niveau moins détaillé *M* pour une hiérarchie d'une dimension. Cette propriété indique les consolidations des mesures qui ont un sens pour les décideurs Husemann et al. (2000).
Exemple : La fonction d'agrégation généralement utilisée par défaut est la somme. Si dans un cas la fonction somme n'a pas de sens pour une mesure, la fiabilité des données obtenues n'es pas garantie.

4.4 Aspect dynamique

La dynamique s'évalue en fonction des traitements que le modèle permet de représenter. Par ailleurs, les traitements du système d'information décisionnel se rapportent à la dérivation des données et à la préparation des données
Propriété 13 : La dérivation des données : elle regroupe les traitements d'extraction, de chargement, de suivi et de sécurité des données.
***Propriété 14* :** La préparation des données : elle regroupe les traitements liés à la validité, l'historisation, le rafraîchissement, l'archivage, le calcul et la consolidation des données. Autrement dit ceux liés à la caractérisation des données pour la prise de décision.

La liste des propriétés constitue un recueil prometteur pour la modélisation conceptuelle multidimensionnelle. Cette liste n'est pas exhaustive, elle peut servir de base de départ. Elle est

appelée à être étendue de par son application aux données complexes avec leur spécificité et surtout pour prendre en compte les aspects qualité, sécurité, et dynamique des phases du processus d'entreposage.
Nous proposons une nouvelle classification de ces propriétés pour dégager les propriétés qui nous servirons pour la phase de modélisation multidimensionnelle du processus d'entreposage.

5 Classification des propriétés

Nous constatons que parmi les propriétés décrites dans la section 4, certaines propriétés sont fondamentales et utiles pour la phase de modélisation. D'autres sont plus nécessaire dans la phase ETL (Extraction, Transformation et Chargement) et d'autres encore concernent la phase d'analyse OLAP. Nous suggérons une nouvelle classification (Tab2) plus appropriée pour notre travail, basée alors selon la phase ETL, la phase conception et modélisation et la phase analyse du processus d'entreposage de données.

Phase du processus d'entreposage	Propriétés
ETL	Dérivation des données (p13), Préparation des données (p14)
Modélisation multidimensionnelle	Domaine(s)(p8), Rôle d'une dimension(p5), Dimension multiple(p7), Eléments terminaux multiples (p4) ,Hiérarchie simples(p1), Hiérarchie multiple, alternative(p2), Hiérarchie non stricte (p3), Mesures dérivées (p9), Liens entre mesures (p10)
Analyse OLAP	Pertinence des agrégations (p12)

Tab 2 : Nouvelle classification des propriétés

Nous constatons d'une part que**,** les propriétés qui se rapportent à la dérivation des données et à la préparation des données, autrement dit celles liées aux traitements et caractérisation des données pour le système décisionnel, correspondent plutôt à la partie ETL (Extraction, Transformation et Changement) du processus décisionnel, à savoir :
Propriété 13 : La dérivation des données : « elle regroupe les traitements d'extraction, de chargement, de suivi et de sécurité des données (notamment garantir l'intégrité des données) », et
Propriété 14 **:** La préparation des données : « elle regroupe les traitements liés à la validité, l'historisation, le rafraîchissement, l'archivage, le calcul et la consolidation des données ».
D'autre part, vu que les possibilités d'analyse dépendent du schéma de l'entrepôt de données et plus particulièrement, des dimensions et de leur(s) hiérarchie(s) implique que la navigation dans les données est conditionnée par cette organisation dimensionnelle des données. Cette navigation se base entre autre sur l'agrégation des données. Donc la propriété liée à la fiabilité des données, qui assure la cohérence de l'interrogation, à savoir :
Propriété 12 : La pertinence des agrégations : « elle définit les fonctions d'agrégation valides pour le passage d'un niveau de détails *N* à un niveau moins détaillé *M* pour une hiérarchie d'une dimension ». Cette propriété indique les consolidations des mesures qui ont un sens pour les décideurs Husemann et al. (2000). Elle peut être classée dans la partie analyse du système décisionnel. Celle-ci est soutenue par le concept de hiérarchie. En effet, dans les entrepôts de données, les hiérarchies vont permettre de représenter comment doivent être agrégées les données. La hiérarchisation des données dans les modèles multidimensionnels

permet des analyses à différents niveaux de détail. Classiquement, les hiérarchies sont représentées par des concepts qui sont relies par des relations un à plusieurs.
Différents types de hiérarchies soulèvent encore des problèmes de représentation. Les propriétés relatives aux hiérarchies citées ci-dessus permettent de représenter quelques cas particuliers. On a donc pu dégager les propriétés de la phase de modélisation multidimensionnelle qui nous permettrons de concevoir notre méta modèle générique.

6 Méta Modèle Multidimensionnel générique M^{3}Gen

Plusieurs méta modèles on été proposés dans la littérature, mais il n'existe toujours aucun standard, ni consensus permettant de concevoir un entrepôt de donnée. Les schémas en étoiles, constellation, flocon de neige ont toujours été considérés comme des modèles logiques ou physiques. D'après plusieurs publications récentes, le méta modèle multidimensionnel est considéré au niveau logique ou même au niveau conceptuel, Nicolas Prat et al. (2006). Ceci met en évidence l'absence d'un modèle standard. Néanmoins l'analyse de l'évolution des modèles multidimensionnels tend à montrer que les modèles les plus récents favorisent l'aspect conceptuel. Ce qui nous amène à imaginer peu à peu l'émergence d'un méta modèle standard permettant de valider toute conception de modélisation multidimensionnelle quelque soit les propriétés intrinsèques.

6.1 Présentation du M^{3}Gen

Nous définissons le méta modèle M^{3}Gen comme étant la combinaison des concepts de base et des propriétés qui leurs sont liées. C'est un méta modèle conceptuel orienté objet et basé sur le langage UML qui est la clé de toute conception orienté objet. Tout l'intérêt de ce méta modèle est d'apporter une valeur ajoutée à toutes les approches qui existent jusqu'à présent sur la modélisation multidimensionnelle. L'un des apports principal est que M^{3}Gen permet de générer tout modèle conceptuel d'entrepôt de donnée. Dans la littérature on retrouve plusieurs approches permettant de concevoir un entrepôt de donnée, on retiendra précisément les deux approches suivantes qui sont dans le même esprit que la notre à savoir : Abello et al. (2002, 2006) qui proposent un modèle multidimensionnel basé sur la spécialisation du méta modèle du diagramme de classes UML et celle de Nicolas Prat et al. (2006) qui proposent un méta modèle utilisant une combinaison d'outil Entité association et UML. Ce méta modèle part des exigences des utilisateurs et ensuite modélisé par un formalisme conceptuel dépourvu de tous concepts multidimensionnels. Ces concepts multidimensionnels sont intégrés dans le modèle logique. Notre position par rapport aux différents méta modèles qui existent, est tout simplement qu'aucun de ces méta modèles ne prend en compte l'ensemble des propriétés que doit respecter toute conception d'entrepôt de données et que nous suggérons de le faire à travers le méta modèle M^{3}Gen. Nous avons choisit UML comme outil de modélisation pour la richesse sémantique qu'offre les différents diagrammes (classe), pour sa variété (agrégation, généralisation etc. …), ces variétés sont utilisées pour la définition des hiérarchies multidimensionnelles dans la phase logique. Le M^{3}Gen est un outil méthodique d'analyse et de conception qui permet

1°) de représenter les concepts de base standards nécessaires pour la modélisation multidimensionnelle (Faits, mesures, dimensions …) basés sur les propriétés liées à la conception que nous avons sélectionné parmi l'ensemble des propriétés étudiées ci-dessus [section 4].

2°) de générer automatiquement tout modèle d'entrepôt de données par instanciation

Les concepts retenus pour cette représentation sont :
- Domaine(s) : domaine d'activité du système décisionnel, exemples : Vente, Production.
- Dimension(s) avec un rôle (R).
- Hiérarchie(s) de type : hiérarchie simple (non stricte, stricte) hiérarchie multiple (alternative ou parallèle).
- Membre(s) pouvant être de type : attribut-Fort qui peut devenir une hiérarchie ou attribut-Faible qui ne devient pas une hiérarchie.

Nous remarquons donc que parmi les concepts retenus pour la modélisation multidimensionnelle, pour nous une dimension est tout au autant rattachée à un domaine qu'un fait. Ce méta modèle conceptuel inclut les entités importantes et les relations existantes entre elle, on a choisit de représenter certains attributs pour que le méta modèle soit compréhensible. Le M3Gen permet donc la structuration des besoins utilisateurs. Comme dans tout modèle conceptuel on note l'absence de clé étrangère.

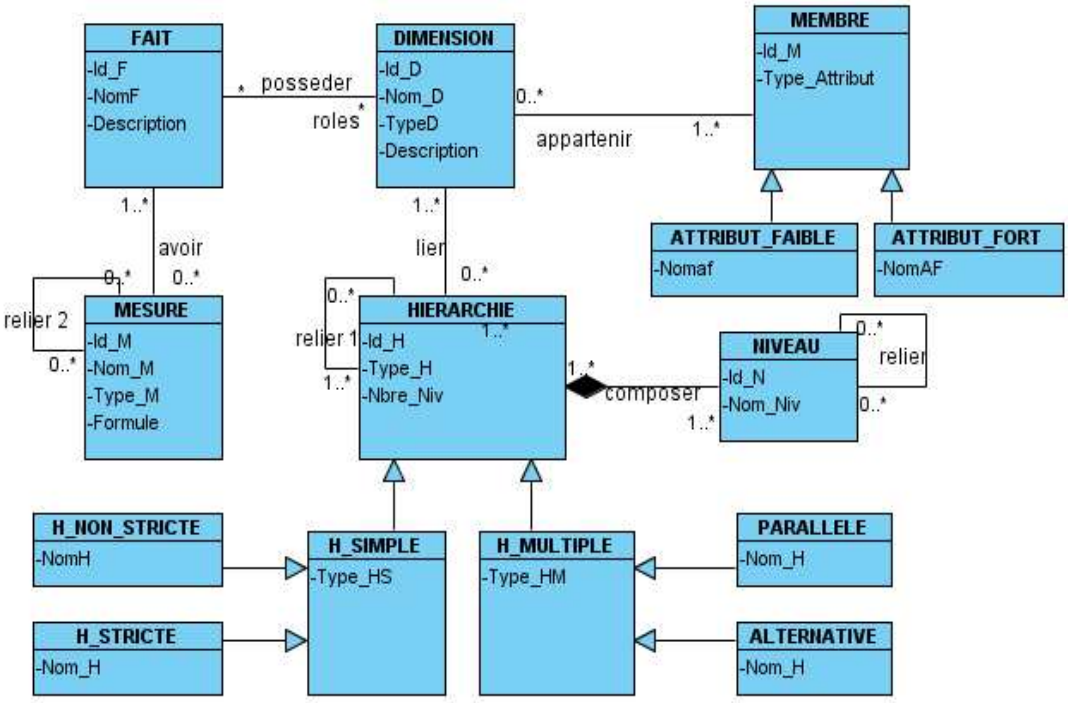


FIG 2 : Méta modèle multidimensionnel générique pour les entrepôts de données M³Gen

6.2 Elaboration de M³Gen

Nous dégageons 12 règles permettant d'élaborer M³Gen. Ces règles permettent de faciliter l'implémentation de n'importe quel entrepôt de donnée. On définit alors les différents faits à modéliser, les dimensions et leurs membres, les mesures du domaine (Fait), les hiérarchies liées aux dimensions, et les niveaux de ces hiérarchies. Tous ceci en respectant les cardinalités.

Règle 1 : Une dimension peut avoir au minimum 0 hiérarchies à plusieurs hiérarchies,
Règle 2 : Une hiérarchie peut être spécialisée en 3 grandes familles,

- Hiérarchie simple ***(propriété 1)*** qui peut être composée d'une hiérarchie stricte et non strite (Propriété 3).
- Hiérarchie multiple, composée d'hiérarchie parallèle ou hiérarchie alternative ***(Propriété 2)***

Règle 3 : Une dimension possède au minimum 0 membre (ou paramètre) à plusieurs membres
Règle 4 : Un membre d'une dimension peut être relié à plusieurs faits par des attributs forts terminaux différents, car les domaines d'observation peuvent avoir des niveaux de granularité différents ***(Propriété 4)***
Règle 5 : Un membre peut posséder un attribut fort, c'est-à-dire qui peut se transformer en hiérarchie ou un attribut faible ne pouvant pas se transformer en hiérarchie,
Règle 6 : Cette relation (* ...*) entre le fait et la dimension donne naissance un rôle de la dimension ***(Propriété 5).***
Règle 7 :

- Un fait possède minimum 1 à plusieurs dimensions ***(Propriété 7)***
- Une dimension appartient au minimum à un fait et à plusieurs fait au maximum avec des rôles éventuellement différents

Règle 8 :

- dans un système décisionnel, on retrouve plusieurs domaines d'activité ***(Propriété 8)***
- Un fait possède une mesure au minimum à plusieurs mesures,
- Une mesure appartient à un fait au minimum ou à plusieurs domaines (faits),

Règle 9 :

- Une mesure peut être obtenue par calcul à partir d'autres mesures et d'autres données, elles peuvent être relié entre elles : attribut numérique de l'entité mesure ***(Propriété 9)***

Règle 10 : Une hiérarchie est composée d'un minimum de 2 niveaux et maximum plusieurs
Règle 11 : Un niveau appartient à 0 hiérarchie minimum et plusieurs hiérarchies maximum
Règle 12 : des hiérarchies peuvent être reliées entre eux (hiérarchie multiple alternative)
La liste de toutes les règles de calcul concernant les mesures calculées de l'entité mesure représente le lien qui existe entre les mesures ***(Propriété 10)***
Ces 12 règles correspondent chacune à des propriétés citez ci-dessus en section 2
D'après la représentation du **M^{3}Gen** on constate que certaines propriétés ne nécessitent pas une représentation particulière car ce sont des propriétés dérivées d'autres propriétés qui ont été représentée, à savoir :
Propriété 6 : Dimension dégénérée
Les concepts fait et dimension on été représentés ce qui est suffisant pour la structuration des besoins d'analyse. Et par conséquent la dimension dégénérée ne nécessite pas une représentation propre.
Propriété 11 : Les faits dégénérés
Les concepts fait et dimension on été représentés ce qui est suffisant pour la structuration des besoins d'analyse. Et par conséquent le fait dégénéré ne nécessite pas une représentation non plus.

6.3 Implémentation

Pour valider notre proposition, nous avons implémenté un prototype sous Oracle 11g.
Le but de cette implémentation est de créer un outil pouvant générer via une interface interactive tout modèle conceptuel d'entrepôt de données.
Le langage de programmation PL/SQL a été utilisé pour cette phase d'implémentation.

A travers une interface, nous pouvons créer le méta modèle et utiliser des procédures d'insertion pour le remplir et ceci pour chaque entité. Par exemple pour Dimension, on crée la add_dimension qui permet de stocker les informations concernant la dimension à savoir le nom, l'Id_dim, description, pour les hiérarchies, une procédure add_hiérarchie permet d'abord de vérifier le type de hiérarchie simple ou multiple, si elle est simple, on doit choisir entre stricte ou non stricte …et ensuite on insert les informations correspondantes, et ainsi de suite avec toutes les entités. Une fois le M^{3}Gen instancié, nous utilisons une procédure pour générer le modèle d'entrepôt. C'est ainsi que l'entrepôt de données Vente a été généré à partir de M^{3}Gen. Les paramètres utilisés sont spécifiques à chaque procédure.
Pour la procédure add_dimension, les paramètres sont : NomD « Nom dimension », Id_D « Id_dimension », description « description », TypeD « type de la dimension ».
Pour la procédure add_hiérarchie, les paramètres sont : Id_H « l'Id de la hiérarchie », nomH « nom de la hiérarchie », typeH « type de la hiérarchie », Idd « la dimension à laquelle l'hiérarchie est liée », Id-N « le niveau lié à la hiérarchie », typeHS « type d'hiérarchie simple », typeHM « type d'hiérarchie multiple, nbre_N « le nombre de niveau » …
Nous montrons ci dessous par un exemple le modèle conceptuel d'entrepôt de données « VENTES» (FIG1).
Par construction, le M^{3}Gen impose que toutes les propriétés d'une entité ont vocation à être renseignées (il n'y a pas de propriété « facultative »).
Le modèle qui en découle du M^{3}Gen dont le domaine d'activité est la « VENTES » est le suivant :

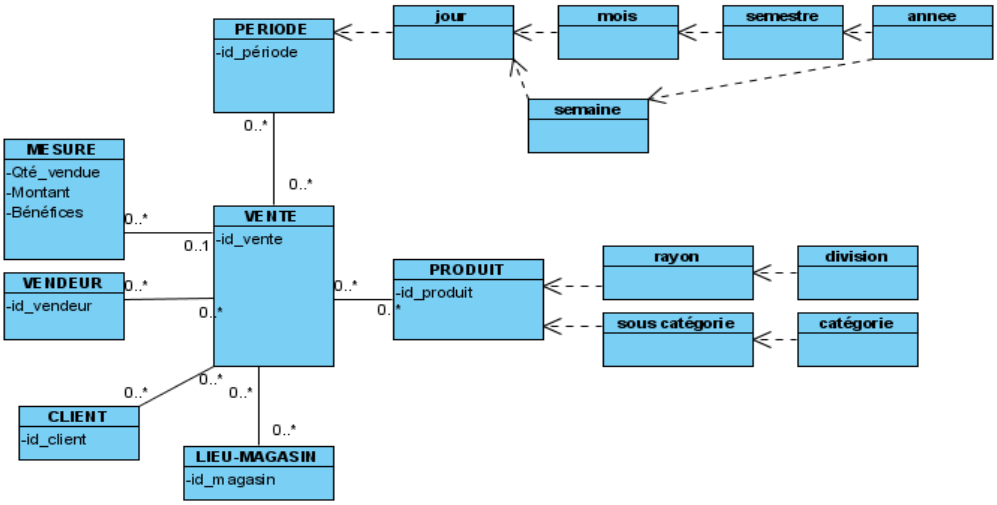


FIG 3 : Modèle conceptuel d'entrepôt VENTES par instanciation de M^{3}Gen

Ce modèle conceptuel est élaboré à partir du M^{3}Gen. On reprend tout simplement les règles de gestion énoncées plus haut à savoir qu'un fait peut posséder plusieurs dimensions ce qui est le cas du fait VENTE qui dans ce modèle possède 5 dimensions. Une dimension peut avoir un type d'hiérarchie et pour chaque type d'hiérarchie on à un nombre de niveau. La dimension PRODUIT à une hiérarchie multiple parallèle de 2 niveaux. En appliquant donc toutes les règles liées à l'élaboration du M^{3}Gen, on instancie ainsi notre modèle conceptuel de l'entrepôt de données du domaine VENTE. On retrouve bien notre fait (vente) les dimensions

liés à cette vente (période, produit, client, vendeur) les hiérarchies liées aux dimensions. Hiérarchie multiple alternative avec la dimension période ainsi que ces niveaux. Hiérarchie multiple parallèle avec la dimension produit et ces niveaux. Et on retrouve également le concept mesure qui est bien lié à notre fait (vente).

7 Conclusion et perspectives

Cet article présente un travail sur la définition d'un modèle conceptuel multidimensionnel standard unifiant dans un même modèle générique toutes les propriétés nécessaires à la conception d'un modèle multidimensionnel. Nous avons tout d'abord fait une étude bibliographique approfondie sur la modélisation multidimensionnelle, grâce à laquelle nous avons pu recensé les propriétés liées aux concepts de base. Il en ressort qu'aucun méta modèle proposé dans la littérature ne prend en compte les concepts de base ainsi que l'ensemble des propriétés qui leurs sont liées. Ensuite nous avons proposé une nouvelle classification de ces propriétés permettant de dégager les propriétés liées à la phase de modélisation multidimensionnelle du processus d'entreposage, ces derniers nous ont permis d'élaborer notre méta modèle (M^3Gen). Pour valider notre proposition, nous avons implémenté un prototype sous Oracle 11g qui à permis l'instanciation du M^3Gen. Par ailleurs nous poursuivons cette implémentation qui permettra de valider l'outil de génération automatique de tout modèle d'entrepôt de donnée. Bien évidement le modèle M^3Gen prend en compte le type de donnée simple (numérique) dans la suite de ce travail, nous définirons des propriétés supplémentaires qui prendrons en compte les données de types complexes (multimédia, objet mobile, etc.….), les propriétés liées à la qualité et qui permettrons d'évaluer la qualité des systèmes d'information décisionnel , les propriétés liées à la sécurité dans toutes les phases du processus d'entreposage (ETL, Modélisation multidimensionnelle, Analyse OLAP). Les propriétés liées à l'aspect dynamique et qui correspondent à la phase ETL du processus d'entreposage sont des éléments structurant d'un modèle, nous verrons donc comment représenter ces propriétés à travers le méta modèle. Une fois toutes ces propriétés recensées, nous les intégrerons dans le méta modèle pour le rendre complet. Le M^3Gen génère donc tout modèle conceptuel d'entrepôt de donnée, nous élaborerons les modèles logiques autre que la représentation en étoile, et le modèle physique qui en découle.

Références

Abelló, A., Samos, J., and Saltor, F. (2001). A framework for the classification and description of multidimensional data models. In Mayr et al. [2001], pages 668–677.3540425276.

Abelló, A., Samos, J., and Saltor, F. (2002), Yam^2 (yet another multidimensional model) : An extension of uml. In Nascimento et al. [2002], pages 172–181.0769516386.

Amous I., Jedidi A., Sedes F., (2002), A contribution to multimedia document modeling and organizing. In: BELLAHSENE Z., PATEL D., ROLLAND C. Eds. Object-oriented information systems : 8th international conference, OOIS 2002, 2-5 septembre 2002, Montpellier, France. Heidelberg, Berlin, Allemagne: Springer-Verlag, 2002, pp 434-444. (Lecture notes in computer science, n° 2425)

Annoni, E. (2007). Eléments méthodologiques pour le développement des systèmes décisionnels dans un contexte de réutilisation. Thèse de doctorat, Université Paul Sabatier – Toulouse1, Toulouse, France.

Blaschka, M., Sapia, C., Hofling, G., and Dinter, B. (1999). An overview of multidimensional data models for olap. Technical report. Technical Report FR-1999-001.

Boukraâ, D., Bouss aid, O. (2008). A multidimensional Conceptual model for Complex Data, in Encyclopedia of Data Warehousing and Mining - 2nd Edition, John Wang Eds , to appear.

Bonifati,F cattaneo, S Ceri, A Fugetta, and S Paraboschi. Designing Data Marts for Data Warehouse.ACM Transactions on Software Engineering and Methodology, 10(4): 452-483, 2001.

Cabibbo L and R Torlone. A Logical Approach to Multidimensional Databases. In Vith International Conference on Extending database technology (EDBT 98), Valencia, Spain,volume 1377 of LNCS, pages 183-197. Springer, 1998.

Chrisment, C., Pujolle, G., Ravat, F., Teste, O., and Zurfluh, G. (2005). Les entrepôts de données. In Zurfluh, G., editor, Trait´e Informatique des Techniques de l'Ingénieur - H3870, page 10. Techniques de l'Ingénieur.

Golfarelli, M., Maio, D., and Rizzi, S. (1998). The dimensional fact model : A conceptual model for data warehouses. Int. J. Cooperative Inf. Syst., 7(2-3) :215–247.

Hurtado C A and A O Mendelzon. Reasoning about Summarizability in heterogeneous Multidimensional Schemas. In Vith International Conference on Database Theory (ICDT 01), London, UK, volume 1973 of LNCS, pages 375-389. springer, 2001.

Husemann, B., Lechtenborger, J., and Vossen, G. (2000). Conceptual data warehouse modeling. In Jeusfeld et al. [2000], page 6.

Hüsemann, J. Lechtenbörger, G. Vossen, (2000). Conceptual datawarehouse design, 2nd International Workshop on Design and Management of Data Warehouses.

Inmon, W. H. (1996). Building the data warehouse (2nd ed.). John Wiley & Sons, Inc., New York, NY, USA. 0471141615.

Jarke, M., Lenzerini, M., Vassiliou, Y., and Vassiliadis, P. (2001). Fundamentals of Data Warehouses. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 3540420894.

Jarke, M. Lenzerini, Y. Vassiliou, P. Vassiliadis,(2003). Fundamentals of Data Warehouses, second ed., Springer-Verlag.

Kimball, R. (1996). The data warehouse toolkit : practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc., New York, NY, USA.

Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., and Thornwaite, W. (1998). The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing. John Wiley & Sons, Inc., New York, NY, USA. 0471255475.

Lujan-Mora, S. (2005). Data Warehouse Design with UML. PhD thesis, Department of Software and Computing Systems, University of Alicante, Alicante, Espagne.

Mahboubi, H., Hachicha M., Darmont D. (2008). XML Warehousing and OLAP. Encyclopedia of Data Warehousing and Mining, Second Edition, IGI Publishing, Hershey, PA, USA.

Moody D L and M A R Kortink. From Entreprise Models to Dimensional Models: a Methodology for Data Warehouse and Data Mart Design. In Design and Management of Data Warehouses, page 5, 2000.

Phipps C and K C Davis. Automating Data Warehouse Conceptual Schema Design and Evaluation. In Vith International Workshop on design and Management of Data warehouse(DMDW 02),Toronto,Canada, volume 58 of CEUR Workshop Processing, pages 23-32. CEURWS.org, 2002.

Prat, J. Akoka, From UML to ROLAP Multidimensional Databases Using a Pivot Model, 18èmes Journées Bases de données Avancées, (Evry, France, Oct. 2002), http://faculty. essec.fr/n.prat/BDA02_Prat_Akoka.pdf.

Rafanelli, M., editor (2003). Multidimensional Databases : Problems and Solutions. Idea Group. 1591400538.

Rizzi, S., Abell´o, A., Lechtenb¨orger, J., and Trujillo, J. (2006). Research in data warehouse modeling and design : dead or alive ? In Song and Vassiliadis [2006], pages 3–10. 1595935304.

Rusu, L., Rahayu, J. W., and Taniar, D.,(2005). A Methodology for Building XML Data Warehouses, International Journal of Data Warehousing & Mining, 67-92

Torlone, R. (2003). Conceptual multidimensional models. In Multidimensional Data bases : Problems and Solutions, pages 69–90.

Tournier, R. (2007). Analyse en ligne (OLAP) de documents. Thèse de doctorat, Université Paul Sabatier – Toulouse1, Toulouse, France.

Vassiliadis, P. and Sellis, T. K. (1999). A survey of logical models for olap data bases.SIGMOD Record, 28(4) :64–69

Object Management Group, OMG Modeling and Metadata Specifications, http://www.omg.org/technology/documents/modeling_spec_catalog.htm.

Élaboration de schémas de magasins de données à partir d'une base de données objet

Salma Ben Mefteh, Jamel Feki, Yasser Hachaichi

Laboratoire MIRACL, Faculté des Sciences Economiques et de Gestion de Sfax
B.P. 1088 - Sfax 3018, Tunisie

salmabm2007@yahoo.fr, {Jamel.Feki, Yasser.Hachaichi}@ fsegs.rnu.tn

RÉSUMÉ. L'entrepôt de données est un support d'aide à la décision, fondé sur une base de données fédérant et homogénéisant des informations collectées dans différents services d'une organisation. L'élaboration d'un schéma d'entrepôt de données passe par les étapes de modélisation conceptuelle, logique et physique. Dans cet article nous nous intéressons à la modélisation conceptuelle. En effet, nous proposons une approche ascendante automatisable de conception de schémas multidimensionnels à partir d'une base de données objet. Cette approche extrait les concepts multidimensionnels (faits, mesures et dimensions) moyennant un ensemble d'heuristiques que nous définissons. Ces dernières exploitent les relations entre les objets et les types des attributs ; elles produisent des schémas de magasins de données en étoile qui seront proposés au concepteur pour validation.

1. Introduction et motivations

La gestion et le pilotage des entreprises nécessitent des systèmes décisionnels assez performants permettant aux décideurs, quels que soient leurs niveaux de responsabilité, d'accéder rapidement et facilement aux informations qui leurs sont utiles. Les systèmes d'information décisionnels (SID) tentent de satisfaire cet objectif. Ces systèmes comprennent essentiellement deux types de composants : un *entrepôt de données* (ED) et des *magasins de données* (MD). L'entrepôt est une collection de données historisées, intégrées, organisées par sujets d'analyses permettant d'offrir les informations nécessaires à la prise de décision. Afin de faciliter l'accès et la manipulation des données, un ED est restructuré en un ensemble de MD, chacun est un extrait ciblé et optimisé de l'entrepôt. Il s'agit d'un sujet d'analyse utile pour une classe d'utilisateurs ou un besoin d'analyse spécifique. Les données d'un magasin peuvent être aisément manipulées grâce à des requêteurs, des tableurs et des logiciels d'analyse (Teste, 2000).

L'élaboration d'un schéma d'ED ou de MD est une phase souvent manuelle ; elle passe par les étapes de modélisation conceptuelle, logique et physique (Prat et al., 2006). Le niveau conceptuel traite la représentation conceptuelle des données (complexes et simples) à incorporer dans l'ED. Ces données proviennent de systèmes d'information opérationnels pouvant être modélisés en relationnel, objet, etc. L'entrepôt peut à son tour être structuré comme une base de données conventionnelle.

Dans la littérature des SID, certains travaux ont proposé des modèles orientés objets pour la conception des ED (Lujàn-Mora et al., 2006), d'autres ont préconisé des démarches semi-

automatiques de conception d’ED/MD à partir de modèles conceptuels orientés objets (*i.e*, diagramme de classe UML) (Zribi et Feki, 2007), (Prat et al., 2006) et (Zepeda et al., 2005). Cependant, et à notre connaissance, il n’existe pas d’approches qui partent d’un schéma logique objet *(i.e,* SGBD Objet) pour construire un schéma multidimensionnel.

Dans cet article nous traitons cette limite en proposant une approche semi-automatique de génération de schémas multidimensionnels en étoile à partir d’une source objet. Notre choix de l’objet est guidé par deux motivations complémentaires : D’une part, compléter les travaux de recherche de notre équipe en vue de pouvoir construire des ED/MD à partir de sources hétérogènes, sachant que nous avons déjà exploré le passage du relationnel vers le multidimensionnel (Feki et Hachaichi, 2007a) et que nous sommes assez avancés sur la construction de schémas multidimensionnels à partir de documents XML conformes à des DTD (« Document Type Definition ») (Hachaichi et al., 2008). D’autre part, les sources BD objet sont de plus en plus présentes au sein des entreprises et présentent des spécificités intéressantes à exploiter. Parmi ces caractéristiques, nous citons la richesse sémantique en éléments de modélisation (*e.g.,* généralisation/spécialisation, héritage) qui aide à identifier les concepts multidimensionnels.

L’approche que nous proposons est fondée sur un ensemble d'heuristiques automatisables qui identifient les concepts multidimensionnels (faits, mesures, attributs …) qui permettent d’associer à chaque concept extrait son correspondant (*i.e.,* classe, attribut …) dans la source.

Cet article est organisé en trois sections : la section 2 présente une étude critique de l’état de l’art des approches de conception des systèmes d’informations décisionnels. La section 3 donne un aperçu de notre démarche de conception. La section 4 détaille les heuristiques d’extraction des concepts multidimensionnels (fait avec ses mesures et dimensions).

2. Étude critique de l'état de l’art sur la conception de systèmes décisionnels

Les approches existantes de conception de SID peuvent être classées en trois types : *descendantes*, *ascendantes* et *mixtes*. Les approches descendantes (Kimball, 2002) partent des besoins décisionnels d’une activité spécifique. Les approches ascendantes se basent sur les sources de données de l’entreprise (Moody et Kortink, 2000), (Golfarelli et al. ,1998), (Hüsemann et al., 2000), (Feki et Hachaichi, 2007a), (Feki et Hachaichi, 2007b), (Lujàn-Mora et al. , 2006), (Romero et Abello, 2007), (Ouaret et al., 2007) et (Zribi et Feki, 2007). Finalement, les approches mixtes combinent les deux approches précédentes (Soussi et al., 2005), (Zepeda et al.,2005) et (Prat et al., 2006) en vue de tirer profits des avantages des deux précédentes.

Dans la première approche les schémas construits constituent une solution qui respecte les besoins des utilisateurs finaux. Dans l’approche ascendante le concepteur peut bénéficier

des relations existantes entres les données et suivre une approche plus structurée pour concevoir la base de données décisionnelles de l'entreprise. Cependant, les méthodes adoptant cette approche sont décrites par des exemples au lieu de procédure explicite de conception et peut mener à des conceptions incorrectes si le concepteur ne comprend pas les relations entres les données (Moody et Kortink, 2000).

Dans la littérature plusieurs méthodes ont adopté ce type d'approche.

- *Diagramme E/R* (Moody et Kortink, 2000), (Golfarelli et al. ,1998), (Hüsemann et al., 2000),
- *Source relationnelle (* Feki et Hachaichi,2007a), (Feki et Hachaichi,2007b),
- *Diagramme de classe UML* (Lujàn-Mora et al. , 2006), (Zribi et Feki, 2007),
- *Documents XML* (Hachaichi et al., 2008), (Ouaret et al., 2007) ,
- *Ontologie* (Romero et Abello, 2007).

Nous nous limitons dans cette partie à détailler les méthodes qui partent de modèles objets (*i.e.,* UML et ontologie décrite par un profil UML).

(Zribi et Feki, 2007) propose une approche semi-automatique de conception de schémas de MD en partant d'un diagramme de classes UML. L'approche consiste à : *1)* construire des packages décisionnels autour des entités du monde réel matérialisant des entités de transaction (produisant des faits), et *2)* identifier et annoter les éléments multidimensionnels dans les classes de chaque package. Cependant, cette approche ne traite pas les mesures calculées.

Dans (Romero et Abello, 2007) les auteurs produisent automatiquement des schémas multidimensionnels à partir d'une ontologie décrivant le domaine d'affaires. Afin d'atteindre leurs objectifs, ils développent une nouvelle méthode ascendante de conception de schémas de MD. Leur méthode est constituée des trois tâches suivantes : *1)* précision des sujets d'analyses (faits) ; *2)* identification des dimensions potentielles ; et *3*) détermination des hiérarchies des dimensions identifiées dans la tâche 2. Le résultat de ces trois tâches est un schéma multidimensionnel visualisé selon la notation spécifique starUML de (Moody et al., 2000). Les ontologies sont sémantiquement plus riches que les schémas relationnels au niveau des métadonnées. Cependant l'ontologie utilisée est conventionnelle, c'est-à-dire, qu'elle présente les concepts du domaine sans mettre en évidence ni les concepts multidimensionnels ni les relations entre ces concepts qui pourraient les relier. Ce travail semble peu différent d'une méthode ascendante partant d'un diagramme E/R.

Dans (Lujàn-Mora et al., 2006) les auteurs proposent une approche ascendante de construction de schéma de MD en utilisant une extension du langue UML et en employant un profil UML. Ce profil est défini par un ensemble de stéréotypes, de contraintes et des valeurs étiquetées pour représenter d'une manière élégante les propriétés principales de MD au niveau conceptuel (comme les liens multiples *(n-n)* entre les faits et les dimensions). Pour accomplir leur proposition, ils présentent l'utilisation des packages UML afin de grouper des classes dans trois niveaux plus élevés : *1) le niveau 1* représente le modèle de définition du schéma conceptuel de MD ; *2) le niveau 2* concerne la définition du schéma en étoile, et 3) *le 3*ème concerne la définition de dimension/fait. Cette approche jouit d'un double avantage : d'une part, elle se base sur UML qui est un langage de modélisation standard bien connu ; d'autre part, l'utilisation des packages permet de représenter des modèles énormes et complexes à différents niveaux de complexité et aussi d'importer des éléments définis dans

un package vers un autre, évitant ainsi la redondance. Cependant, cette approche ne traite pas les faits dégénérés et les liens multiples *(n, n)* entre les faits et les dimensions.

De leur part, (Prat et al., 2006) ont traité la problématique de conception d'ED basée sur un diagramme de classe UML. La méthode proposée est décrite par un ensemble de méta-modèles utilisés à chaque phase de conception et une formalisation des règles de transformation semi-automatisées entre les méta-modèles de chaque phase. Les principaux avantages de cette méthode sont :*1)* elle se base sur la notation graphique UML assez populaire, et *2)* elle possède un niveau élevé d’automatisation obtenue au moyen des heuristiques. Cependant, cette méthode ne traite pas les mesures calculées.

(Zepeda et al., 2005) proposent une méthode de conception de MD mixte à partir de sources opérationnelles (diagramme de classe UML) et de besoins décisionnels. Elle est constituée de deux phases essentielles : *1)* obtention d’un schéma candidat multidimensionnel à partir d’un diagramme de classe UML, et *2)* raffinement du schéma obtenu par les exigences des utilisateurs décisionnels à l’aide d’interviews. Cette méthode exige du concepteur un effort intense lors de la deuxième phase car elle est effectuée manuellement.

Étant semi-automatique ou manuelle, ces approches nécessitent beaucoup d’efforts pour l’identification du fait à analyser et de ses dimensions. Afin d’éviter cette limite et travailler sur la version la plus récente du système d’information opérationnel, notre approche ascendante de construction de schémas de MD part d’une BD objet décrite par son schéma logique et exploite les éléments de base du schéma source : d’une part, les cardinalités des relations entres les objets, et d’autre part, les types des attributs (numérique, temporelle ou textuel) ainsi que les relations interclasses.

3. Aperçu de notre approche

Dans cette section, nous décrivons succinctement les étapes de notre approche de construction de schéma de MD à partir d’une base de données orientée objet. Cette approche construit des schémas de MD candidats en exploitant les relations entre les objets ainsi que les types de leurs attributs (numérique, booléen, date/heure ou caractère). Elle se compose de deux étapes : *l’extraction des concepts multidimensionnels de MD* ; et la *validation* (cf. Figure 1).

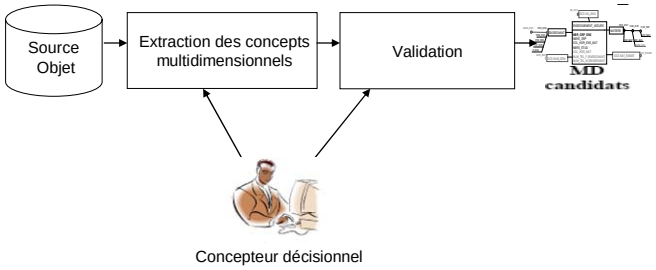


FIG. 1 - Approche de construction de schéma de MD.

Extraction des concepts multidimensionnels : Cette étape produit des schémas de MD en étoile à partir d'un schéma logique d'une BD objet. Pour ce faire, elle commence par accéder au référentiel du SGBD pour y extraire le schéma logique. Ensuite, elle identifie les concepts multidimensionnels (i.e., faits et leurs mesures ainsi que les dimensions avec leurs attributs organisés en hiérarchies) en appliquant automatiquement un ensemble d'heuristiques. Ces heuristiques se définissent en examinant les liens entre les objets et les types de leurs attributs.

Validation : Le concepteur peut intervenir pour valider les schémas en étoile candidats construits, c'est-à-dire, adapter ces schémas aux besoins analytiques du système de pilotage (e.g. supprimer ou renommer des éléments du schéma tout en préservant l'origine de l'élément dans la source).

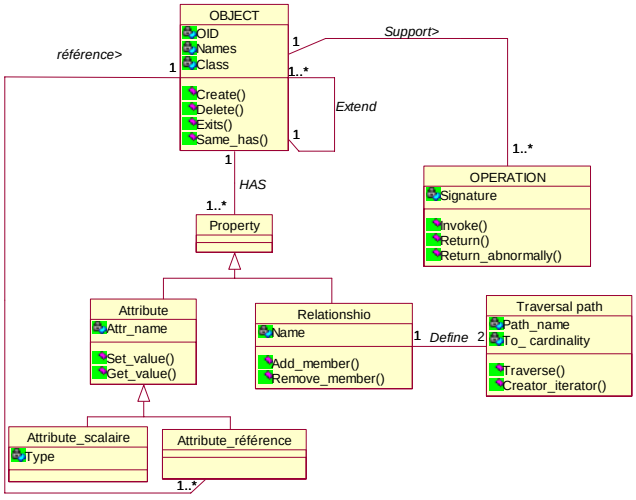


FIG. 2 - Vue simplifiée du méta-modèle du schéma objet du SGBD objet Matisse.

Pour mieux expliquer les différentes étapes, nous commençons par décrire les composants d’un schéma objet (*i.e.,* le méta-modèle objet).

Le méta-modèle objet du SGBD objet Matisse[1] que nous utilisons pour illustrer notre approche est partiellement montré dans la figure 2 ; il est conforme à la norme ODMG[2] (Cattell et Douglas, 1997). Dans ce méta-modèle chaque objet est identifié par son OID (« Object Identifier »), décrit par son nom « Name » et sa classe d’appartenance « Class ». Un objet possède une ou plusieurs propriétés pouvant être des attributs « Attribute » ou des liens entre objets « Relationship ». Un attribut peut être un scalaire (entier, réel, chaîne de caractère, date …) ou une référence vers un autre objet. En en outre, un lien entre deux objets possède deux rôles dans l’objet « Traversal path ». Un rôle « traversal path » est caractérisé par un nom « Path_name » et une cardinalité « To_cardinality ».

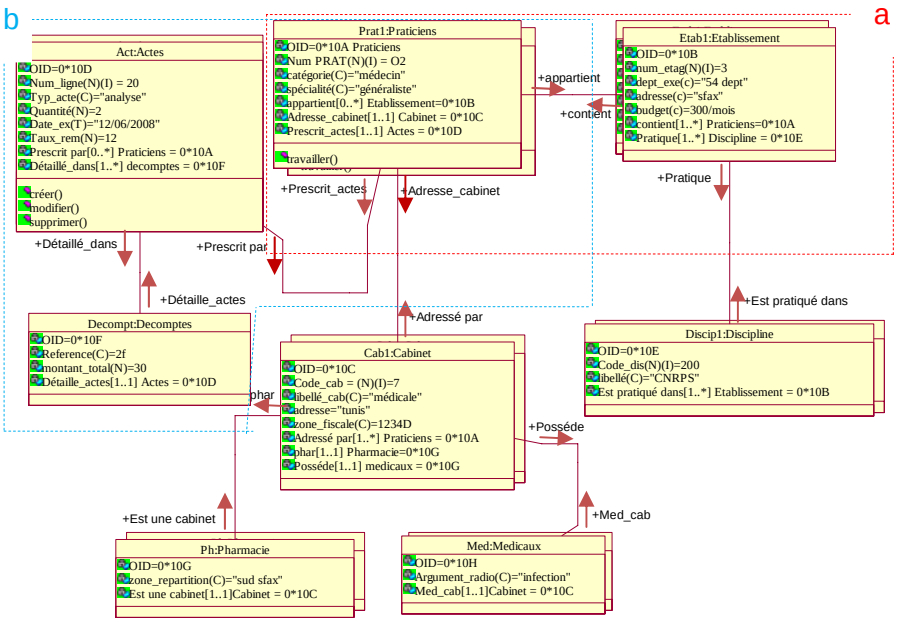


Fig. 3 - Un exemple de schéma objet pour la source UML du «système assurance maladie ».

[1] Matisse est un gestionnaire de base de données orienté objet. Sa puissance réside principalement dans sa gestion optimisée de son architecture client-serveur et par la puissance de son méta schéma.

[2] ODMG « Object Data Management Group » est une norme des schémas des bases de données objet.

La figure 3 (Bret et Teste, 1999). illustre un exemple d'instanciation du méta-modèle objet de la figure 2.

Cet exemple est un extrait du modèle objet du « système assurance maladie» tiré de (Bret et Teste, 1999). Ce schéma décrit les données relatives au domaine médical où chaque praticien prescrit une acte, intervient dans des établissements (moyennant deux rôles *appartient* et *contient*), et dirige un cabinet (rôles *Dirige_cabinet* et *dirige*).

Chaque établissement possède plusieurs disciplines à travers les deux rôles Pratique et Est_Pratiqué_dans. Chaque acte est détaillé dans un ou plusieurs décomptes (rôles Détaillé_dans et Détaile_actes). Les lettres *N, C, T et I* devant les attributs désignent respectivement les types : numérique, caractère, temporel et index.

Nous détaillons dans la section suivante l'étape d'extraction de concepts multidimensionnels. Pour des raisons de limitation d'espace, nous nous limitons dans ce papier à l'identification des faits, des mesures et des dimensions.

4. Extraction des concepts multidimensionnels

Cette étape produit des schémas de MD candidats pour une source objet décrit par son schéma logique. Elle se base essentiellement sur les multiplicités des liens entre objets et les types des attributs pour identifier les concepts multidimensionnels moyennant un ensemble de règles d'extraction que nous illustrons sur l'exemple de la Figure 3.

Soit *O1* et *O2* deux objets, nous désignons par :

- Lien *(x, 1)* : un lien entre *O1* et *O2* telle que *x* et *1* désignent respectivement les cardinalités maximales du côté d'O1 et de *O2,* et $0 \leq x \leq 1$.
- Lien *(x, *)* : un lien entre *O1* et *O2* telle que *x* ($0 \leq x \leq 1$) et * désignent respectivement les cardinalités maximales du côté de *O1* et de *O2*.
- Lien *(*, *)* : un lien entre *O1* et *O2* avec deux rôles de multiplicité *(x, *)* et *(x, *)* respectivement du côté *O1* et *O2* avec $0 \leq x \leq 1$.
- Fermeture transitive d'*O1* : tous les objets directement ou transitivement liés à *O1* par un lien *(1,1).*

Notons qu'il existe d'autres types de liens aussi pertinents pour la conception multidimensionnelle, comme par exemple, l'héritage et la composition. Nous ne les abordons pas dans cet article.

L'extraction des concepts multidimensionnels commence par l'identification des faits.

4.1 Extraction des faits

Le fait représente un centre d'intérêt pour la prise de décision. En effet, il modélise un sujet d'analyse représentant un événement qui se produit au sein d'une organisation.

Dans le cadre des méthodes de conception ascendantes, les faits sont identifiés manuellement/semi automatiquement parmi :

- Les représentations conceptuelles (entités ou associations n-aire) (Kimball, 2002), (Soussi et al., 2005),

- Les relations (Feki et Hachaichi, 2007a) possédant au moins un attribut numérique non clé (primaire ou étrangère).
- Les associations (*, *) non porteuses de données dans le diagramme de classe UML (Prat et al., 2006). Ce type d’association produit des faits de type *suivi d’événement* c’est-à-dire, ne contenant aucune mesure « factless fact » (Kimball, 2002).

Dans notre approche, nous identifions les faits candidats à partir des liens *(*, *)* entre objets en appliquant l’heuristique *Hf 1* ou à partir d’un objet par l’heuristique *Hf 2*.

Fait construit à partir d’un lien entre objets

Hf1 : Chaque lien *(*, *)* entre deux objets est transformé en un fait.

Exemple :

Dans la figure *3-a*, le lien *(*, *)* entre *Praticiens* et *Etablissement* est transformé en un fait nommé *appartient_contient*, il est sans mesure. Cet événement permet le suivi du travail des Praticiens dans les Etablissements et leurs évolutions.

Fait construit à partir d’un objet

Hf2 : Chaque objet contenant au moins un attribut numérique non identifiant et contenant, au moins, un lien *(x, *)* vers un autre objet, devient un fait *F* avec $0 \leq x \leq 1$ et la multiplicité * est du côté de *F* (nous écartons les objets dont le seul attribut numérique est l’identifiant) (Zepeda et al., 2005).

Nous pouvons améliorer ces heuristiques par le critère suivant : un Objet-fait est un objet fréquemment mis à jour dans la source objet (un objet qui contient des attributs dynamiques) (Kimball, 2002). Il s'agit d’un objet de transaction (Moody et Kortink, 2000), (Golfarelli et al. ,1998). L’analyse du journal des transactions peut aider à réaliser une telle amélioration.

Exemple :

Dans la figure 3-b, l’objet *Actes* est un fait puisqu’il satisfait l’heuristique *Hf 2*. En effet, il contient deux attributs numériques non identifiants (*Quantité et Taux_rem),* de plus il est relié aux objets *Praticiens* et *Decomptes* par un lien *(x, *)* (avec la multiplicité * est du côté du fait *Actes*).

L’extraction des faits se complète par celle des mesures.

4.2 Extraction des mesures

Les mesures d'un fait sont numériques et généralement valorisées de manière continue (Kimball, 2002). Elles sont numériques pour permettre de résumer un grand nombre d'enregistrements en quelques uns (on peut les dénombrer ou bien les agréger) durant les opérations OLAP (Forages Haut et bas). Il s'agit d’observations régulières des évolutions du fait par rapport à des objectifs fixés (Teste, 2000).

Nous déterminons les mesures d’un fait *F* (défini dans *Hf 2*) à partir de *F* ou des objets qui lui sont liés. L’heuristique *Hm 1* identifie les mesures.

Mesure extraite d'un objet obtenu par Hf 2

Hm1 : Les mesures d'un fait *F* défini sur un Objet-fait sont les attributs numériques non identifiant appartenant à :

- L'objet-fait *F* (nous pouvons inclure aussi les attributs booléens comme mesure).
- L'objet *O* lié à *F* par un lien *(1,1)* ou *(0,1)* avec la multiplicité 0 du côté de *F*. (Romero et al., 2007),(Soussi et al., 2005).

Ce dernier point vérifie les deux propriétés suivantes (Romero et al., 2007) :

- *Complétude* : la multiplicité maximale 1 du côté de l'objet *O* garantit que chaque instance du fait *F* doit obligatoirement être reliée à une seule valeur de mesure alors que l'interdiction de la valeur 0 (du côté de l'objet *O)* permet d'assurer la complétude.
- *Disjonction* : la multiplicité 1 du côté du fait *F* assure la disjonction entre les faits du schéma multidimensionnel. C'est-à-dire qu'une instance de mesure n'est liée qu'à un seul fait. Malgré l'existence de la multiplicité 0 du côté de fait, la propriété de disjonction entre les faits reste valable mais le seul problème c'est que la mesure peut ne pas être reliée à aucune instance de fait.

Exemple :

L'application de l'heuristique *Hm1* sur le schéma objet de notre exemple dégage les mesures *Quantité* et *Taux-rem* pour le fait *Actes*.

Pour compléter la construction des schémas de MD candidats, nous déterminons pour chaque fait ses dimensions.

4.3 Extraction des dimensions

Le sujet ou fait est observé et analysé suivant différentes dimensions. Une dimension modélise un axe d'analyse et se compose d'attributs correspondant aux informations faisant varier les mesures de l'activité. Certains de ces attributs (dits paramètres) sont ordonnés pour former des perspectives d'analyse (*i.e.,* hiérarchies) ; d'autres sont descriptifs et servent à libeller les paramètres ou à restreindre la portée des requêtes afin de limiter la taille des réponses. Les paramètres sont discrets, c'est à dire que leurs valeurs possibles sont bien déterminées et sont des descripteurs constants (Teste, 2000).

Dans notre méthode, les dimensions sont extraites à partir d'un attribut (booléen ou date) d'un fait (dimensions dégénérées (Kimball, 2002)) ou à partir d'un objet.

Dimension construite à partir d'un attribut

Cas d'un attribut booléen. Un attribut booléen appartenant à un fait *F* constitue un axe d'analyse. D'où l'heuristique suivante :

Hd1. Tout attribut booléen appartenant à un fait *F* génère une dimension candidate pour *F* dont il est l'identifiant (Feki et Hachaichi, 2007b).

Cas d'un attribut temporel. Dans le modèle multidimensionnel, la dimension *Temps* figure systématiquement dans tout entrepôt (Kimball, 2002) considéré comme une série temporelle. Nous construisons une dimension à partir d'un attribut temporel par la règle suivante :

Hd2. Tout attribut temporel (date ou temps) appartenant à un fait *F* construit une dimension temporelle dont il est l'identifiant.

Exemple

L'application de l'heuristique *Hd2* sur le schéma objet du « système assurance maladie » identifie l'attribut *Dat_ex* comme dimension pour le fait Actes.

Dimension construite à partir d'un objet

Pour déterminer les dimensions d'un fait *F*, nous nous basons sur les liens entre un fait *F* et ses objets reliés. Nous définissons les heuristiques suivantes :

Hd3. Tout objet *O* directement lié à un fait *F* par un lien *(1, n)* (avec une multiplicité *n* du côté du *F* et la multiplicité *1* du côté de *O)* est une dimension candidate pour *F*. Le nom de cette dimension est celui de *O*, son identifiant est l'attribut index de *O* (Soussi et al., 2005), (Romero et Abello, 2007), (Zribi et Feki, 2007).

Exemple :

L'application de l'heuristique *Hd3* sur notre exemple courant identifie les deux objets *Praticiens* et *Decomptes* comme dimensions pour le fait *Actes*.

Hd4. Tout objet *O* appartenant à la fermeture transitive des objets trouvés dans *Hd3* est une dimension pour *F*.

La figure 4 présente le schéma en étoile résultat du magasin de données généré à partir du schéma objet de la figure 3. Ce schéma limite la représentation des dimensions à leurs noms puisque nous n'avons pas abordé la détermination des hiérarchies dans cette présentation.

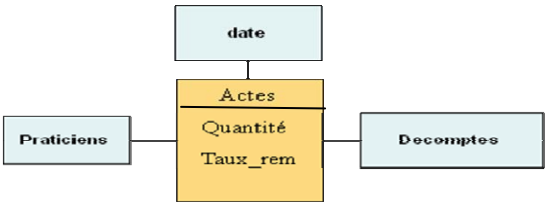


Fig. 4 - Schéma en étoile du MD pour le fait Actes.

5. Conclusion

Ce travail présente une approche ascendante de conception semi-automatique de schémas de MD en partant d'une source objet décrite par son schéma logique. L'approche consiste à extraire les éléments multidimensionnels (faits, mesures et dimensions). Pour ce faire, nous avons proposé un ensemble d'heuristiques d'identification des concepts multidimensionnels de fait, mesures et dimensions. Le résultat est visualisé par un schéma en étoile selon le

formalisme DFM (« Dimensional Fact Model ») de Golfarelli. Les schémas obtenus peuvent être présentés au décideur pour les adapter et exprimer ainsi ses besoins particuliers.

Dans cet article, nous nous sommes limités à l'extraction des concepts multidimensionnels fait, mesure et dimension. Nous n'avons pas détaillé l'extraction des attributs dimensionnels (paramètres de niveau et attributs faibles) et la construction des hiérarchies et ceci par contrainte d'espace. Nous travaillons sur l'identification des hiérarchies en exploitant essentiellement les liens entre classes et nous comptons poursuivre ce travail par le développement d'un outil logiciel qui automatise la construction de schémas multidimensionnels à partir d'une source de données objet et donner la main au décideur pour valider/adapter les schémas de MD obtenus.

Références

Bret F., Teste O. (1999). *Construction graphique d'entrepôts et de magasins de données*. INFORSID'99, La Garde (France).

Cattell R. G. G., Douglas K. B. (1997). *The Object Database Standard: ODMG 2.0*, Ed. Morgan Kaufmann Publishers.

Golfarelli M., Maio D., Rizzi S., (1998). *Conceptual design of data warehouses from E/R schemes*. 31st Hawaii International Conférence on System Sciences.

Feki J., Hachaichi Y. (2007a). *Du relationnel au multidimensionnel : Conception de magasins de données, Revue des Nouvelles Technologies de l'Information* (RNTI), Ed. Cépaduès, vol. n° B-3, pp.5-19.

Feki J., Hachaichi Y. (2007b). *Conception assistée de MD : Une démarche et un outil*, *Journal of Decision Systems* (JDS). Ed. Lavoisier, vol. 16 – No.3/2007, pp. 303-333.

Hachaichi Y., Feki J., Ben-Abdallah H. (2008). *Du XML au multidimensionnel: Conception de magasins de données, Revue des Nouvelles Technologies de l'Information* (RNTI), Ed. Cépaduès, vol. n° B-4, pp. 45-59.

Husemann B., Lechtenbörger J., Vossen G., (2000). *Conceptual Data Warehouse Design.* Proc. Of the Int'l Workshop on Design and Management of Data Warehouses Stockholm Suede, pp 1-6.

Kimball R. (2002). *The Data Warehousing Toolkit second edition*, Wiley, New York.

Lujàn-Mora S., Trujillo J., Song Y. (2006). *A UML profile for multidimensional modeling in data warehouses*. Data & Knowledge Engineering, Volume 59, Issue 3, pp. 725-769.

Moody D., Kortink M., (2000). *From Enterprise Models to Dimensional Models : A Methodology for Data Warehouse and Data Mart Design*. DMDW'00 Suède.

Prat N., Akoka J., Isabelle Comyn-Wattiau (2006). A UML-based data warehouse design method. *Decision Support Systems*, vol. 42, pp. 1449-1473.

Ouaret, Z., Bellatreche L. et Boussaid O. (2007). *XUML Star : Conception d'un entrepôt de données XML*, Atelier des Systèmes d'Information Décisionnels, Sousse – Tunisie.

Romero O., Abello A. (2007). *Automating Multidimensional Design from Ontologies*. *DOLAP'07,* Lisboa, Portugal. Copyright ACM 978-1-59593-827-5/07/0011.

Soussi A., Feki J. Gargouri F. (2005). *Approche Semi-automatisée de Conception de Schémas Multidimensionnels Valides*, *Revue des Nouvelles Technologies de l'Information* (RNTI), Ed. Cépaduès, vol. n° B-1, pp. 71-90.

Teste O. (2000). *Modélisation et manipulation d'entrepôts de données complexes et historisées* . Thèse de doctorat, Université Paul Sabatier (Toulouse III).

Zepeda L., Celma M., Zatarain R. (2005). *A Methodological Framework for Conceptual Data Warehouse Design*. 43rd ACM Southeast Conference, Mars 18-20, Kennesaw, GA, USA. Copyright ACM 1-59593-059-0/05/0003.

Zribi S., Feki J. (2007). Du *diagramme de classes UML au multidimensionnel*, Atelier des Systèmes Décisionnels (ASD'07), Sousse, Tunisie.

Summary

The data warehouse is a decision support system component based on the federation of information issued from various departments of an organization. The data warehouse design steps cover the conceptual, logical and physical modelling. In this paper, we address the conceptual modelling. More precisely, we propose an automatic, ascending approach for the design of multidimensional schemas starting from an object database. This approach extracts the multidimensional concepts (facts, measures and dimensions) by means of a set of heuristics we define. These heuristics explore the relationships between the objects and the types of the attributes in objects; they produce a data mart star schema that may be validated by the decisional designer.

Vers une réutilisation orientée langage naturel de patrons multidimensionnels

Ines Ben Messaoud, Jamel Feki
Laboratoire Mir@cl
Département d'Informatique, Faculté des Sciences Economiques et de Gestion de Sfax, Route de l'Aéroport Km 4 – 3018 Sfax, BP. 1088
bm_iness@yahoo.fr ; jamel.feki@fsegs.rnu.tn

Résumé. La réutilisation conceptuelle est introduite dans le domaine décisionnel par le biais du concept de patron multidimensionnel (PM) représenté graphiquement par un schéma en étoile selon le formalisme DFM (« Dimensional Fact Model ») de Golfarelli. Cet article propose une méthode originale de réutilisation d'un PM, elle masque la structure multidimensionnelle d'un PM en générant des requêtes types en langage quasi-naturel. Pour réutiliser un PM, le décideur sélectionne parmi ces requêtes générées celles qui traduisent au mieux ses besoins analytiques spécifiques. La volumétrie des requêtes types générées est réduite en recourant essentiellement à l'orthogonalité des dimensions d'un PM.

1 Introduction

Les systèmes d'information décisionnels (SID) constituent un thème qui intéresse à la fois les décideurs et les chercheurs. D'une part, les décideurs y trouvent une solution prometteuse pour leurs problèmes stratégiques et tactiques en accélérant/facilitant le processus de prise de décisions. D'autre part, les chercheurs sont intéressés par le processus de conception et de développement de ces SID. Généralement, ces SID sont construits autour d'un entrepôt de données (ED) regroupant toute l'information utile à la prise de décision. Les travaux de recherche en entreposage de données (« Data warehousing ») ont touché les trois niveaux d'abstraction habituels : Conceptuel, logique et physique. Les approches proposées sont de trois types ; elles couvrent les méthodes dites descendantes (Kimball 1977), ascendantes (Golfarelli et al., 1998) (Moody et al., 2000) (Husemann et al., 2000) et mixtes (Böhnlein et al., 1999) (Phipps et Davis, 2002) (Prat et al, 2006). Certains travaux récents reprennent les approches classiques dans une optique dirigée par les modèles et s'alignent ainsi sur la démarche MDA (« Model Driven Architecture ») initiée par le groupe OMG (« Object Management Group »). MDA vise à automatiser les étapes de développement d'un système en employant des modèles à différents niveaux d'abstraction et des transformations inter et intra modèles. Cependant, dans cet axe, la plupart des travaux récents focalisent sur deux niveaux d'abstraction (i.e., de modèles) et ne touchent que peu le niveau relatif à l'ingénierie des besoins nommé CIMM (« Multidimensional Computation Independent Model »). Dans ce travail nous nous sommes fixés l'objectif de proposer une méthode d'ingénierie des besoins décisionnels. Cette méthode allie deux composants fondamentaux : *i) le concept de patron multidimensionnel* (PM) (Feki et al., 2007) et *ii) le langage naturel*. L'emploie du PM permet de doter la méthode des avantages de la réutilisation et procure consistance et généricité au moment de l'expression des besoins analytiques, héritées des propriétés et contraintes du

patron. Le langage naturel est justifié par le fait qu'il représente une forme non artificielle ; d'ailleurs, la plus simple et la plus compréhensible par les preneurs de décisions.

La conjugaison du langage naturel et des PM permettra de faciliter la collecte/expression des besoins des décideurs. Il s'agit d'adapter un PM aux besoins d'un preneur de décisions et ceci en évitant d'opérer directement sur la structure multidimensionnelle du patron, que nous envisageons masquer. Pour ce faire, nous générons automatiquement des requêtes simples qui décrivent les potentialités analytiques du patron. Le décideur sélectionne alors parmi ces requêtes celles qui traduisent partiellement et au mieux ses besoins pour instancier un PM ; il dérive ainsi un schéma de magasin de données (MD) approprié.

Cet article traite de cette problématique. Il est organisé comme suit : la section 2 présente les modes de représentation des besoins décisionnels les plus populaires de la littérature. La section 3 définit le concept de patron multidimensionnel et décrit brièvement sa démarche de construction. La section 4 présente notre méthode de réutilisation d'un PM par génération et sélection de requêtes analytiques types par le décideur. Finalement, la section 5 synthétise ce travail et présente nos perspectives.

2 Travaux relatifs à l'ingénierie des besoins analytiques

En entreposage de données, l'ingénierie des besoins permet d'identifier les informations décisionnelles à stocker dans l'entrepôt. Elle traite l'identification des objectifs de l'organisation, des décisions qui peuvent être prises pour atteindre ces objectifs et les informations nécessaires pour la prise de décision (Prakash et Gosain 2003). Paradoxalement, il existe des approches d'ingénierie qui ne proposent pas de notation pour représenter les besoins, comme dans (Winter et Strauch 2003), (Winter et Strauch 2004), (Paim et Castro 2003). Dans cette section, nous étudions les représentations les plus pertinentes des besoins.

Dans (Feki 2004) et (Nabli et al. 2005), les auteurs proposent une méthode tabulaire pour l'expression des besoins analytiques des preneurs de décisions. En fait, pour les décideurs un tableau est une forme de présentation familière et intuitive. Pour assister le décideur à spécifier correctement ses besoins, les auteurs font recours à une ontologie décisionnelle (Nabli et al. 2005). Ceci permet de faciliter l'emploi de la forme tabulaire qui demeure délicate si le nombre d'axes d'analyses est élevé.

Les auteurs de (Mazón et al. 2005) catégorisent en premier lieu les besoins des utilisateurs et adoptent la technique i* pour leur représentation. Cette technique permet de construire deux modèles : le modèle *de dépendance stratégique* et le *modèle rationnel stratégique*. La définition de ces deux modèles est basée sur un ensemble de directives, et les modèles stratégiques décrivent convenablement les exigences. En contre partie, les développeurs de l'entrepôt doivent bien comprendre les buts métier de l'organisation afin de définir des directives qui répondent aux besoins des décideurs.

(Gam et Salinesi 2006) catégorisent les besoins des décideurs au cours de leur recensement. Ils adoptent l'approche *orientée but* et plus précisément le *formalisme de la carte*. En fait, ce formalisme permet de représenter conjointement le quoi et le pourquoi ; il représente les buts sous forme d'intention à différents niveaux de détails par le mécanisme d'affinement. Ensuite, les auteurs utilisent le langage naturel pour opérationnaliser les actions à entreprendre par l'utilisateur. En fait, les besoins exprimés en langage naturel repré-

sentent la forme la plus simple pour l'utilisateur. Cependant, lorsque l'utilisateur spécifie ses besoins en langage naturel, il peut probablement employer des styles d'expression différents ; ce qui risque d'engendrer des ambiguïtés d'interprétation/compréhension.

Les auteurs de (Giorgini et al. 2005) et (Giorgini et al. 2008) proposent la *technique orientée but* nommée GRAnD (Goal-oriented Requirement Analysis for Data warehouses), pour l'analyse des besoins utilisées dans la construction de l'entrepôt. Cette technique est basée sur la méthodologie Tropos. Ces auteurs distinguent deux types de modélisation : *organisationnelle* et *décisionnelle*. La représentation de ces diagrammes repose sur une notation bien spécifique et décrit clairement les besoins des utilisateurs. Néanmoins, la représentation de tels diagrammes nécessite une bonne connaissance de la structure organisationnelle de l'entreprise et des besoins des décideurs.

Dans cette section, nous avons examiné très succinctement les travaux qui nous semblent les plus pertinents en ingénierie des besoins analytiques. Nous constatons que chacun de ces travaux se limite à l'usage d'une seule technique à la fois. Nous proposons une solution combinant la réutilisation de patron multidimensionnel avec le langage naturel. Il s'agit plus précisément de masquer la structure relativement complexe du patron en étoile par une interface générant des requêtes analytiques types exprimées en langage quasi-naturel.

3 Aperçu sur le concept de patron multidimensionnel

Dans des travaux antérieurs (Feki et al. 2006) (Feki et al. 2007) (Ben-Abdallah et al. 2008), nous avons introduit le concept de patron multidimensionnel (PM) dans une tentative d'assister, d'une part, le décideur à exprimer facilement ses besoins analytiques en partant d'une solution assez générique et éprouvée pour un problème décisionnel donnée (i.e., sujet d'analyse) et, d'autre part, à renforcer l'automatisation du processus conceptuel d'élaboration de schémas de magasins de données.

En vue d'une meilleure clarté de l'exposé, nous commençons d'abord par rappeler la définition du concept de PM, après quoi nous décrivons brièvement la démarche de sa construction.

3.1 Définition d'un PM

Dans (Feki 2008), un *patron multidimensionnel* est défini comme « une solution conceptuelle pour un problème décisionnel. Il est *orienté sujet, générique, documenté,* et *spécifique à un domaine d'activité de l'entreprise.* Un PM est modélisé par un *schéma en étoile* et conçu pour la spécification des besoins OLAP et la mise en œuvre de magasins de données ».

Un PM décrit un sujet d'analyse (fait) en fonction d'un ensemble d'axes d'analyse (dimensions). Le centre de l'étoile représente un fait contenant une ou plusieurs mesures. Ce fait est relié à des dimensions composées d'attributs. Certains de ces attributs sont organisés en hiérarchies. Les hiérarchies constituent des perspectives d'analyses sur les axes.

Afin d'assister la réutilisation, tous les composants d'un PM (fait, mesure, attributs, dimension, hiérarchie) sont textuellement documentés au moment de sa construction. Cette documentation est accessible via l'interface de réutilisation.

3.2 Construction d'un PM

La démarche de construction d'un PM est orientée par les entités du monde réel (EMR) qui sont en fait des artefacts circulant dans l'entreprise comme une facture, une fiche produit, un écran de saisie ou de résultat, etc. Dans le reste de cet article, ces EMR seront désignées par le terme document. La démarche de construction d'un PM examine un large échantillon de documents appartenant à des entreprises différentes d'un même secteur d'activité. Cette construction s'effectue en trois phases appelées : *Standardisation* des documents, *classification des documents*, et *identification des éléments multidimensionnels* (i.e., fait, mesures, dimensions, attributs, hiérarchies) qui composent le PM.

Ces phases utilisent un ensemble d'heuristiques développées dans (Feki et al. 2007), l'application de la démarche de construction sur un exemple est détaillée dans (Ben-Abdallah et al. 2008).

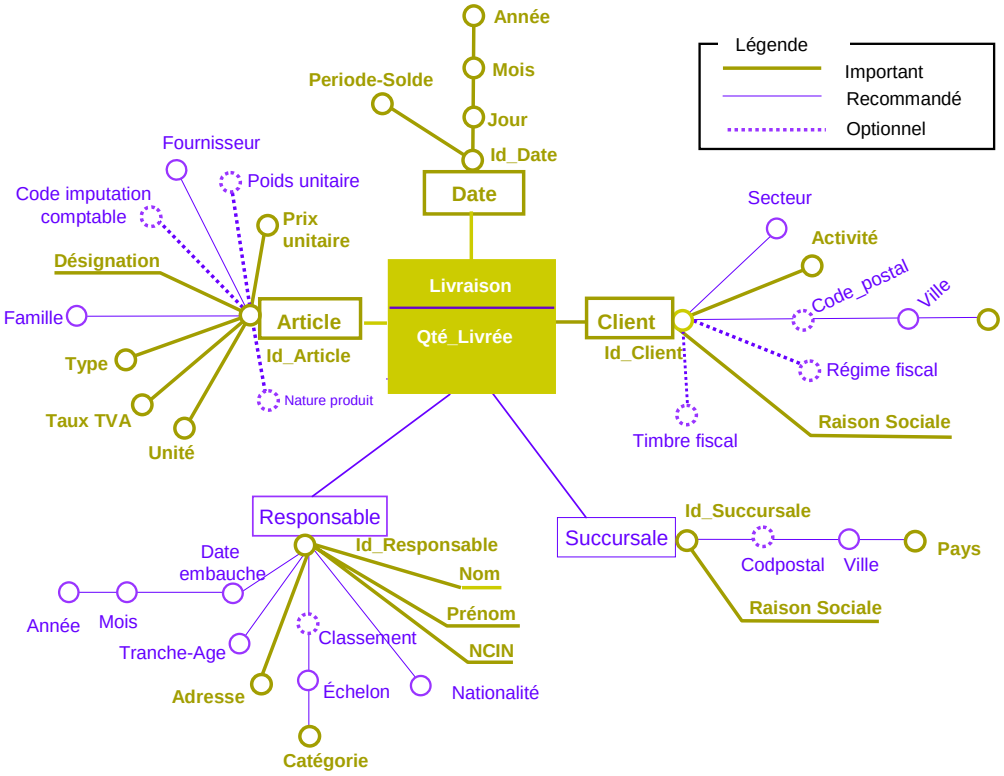


FIG.1- Exemple de patron multidimensionnel analysant le fait Livraison

Particulièrement, la phase de standardisation repose sur une étude empirique menée sur un grand nombre de documents utilisés dans un domaine particulier au sein d'un ensemble d'entreprises. Elle calcule le taux de présence de chaque rubrique (i.e., attribut) dans l'échantillon de documents étudiés. Nous utilisons ce taux comme un indice de généricité qui

mesure le potentiel analytique des rubriques retenues pour la construction du patron. Afin d'assister la réutilisation d'un patron, cette démarche classe les éléments multidimensionnels d'un patron en trois niveaux d'importance *Important*, *recommandé*, et *optionnel.* De plus, elle les distingue dans la représentation graphique du PM (cf. Figure 1). Les éléments *important* constituent le noyau dur du patron, c'est-à-dire, sa partie la plus stable qui ne nécessite pas beaucoup d'efforts d'adaptation pour être réutilisée dans une entreprise particulièr. Cependant, les éléments marqués *recommandé* et *optionnel* ne sont pas présents dans la quasi-totalité des systèmes d'information opérationnels selon les résultats de l'étude empirique ; en conséquence, ils représentent la partie variable du patron, c'est-à-dire, la partie la plus susceptible d'être adaptée en phase de réutilisation.

Par ailleurs, notre démarche de construction produit un PM contraint (Ben Abdallah et al. 2008). Les contraintes définies sur un PM sont des règles de contrôles permettant de vérifier sa validité structurelle ainsi que la validité des données qui seront alimentées dans le magasin dérivé à partir du PM. Nous nous basons sur ces contraintes, que nous considérons comme des propriétés du PM, afin de produire des requêtes types simples et, en conséquence faciliter la réutilisation du patron.

Parmi ces contraintes, nous énumérons principalement trois qui intéressent le présent travail : *l'orthogonalité* des dimension, *l'acyclicité* des hiérarchies et *la racine hiérarchiqu*e d'une dimension.

- L'orthogonalité des dimensions exige que deux attributs quelconques appartenant à deux dimensions distinctes ne soient pas en dépendance fonctionnelle (Lechtenbörger et al., 2003). Cette contrainte justifie bien que les requêtes générées peuvent être monodimensionnelle, c'est-à-dire, que chacune porte sur une seule dimension et qu'il est inutile de générer des requêtes complexes.
- L'acyclicité contrôle l'absence de cycles dans une hiérarchie. Elle exige q'un paramètre ne puisse être père et fils du même paramètre par transitivité (Hutardo et al., 2002). Nous exploitons cette contrainte pour affirmer que chaque paramètre ne doit pas exister plus qu'une fois dans une même requête.
- La racine hiérarchique contrôle que toutes les hiérarchies d'une dimension *d* aient en commun le même paramètre qui est l'attribut le plus fin ; ce paramètre est l'identifiant de *d*. Nous veillons à ce que toutes requête générée doit nécessairement inclure l'identifiant de la dimension afin d'éviter une instanciation de dimensions ou de hiérarchies ne contenant pas la racine hiérarchique

Finalement, le processus de réutilisation consiste à dériver un schéma de MD à partir d'un PM et commence par une étape de *pré-instanciation* qui a pour but d'adapter le PM aux besoins analytiques des décideurs

Rappelons que notre objectif dans cet article est de proposer une méthode simple pour réutiliser un PM par un décideur non informaticien. Plus précisément, nous nous intéressons à la pré-instanciation du patron. La section suivante introduit et détaille notre méthode proposée ; elle se base sur la génération des requêtes en langage quasi-naturel à partir d'un PM et ceci en se basant sur les contraintes ci-dessus et les niveaux de généricité des éléments du patron.

4 Méthode de pré-instanciation proposée

L'étape de pré-instanciation consiste à modifier le PM pour l'adapter à un ensemble de besoins analytiques. Suite à cette étape, seuls les éléments multidimensionnels pertinents pour réaliser ces besoins sont retenus ; les autres seront systématiquement retirés du PM. Notre méthode proposée vise à masquer la structure multidimensionnelle, relativement assez complexe, du PM en phase de sa réutilisation et ceci par la génération de requêtes types exprimées en langage quasi-naturel. Le langage naturel présente plusieurs avantages ; entre autres, c'est une forme native et compréhensible par et pour les décideurs. En phase de pré-instanciation, le rôle du décideur consiste à sélectionner les requêtes types qui décrivent ses perspectives d'analyse. La Figure 2 montre l'enchaînement des étapes de cette préinstanciation.

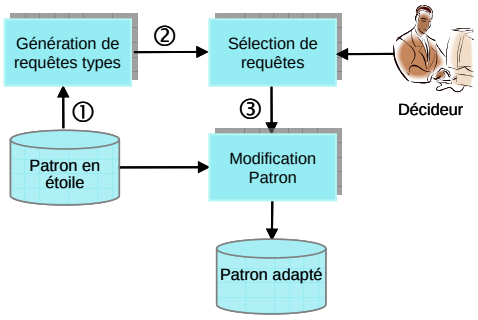


Fig. 2 – Pré-instanciation d'un patron.

4.1 Principe de la méthode

Nous proposons une méthode qui se base sur la génération de requêtes exprimées sur les éléments multidimensionnels du patron (i.e., mesures, dimensions, paramètres...). Vu la combinatoire élevée de ces éléments, les requêtes que nous pouvons générer sont très redondantes (i.e., avec des éléments qui se répètent) et en nombre certainement important. Pour éviter ces inconvénients, nous exploitons la propriété d'orthogonalité (cf. section 3.2) des dimensions afin de réduire le nombre de requêtes et les simplifier, c'est-à-dire, éviter des requêtes complexes exprimées sur plusieurs dimensions à la fois. En se basant sur cette propriété, notre méthode de génération de requêtes types assemble en plus des mesures, les paramètres et les attributs faibles appartenant à une hiérarchie d'une dimension.

Nous définissons la structure d'une requête type élémentaire (RTE) comme suit :

$RTE = (Id^{RTE}, Nom^{P}, MES^{RTE}, Nom^{d}, Nom^{h}, PARAM, Gen)$ avec :

- Id^{R} : identifiant d'une requête type,
- Nom^{P} : nom du patron en étoile (ou de son Fait),
- $MES = \{m_1, m_2, ..., m_m\}$ est un sous ensemble des mesures du fait du patron,
- Nom^{d} : nom d'une dimension (axe d'analyse) du patron,
- Nom^{h} : nom d'une hiérarchie,
- $PARAM = [p_1 < p_2 < ... < p_p]$ est une liste ordonnée de paramètres de la hiérarchie Nom^{h} avec p_1 est le paramètre le plus fin et $p_i < p_j$ signifie que p_i détermine fonctionnellement p_j, et
- *Gen* : niveau de généricité (important, recommandé, optionnel) de la requête type.

Par définition, chacune des requêtes types élémentaires permet de répondre à un besoin simple ; ceci n'exclue pas qu'un ensemble de requêtes élémentaires retenues permettra de répondre à des besoins complexes par composition. Par exemple, à partir des deux requêtes élémentaires indépendantes *R1* « *Total des quantités vendues par article »* et *R2 « Total des quantités vendues par client »* exprimées sur un même PM, nous pouvons répondre à une requête plus élaborée pour déterminer le « *Total des quantités vendues par article et par client* ».

Plus généralement, si Re = {R_1, R_2 …, R_n } est un ensemble de *n (n>1)* requêtes élémentaires définies sur des dimensions quelconques d'un même patron *P*, alors nous pouvons répondre à une requête complexe *Rc* sur *P,* telle que l'ensemble des mesures de *Rc* est l'union des ensembles de mesures des requêtes élémentaires *Ri (1≤n),* et l'ensemble des dimensions de *Rc* est l'union de l'ensemble des dimensions de ces mêmes requêtes *Ri.*

4.2 Génération de requêtes types pour un PM

Afin de simplifier la pré-instanciation d'un PM, nous commençons premièrement par générer l'ensemble des requêtes qui sera proposé au décideur pour y sélectionner les axes d'analyses (dimensions) qui l'intéresse (cf. algorithme 1). Puis pour chaque axe d'analyse retenu, des requêtes types seront générées (cf. algorithme 2). Ceci évite de générer des requêtes inutiles sur des axes non intéressants et optimise en conséquence le processus de pré-instanciation.

Vu les divergences des styles d'écriture du langage naturel et les ambiguïtés qui en découlent, nous avons fixé une grammaire pour produire des requêtes simples et compréhensibles. La figure 3 montre la syntaxe de cette grammaire.

Besoin ::= *Analyser* Indicateur *par* Nom_DIM {selon | durant} GN_PARAM
Indicateur ::= $mesure_1$, $mesure_2$, …, $mesure_m$
GN_PARAM ::= $paramètre_1$, $paramètre_2$, … et $paramètre_p$
Nom_DIM ::= texte
Mesure ::= texte
Paramètre ::= texte

Fig. 3 – Grammaire pour la génération de requêtes élémentaires.

Dans la suite de cet article , nous adoptons les notations suivantes :

- Nom^F : Nom du fait d'un PM.
- MESURE : {m_1, m_2, …, m_m} l'ensemble des mesures du fait d'un PM.
- DIM = {d_1, d_2, …, d_d} l'ensemble des dimensions d'un PM .
- DIM_r : sous ensemble de dimensions de DIM, retenu par l'utilisateur.
- HIER = {h_1, h_2, …, h_h} l'ensemble des hiérarchies appartenant à une dimension.
- PARAM = [p1<p2<…<pp] une liste de paramètres ordonnée d'une hiérarchie.
- ATTF(p) : fonction retournant l'ensemble {a_1, a_2, …, a_a} des attributs faibles associés à un paramètre *p*.
- Gen = {important, recommandé, optionnel} le niveau de généricité de la requête type.
- Gen(p) : fonction retournant la généricité d'un paramètre *p*.

L'algorithme suivant génère les requêtes pour sélectionner les dimensions à retenir.

Algorithme 1 :

```
// Génération des requêtes pour le sélection des dimensions
Pour chaque di ∈DIM faire
   RTD = 'Analyser' + NomF + 'par' + di
   Afficher RTD
Fin pour
```

Pour le PM de la Figure 1, l'application de cet algorithme produit les cinq requêtes du tableau 1. Le décideur intervient pour cocher celles qui intéressent ses besoins. Notons que la documentation du patron peut être consultée pour expliquer la signification de chacune des dimensions du PM.

- ☐ Analyser Livraison par date.
- ☐ Analyser Livraison par article.
- ☐ Analyser Livraison par responsable.
- ☐ Analyser Livraison par succursale.
- ☐ Analyser Livraison par client.

TAB. 1- Requêtes types générées pour la sélection des dimensions.

Après sélection des requêtes exprimées sur les dimensions, le patron est allégé en supprimant les dimensions inutiles. Ensuite, sa réutilisation se poursuit par l'examen des requêtes sur les hiérarchies.

L'algorithme 2 génère des requêtes significatives pour chaque hiérarchie de chaque dimension retenue par l'utilisateur.

Algorithme 2 :

```
// Génération des requêtes types élémentaires pour la définition des hiérarchies
Pour chaque di ∈DIMr faire
   Ind = 'selon'
   RTE = 'Analyser' + MESURE + 'par'
   Si di est une dimension temporelle alors
       Ind = 'durant'
   Finsi
   Pour chaque hj∈di.HIER faire
       Pour chaque pk∈hj.PARAM faire
             RTE = RTE + Ind   + pk
             Gen = Gen(pk)
             Si ATTF(pk)≠ Ø alors
             RTE = RTE + '(' + ATTF(pk) + ')'
             Finsi
             Afficher RTE
       Fin pour
   Fin pour
Fin pour
```

Pour le patron de notre exemple (Figure 1), et en se limitant à la dimension *Client* (supposée retenue dans l'étape précédente), le tableau 2 montre des exemples de requêtes types générées.

	Requête	**Généricité**
☑	Analyser Qté_Livrée par client (Raison Sociale) selon Activité.	Important
☑	Analyser Qté_Livrée par client (Raison Sociale) selon Secteur.	Recommandé
☑	Analyser Qté_Livrée par client (Raison Sociale) selon Ville.	Recommandé
☑	Analyser Qté_Livrée par client (Raison Sociale) selon Pays.	Important
☐	Analyser Qté_Livrée par client (Raison Sociale) selon Code_postal.	Optionnel
☐	Analyser Qté_Livrée par client (Raison Sociale) selon Régime fiscal.	Optionnel
☐	Analyser Qté_Livrée par client (Raison Sociale) selon Timbre fiscal.	Optionnel
☐	…	

TAB. 2- Requêtes types pour la dimension Client.

En supposant que l'utilisateur décisionnel a retenu les quatre requêtes cochées du tableau 2, la pré-instanciation partielle du patron (i.e., réduite à la dimension *Client*) est montrée par la figure 4 où tous les éléments sont ramenés au même niveau d'importance.

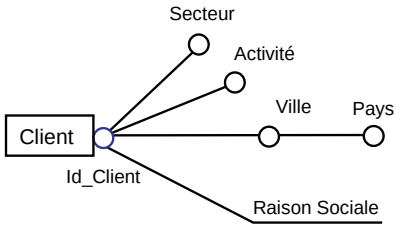


Fig. 4 – Dimension Client pré-instanciée.

De manière analogue, le traitement de toutes les dimensions retenues produira un schéma de magasin de données en étoile.

5 Conclusion

Le travail présenté dans cet article s'inscrit dans le domaine de l'ingénierie des besoins pour les systèmes d'information décisionnels. Particulièrement, nous avons ré-exploité le concept de patron multidimensionnel (PM) en tant qu'outil pour l'expression des besoins analytiques. Afin de présenter une méthode de réutilisation simple, nous avons conjugué la puissance du patron avec l'expressivité du langage naturel pour masquer la structure multidimensionnelle assez complexe d'un PM. Le résultat est une méthode compréhensible par l'utilisateur décisionnel qui manque souvent de culture dans le domaine d'entreposage de données et plus spécifiquement en concepts multidimensionnels. Par ailleurs, notre méthode a le mérite de construire un schéma de MD de manière incrémentale et ceci en commençant

par la sélection des dimensions puis, par celle des requêtes types exprimées sur les hiérarchies des dimensions retenues ; d'où une optimisation de la méthode.

Nous continuons à améliorer cette méthode selon diverses perspectives ; par exemple, réduire le nombre de requêtes en regroupant plusieurs paramètres puis, classer les requêtes types générées par niveau de potentiel analytique en vue de les privilégier pendant la pré-instanciation. Ceci, nécessiterait l'introduction d'une métrique de calcul du potentiel analytique d'une requête. Aussi, il est intéressant de limiter le nombre de paramètres dans une requête générée sur une hiérarchie de dimension ayant un grand nombre de niveaux.

Références

Ben Abdallah M., Ben Saïd N., Feki J. and Ben Abdallah H. (2007). *MP-BUILDER: A tool for multidimensional pattern construction*. International Arab Conference on Information Technology (ACIT'07), Lattakia, Syria.

Ben Abdallah M., Feki J. et Ben-Abdallah H. (2008). *Patrons multidimensionnels constraints*. conférence Internationale sur les Systèmes d'Information et Intelligence Économique (SIIE'2008), pp. 237-250, Hammamet, Tunisie.

Ben-Abdallah H., Feki J. et Ben-Abdallah H. (2008). *A Multidimensional Pattern based Approach for the Design of Data Marts*, Progressive Methods in Data Warehousing and Business Intelligence: Concepts and Competitive Analytics", Volume 3 of the Advances in Data Warehousing and Mining Series, Ed. David Taniar (à paraître 2008).

Feki J. (2008). *Approches de modélisation conceptuelle de systèmes d'information décisionnels,* Rapport d'Habilitation Universitaire, Université de Sfax, Tunisie.

Feki J., Ben-Abdallah H. (2007). Multidimensional Pattern Construction and logical Reuse for the Design of Data Marts. *International Review on Computers and Software* (I.RE.CO.S).

Feki J. (2004). *Vers une conception automatisée des entrepôts de données : Modélisation des besoins OLAP et génération de schémas multidimensionnels*. Maghrebian Conference on Software Engineering and Artificial Intelligence. Sfax, Tunisie.

Feki J., Ben-Abdallah H., Ben-Abdallah M. (2006). *Réutilisation des patrons en étoile*. XXIVème Congrès Informatique des organisations et systèmes d'information (INFORSID'06), pp 687-701, Hammamet, Tunisie.

Gam I., Salinesi C. (2006). *Analyse des exigences pour la conception d'entrepôts de données.* Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID'06), Hammamet, Tunisie.

Giorgini P., Rizzi S., Garzetti M. (2005). *Goal-oriented requirement analysis for data warehouse design.* Proceedings of the 8th ACM international workshop on Data warehousing and OLAP (DOLAP '05).

Giorgini P., Rizzi S., Garzetti M. (2008). *GRAnD: A goal-oriented approach to requirement analysis in data warehouses.* Decision Support Systems.

Golfarelli M., Maio D., Rizzi S., (1998). *Conceptual design of data warehouses from E/R schemes*. 31st Hawaii International Conference on System Sciences.

Hurtado C.A., Mendelzon A.O. (2002). *OLAP Dimension Constraints*. Dans 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02), Madison, USA, pp. 169-179.

Husemann B., Lechtenbörger J., Vossen G., (2000). *Conceptual Data Warehouse Design. Proc*. Of the Int'l Workshop on Design and Management of Data Warehouses Stockholm Suède, pp. 6.1-6.11.

Kimball R., (1997) *The Data Warehouse Toolkit*, John Wiley and Sons Inc.

Lechtenbörger J., Vossen G. (2003). *Multidimensional normal forms for data warehouse design*. Dans Revue Information Systems, Vol. 28, N. 5, pp. 415-434.

Mazón J.N, Trujillo J., Serrano M., Piattini M. (2005). *Designing Data Warehouses: From Business Requirement Analysis to Multidimensional Modeling.* Proceedings 1st International Workshop on Requirements Engineering for Business Need and IT Aligment (REBNITA'05).

Moody D., Kortnik M., (2000). *From Enterprise Models to Dimensional Models : A Methodology for Data Warehouse and Data Mart Design*. DMDW'00 Suède.

Nabli A., Feki J., Gargouri F. (2005). *Automatic Construction of Multidimensional Schema from OLAP Requirements.* Arab International Conference on Computer Systems and Applications (AICCSA'05), Cairo, Egypt.

Paim F.R, Castro J.B (2003). *DWARF: An Approach for Requirements Definition and Management of Data Warehouse Systems.* 11th IEEE Iternationnal Requirements Engineering Conference (RE'03) Monterey Bay, California, USA.

Prakash N., Gosain A. (2003). *Requirements Driven Data Warehouse Development.* CAiSE Short Paper Proc.

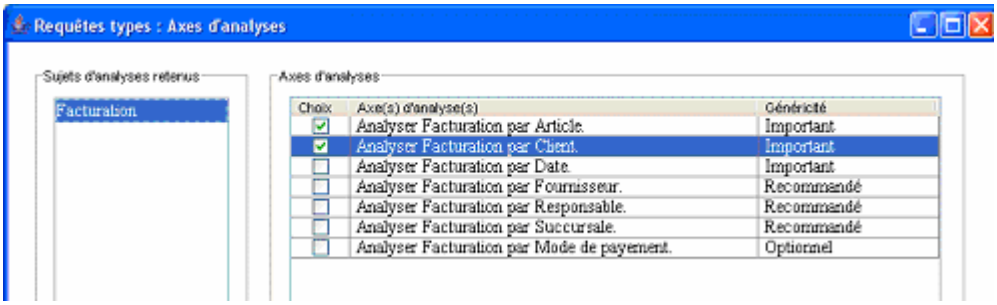
Winter R., Strauch B. (2003). *A Method for Demand-driven Information Requirements Analysis in Data Warehousing Projects.* Proceeding of the 36th Hawaii International Conference on System Sciences.

Winter R., Strauch B. (2004). *Information Requirements Engineering for Data Warehouse Systems.* ACM Symposium on applied Computing (SAC'04).

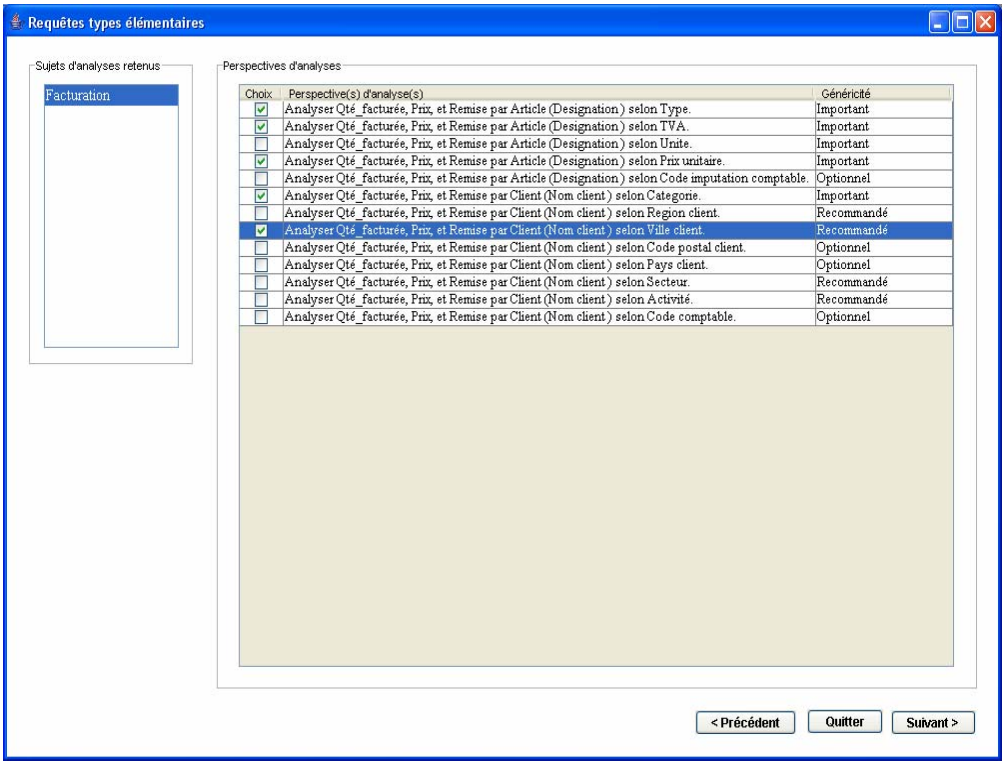
Summary

The design reuse is introduced into the decisional field by means of the concept of multidimensional pattern (MP) represented graphically by a star schema accordingly to the Golfarelli DFM (Dimensional Fact Model). This paper proposes an original reuse method for MP; it hides the multidimensional structure of a MP by generating natural language, standard queries. To reuse a MP, the decision maker selects among these generated queries those which translate their specific analytical needs. The volume of the generated standard queries reduced mainly thanks to the orthogonality of dimensions.

Annexe



Exemples de requêtes types générées pour la sélection des dimensions du fait Facturation.



Exemples de requêtes types pour les deux dimensions Article et Client.

Une Approche Automatique de Vérification de Schéma de Hiérarchie

Ali Salem[1], Hanene Ben-Abdallah[1], Faiza Ghozzi[2]
Laboratoire Mir@cl
[1] *Faculté des Sciences Economiques et de Gestion de Sfax, Tunisie*
[2] *Institut Supérieur d'Informatique et de Multimédia de Gabès, Tunisie*
al_salemfr@yahoo.fr,hanene.benabdallah@ fsegs.rnu.tn, Jedidi_Faiza@yahoo.fr

Résumé : Nous proposons une approche générique de vérification des contraintes structurelles des hiérarchies de modèles multidimensionnels. Cette approche se base sur la spécification formelle en Z du concept hiérarchie au niveau méta-modèle et la traduction de modèles particuliers vers Z. Pour assister les non-experts en Z, notre approche extrait automatiquement à partir de la base du démonstrateur de théorème Z\eves et génère à partir d'un modèle d'hiérarchie donné les axiomes nécessaires à introduire durant la preuve de la bonne formation de la hiérarchie. De plus, elle les introduit à travers les commandes nécessaires pour mener la preuve automatiquement.

1 Introduction

Dans les systèmes de traitements analytiques en ligne, reconnus par les systèmes OLAP, les données sont souvent stockées dans des bases de données multidimensionnelles. Ces données sont organisées par centre d'intérêt (Fait) et étudiées en fonction de différents axes (Dimensions) et perspectives (Hiérarchies) (Kimball et Ross, 2002).

Par ailleurs, comme pour tout autre système, la qualité d'un système OLAP dépend étroitement de la qualité de son modèle conceptuel. Parmi les critères de qualité primordiaux de ces modèles, on retrouve leur bonne formation (structurelle et sémantique). C'est dans ce cadre que s'inscrivent les travaux présentés dans cet article, qui visent le développement d'un framework pour la vérification formelle de la bonne formation des modèles multidimensionnels.

Pour ce faire, nous avons commencé par collecter à partir de la littérature les contraintes qui assurent aussi bien la conformité d'un schéma multidimensionnel aux concepts multidimensionnels (niveau méta-modèle), que la cohérence des données alimentant le schéma (niveau instance), *cf.,* (Hurtado et Mendelzon, 2002), (Lechtenbörger et Vossen, 2003), (Ghozzi et al., 2003). En particulier, dans nos travaux antérieurs nous nous sommes intéressés à la bonne formation du concept hiérarchie pour lequel nous avons formalisé en Z (Spivey, 1992) ses composants ainsi que les contraintes structurelles et d'alimentation (instances) (Salem et al., 2008). Dans cet article, nous complétons ces travaux par une démarche automatique et générique de vérification des contraintes structurelles.

Notre démarche repose, d'une part, sur la formalisation du concept de hiérarchie (méta-modèle), et d'autre part, la traduction de hiérarchies (modèles particuliers) vers Z. Pour assister les non-experts en Z, notre approche automatiquement extrait à partir de la base du démonstrateur de théorème Z/eves et génère, à partir d'un modèle de hiérarchie donné, les axiomes nécessaires à introduire durant la preuve de la bonne formation de la hiérarchie. De

plus, elle les introduit à travers les commandes nécessaires pour mener la preuve automatiquement.

Le reste de cet article est organisé comme suit. La section 2 commence par passer en revu l'état de l'art sur les schémas multidimensionnels contraints ; puis, elle résume les contraintes structurelles pertinentes au concept hiérarchie. La section 3 présente la formalisation du concept hiérarchie contraint. La section 4 introduit notre environnement de vérification et présente notre démarche de vérification de la bonne formation de hiérarchies. La section 5 clôture l'article par un résumé et des perspectives.

2 Etat de l'art

Au cours de notre étude des travaux antérieurs dans ce domaine, nous nous sommes intéressés aux contraintes intégrées dans les différents concepts de base des modèles multidimensionnels. Dans cette section, nous résumons les propositions de la littérature, ensuite nous présentons informellement les contraintes structurelles pertinentes au concept hiérarchie.

2.1 Schémas multidimensionnels contraints

Parmi les propositions de contraintes pour les schémas multidimensionnels, nous distinguons le modèle GMD (Franconi et Kamble, 2004) où les auteurs présentent un ensemble de contraintes dans deux catégories : les contraintes liées à la création des cubes et les contraintes liées à l'agrégation. D'autre part, (Abelló et al., 2006) présente un modèle multidimensionnel orienté objet conçu en UML, dans lequel les contraintes multidimensionnelles sont classifiées en deux autres catégories : les contraintes dites d'emplacement pour la construction du cube et les contraintes d'agrégation. (Lujàn et al., 2002) présente une extension d'UML qui permet de représenter les propriétés structurelles des modèles multidimensionnels au niveau conceptuel.

Par ailleurs, dans (Hurtado et Mendelzon, 2002), nous retrouvons une proposition d'un ensemble de contraintes pour résoudre le problème d'agrégation : les contraintes liées aux dimensions, plus précisément à la structuration hiérarchique des attributs des dimensions et les contraintes liées aux instances d'une dimension. De sa part, (Ghozzi et al., 2003) présentent un modèle multidimensionnel en constellation avec une classification assez claire de contraintes multidimensionnelles.

Face à cette multitude de propositions, dans (Salem et al., 2008), nous avons commencé à harmoniser les contraintes exprimées dans les différents travaux. Nous avons proposé une spécification formelle en langage Z intégrant toutes les contraintes indispensables pour maintenir la cohérence du schéma de hiérarchie. Ces contraintes regroupent les contraintes structurelles et d'alimentation (ou d'instances). Nous avons validé cette spécification à l'aide de l'outil Z\Eves (Saaltink, 1999). Cependant, pour un concepteur non expert en langage Z, la vérification d'un schéma en se basant sur cette spécification est une mission impossible. Il est, alors, nécessaire de lui offrir un moyen plus pratique pour la vérification de son schéma. Ainsi, notre objectif est de proposer une approche générique et automatique de vérification de schéma.

2.2 Contraintes structurelles des hiérarchies

Les contraintes structurelles décrivent essentiellement les règles d'ordonnancement des attributs dans une hiérarchie :

- Unicité de l'identifiant (Ghozzi et al., 2003). Par exemple, dans la dimension *Fournisseur* (Fig 2.1) nous pouvons trouver plusieurs fournisseurs dont chacun possède son propre identifiant. L'existence de plusieurs fournisseurs ayant le même identifiant engendre des ambiguïtés au niveau de l'analyse des données.

- L'identifiant est l'attribut de granularité la plus fine (Ghozzi et al., 2003). Dans une hiérarchie, les attributs sont classés selon un ordre reflétant leur dépendance fonctionnelle. Par exemple, dans la hiérarchie de *Fournisseur* (Fig 1), *ID* détermine *Ville*, *Ville* détermine *Pays*, etc. Seulement ID peut déterminer toutes les informations liées à un fournisseur.

- Unicité de l'attribut *All* (Ghozzi et al., 2003). Il est défini pour clôturer une hiérarchie et permettre une agrégation maximale. Il est, donc, de plus haute granularité. Puisque l'attribut All sert à clôturer une hiérarchie, ID est l'attribut de granularité la plus fine et les attributs sont classés de la granularité la plus fine à la granularité la plus haute, on déduit alors que All est l'attribut de plus haute granularité. Les paramètres de la dimension forment un treillis (Fig 1).

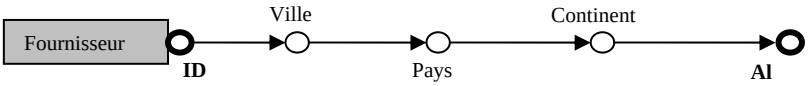


Fig 1 - Exemple de hiérarchie pour la dimension *Fournisseur*

- Hiérarchie non vide (Ghozzi et al., 2003) : chaque hiérarchie possède au moins deux niveaux de paramètres ; *ID* et *All*.

- Acyclicité (Franconi et Kamble, 2004) (Abelló et al., 2006) (Hurtado et Mendelzon, 2002) (Carpani et Ruggia, 2001) (Ghozzi et al., 2003) : elle interdit l'existence d'un cycle entre les attributs de la hiérarchie. Par exemple, dans la version modifiée de la dimension *Fournisseur* (Fig 2), le cycle d'une part implique une dépendance fonctionnelle circulaire entre *Ville* et *Continent,* et d'autre part il engendre un forage infini lors de l'analyse des données.

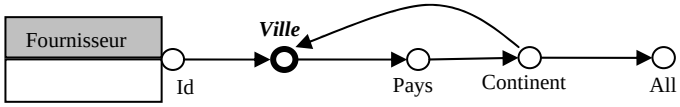


Fig 2 - Un contre exemple de l'acyclicité

- Connexion vers le haut (Hurtado et Mendelzon, 2002) (Ghozzi et al., 2003) : cette contrainte impose que tous les paramètres, sauf *All*, possèdent au moins un père (un paramètre de granularité moins fine).

3 Formalisation en Z et validation du concept de hiérarchie

La spécification formelle du méta-modèle (du concept de hiérarchie) permet d'exprimer les contraintes de manière exacte et précise offrant aussi le moyen de les vérifier. Le langage de spécification choisi est le langage Z. Celui-ci est basé sur la théorie des ensembles et la logique de prédicats (Spivey, 1992). La théorie des ensembles utilisée inclut les opérateurs standard des ensembles, les produits cartésiens et les ensembles de puissance. La logique de prédicats est une logique du premier ordre. Ces deux théories offrent un pouvoir d'expression suffisamment puissant pour spécifier les concepts des modèles multidimensionnels.

Un schéma Z se compose de deux parties : une partie pour la déclaration des constituants du concept spécifié et une partie pour les prédicats représentant les contraintes sur les constituants déclarés.

Avant d'introduire la formalisation du concept dimension, introduisons quelques définitions utiles. Nous commençons notre spécification par définir les deux types *[NomH, Dom]* désignent les ensembles de noms des hiérarchies et de valeurs d'attributs ; le type libre *Type* indique la classe des attributs des dimensions :

$$Type ::= Faible \mid Parametre \mid ID \mid All$$

La relation *Determine* indique la dépendance fonctionnelle entre les attributs :

$$\mid Determine: AttDim \leftrightarrow AttDim$$

La fonction *Poids* affecte à chaque attribut un type : $Poids: AttDim \rightarrow Type$

Chaque attribut d'une dimension comprend un ensemble fini de valeurs (*Dom*). Nous le définissons comme un type Z composé :

AttDim

$$val: \mathbb{F}\ Dom$$

La définition formelle d'une hiérarchie en langage Z se traduit par le schéma nommé *Hierarchie* où : *N* est le nom de la hiérarchie ; *Att* est un ensemble fini d'attributs de dimension appartenant à *AttDim* ; *ParamH* est une séquence décrivant la hiérarchie des attributs. (Une séquence, en langage Z, peut être considérée comme une fonction dont le domaine est un sous ensemble contigu des nombres naturels.)

Hierarchie

N: *NomH*
Att: $\mathbb{F}$ *AttDim*
ParamH: seq *AttDim*

---

(1) $\#ParamH \geqslant 2$
(2) $\exists_1 x: Att \bullet Poids\ x = ID$
(3) $\exists_1 x: Att \bullet Poids\ x = All$
(4) ran $ParamH = Att \setminus \{ y: Att \mid y\ .\ poids = Faible \}$
(5) $\forall x: Att \bullet x \in \text{ran}\ ParamH \land ParamH\ 1 = x \Rightarrow Poids\ x = ID$
(6) $\forall x: Att \bullet x \in \text{ran}\ ParamH \land ParamH\ (\#ParamH) = x \Rightarrow Poids\ x = All$
(7) $\forall i, j: 1\ ..\ \#ParamH \mid i \neq j \bullet ParamH\ i \neq ParamH\ j$
(8) $\forall x, y: Att; i: 1\ ..\ \#ParamH - 2 \mid$
$ParamH\ i = x \land ParamH\ (i + 1) = y$
$\bullet (x, y) \in Determine$

Nous avons validé cette spécification du méta-modèle du concept de hiérarchie à l'aide de l'outil Z\Eves, dans Salem et al. (2008).

4 Approche automatique de vérification d'hiérarchie

Dans cette section, nous proposons notre approche basée sur une démarche formelle de vérification. L'idée consiste en la définition d'une démarche de vérification générique. Ensuite, d'implanter un outil qui permet d'appliquer automatiquement cette démarche (Figure 3).

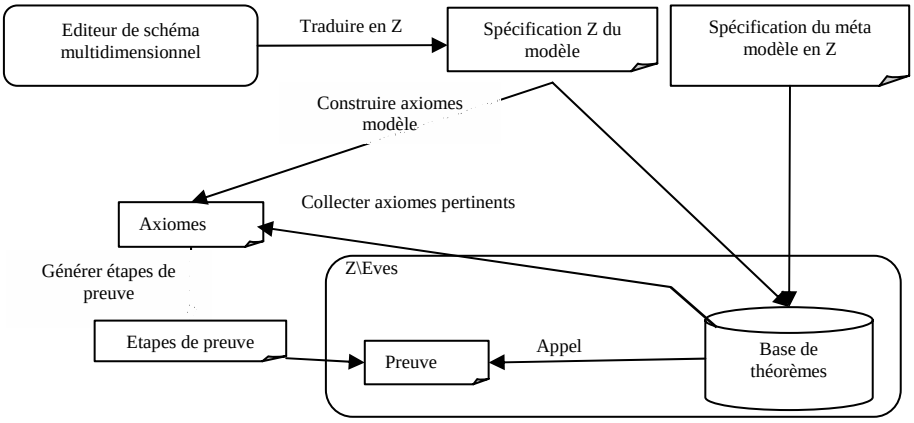


Fig. 3 - Environnement de vérification

Dans un premier temps, le concepteur dessine son schéma dans un éditeur graphique. Ensuite, nous regroupons la spécification formelle du méta modèle, la traduction en Z du schéma dessiné et nous introduisons la spécification complète au démonstrateur de théorème Z\eves. Celui-ci construit automatiquement une base de théorèmes qui comprend : des

axiomes pertinents à la spécification du méta-modèle, et des axiomes représentant les constituants du schéma introduit par le concepteur.

D'autre part, nous construisons des axiomes nécessaires pour la vérification du schéma introduit. Dans cette étape nous devons respecter le fait que ces axiomes soient cohérents (en terme de syntaxe et numérotation) avec ceux construits par l'outil Z\Eves. Enfin, notre assistant génère automatiquement les étapes de vérification (les commandes) en faisant appel aux axiomes que nous avons construits et aux axiomes extraits de la base de Z/eves. Ainsi, le concepteur n'a qu'à copier ces étapes dans l'éditeur de preuve de Z\Eves et par des simples clicks il peut vérifier son schéma. En cas d'erreur, il a des informations sur le concept qui n'adhère pas à une contrainte particulière.

Dans ce qui reste nous expliquerons, en traitant un exemple, la phase de traduction en Z, et la génération automatique des étapes de vérification.

4.1Traduction d'un schéma graphique en langage Z

Dans cette section, nous expliquons la traduction d'un schéma d'une dimension en langage Z au travers de l'exemple de la dimension *Véhicule* contenant la hiérarchie *Clas_Cons* (Figure 4) :

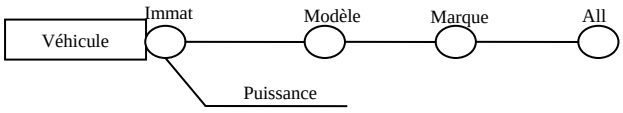


Fig 4 - La hiérarchie *Clas_Cons* de la dimension *Véhicule*

Parmi les attributs de la dimension *Véhicule*, nous avons *Immat, Modèle, Marque, Puisance* et l'attribut *All*. L'attribut *Immat* est l'identifiant. Les attributs *Modèle* et *Marque* sont classés comme des paramètres. *Puissance* est un attribut faible. Ces attributs sont classés selon la hiérarchie *Clas_Cons* (Fig 4). La dépendance fonctionnelle entre ces différents attributs est exprimée par leur ordre dans la hiérarchie.

La description de ce schéma en Z se fait en deux étapes. La première étape déclare les différents attributs et leurs contraintes. Nous réalisons cette phase à travers un axiome box contenant deux parties :

- la déclaration des constituants de la hiérarchie, et
- les prédicats représentant les contraintes, sur ces éléments, qui seront définies par Z\Eves comme des axiomes.

Immat, Modèle, Marque, all, Puissance: AttDim
clas_Cons: NomH

Poids Immat = ID
Poids Marque = Parametre
Poids Modèle = Parametre
Poids all = All
Poids Puissance = Faible
Determine
= {(Immat, Modèle), (Modèle, Marque), (Immat, Puissance)}
Immat ≠ Modèle
Immat ≠ Marque
Immat ≠ Puissance
Immat ≠ all
Modèle ≠ Marque
Modèle ≠ Puissance
Modèle ≠ all
Marque ≠ Puissance
Marque ≠ all
Puissance ≠ all

Dans la deuxième étape, nous définissons le schéma *InstanceHierarchie* qui jouera le rôle d'une instance Hiérarchie. Par ailleurs, nous devons affecter les différentes valeurs nécessaires aux différents ensembles déjà déclarés au niveau du schéma *Hierarchie*.

InstanceHierarchie

Hierarchie

N = clas_Cons
Att = {Immat, Modèle, Marque, all, Puissance}
ParamH = ⟨Immat, Modèle, Marque, all⟩

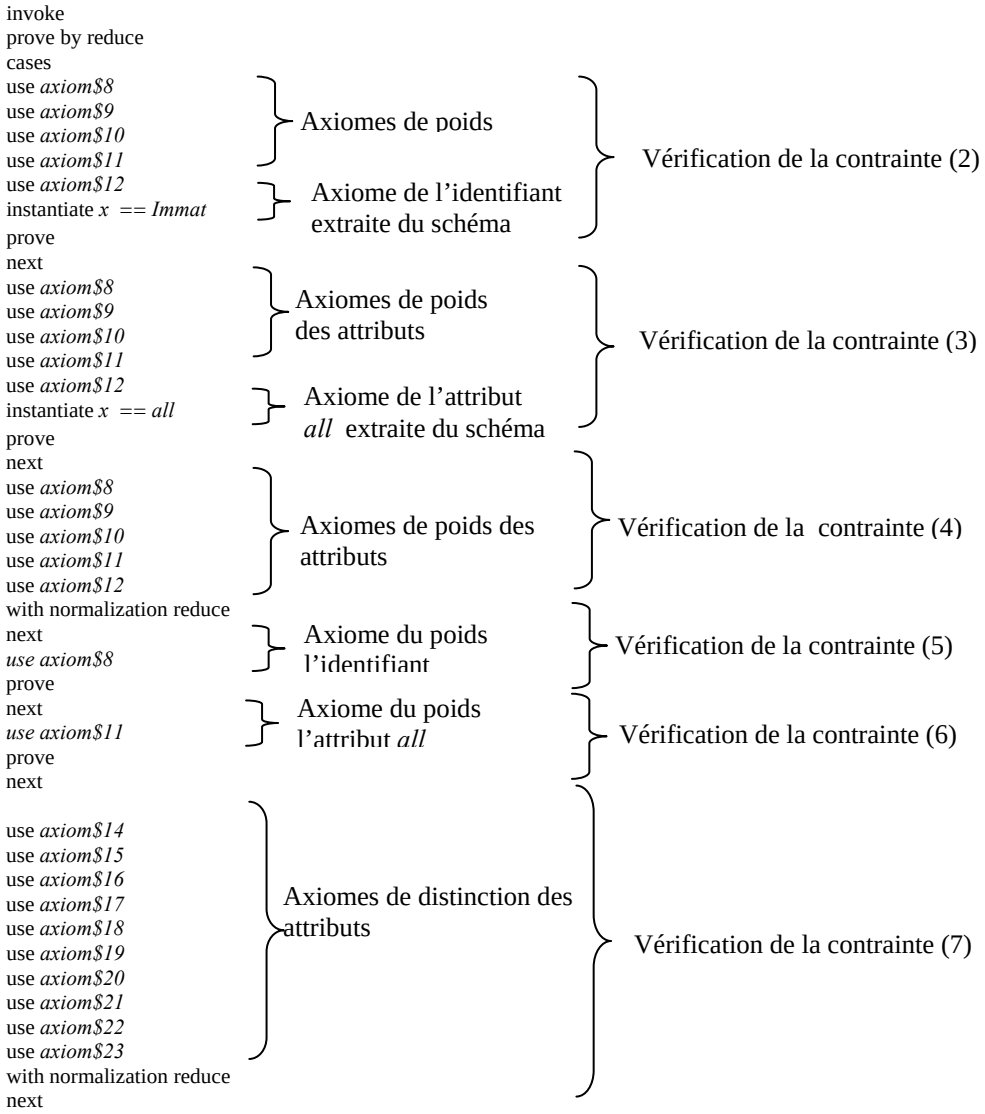
Enfin, nous éditons le théorème d'initialisation. Dans les spécifications formelles en Z la preuve de ce théorème est toujours nécessaire pour la validation d'une spécification. Ce théorème exprime l'exactitude de l'instance. Nous utiliserons alors ce théorème dans le processus de vérification du schéma du concepteur.

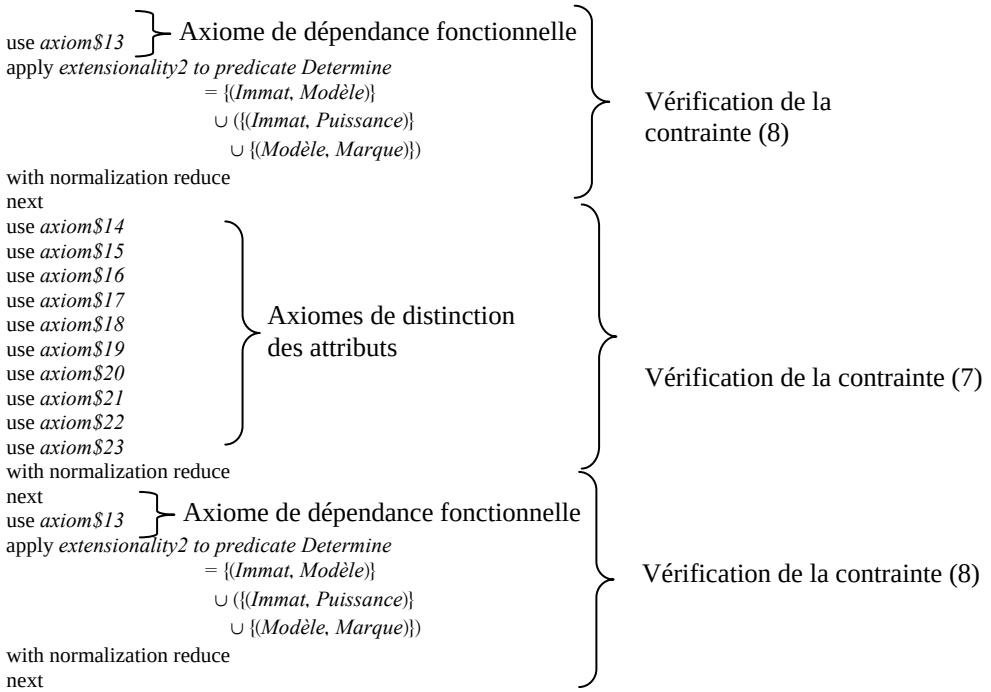
theorem *HierarchieClas_Cons*
∃Hierarchie • InstanceHierarchie

4.2 Etapes de vérification

La vérification consiste en la démonstration du théorème d'initialisation. En traitant plusieurs exemples, nous avons pu dégager une démarche générique. Pour mieux comprendre cette démarche nous proposons la preuve de ce théorème pour l'exemple précédent.

La démonstration commence par la commande *invoke* suivie par *prove by reduce*, qui vérifie automatiquement la contrainte (1) de hiérarchie non vide. Par conséquent, notre démarche commence réellement à partir de la contrainte (2) :


invoke
prove by reduce
cases
use axiom$8
use axiom$9
use axiom$10
use axiom$11
use axiom$12
instantiate x == Immat
prove
next
use axiom$8
use axiom$9
use axiom$10
use axiom$11
use axiom$12
instantiate x == all
prove
next
use axiom$8
use axiom$9
use axiom$10
use axiom$11
use axiom$12
with normalization reduce
next
use axiom$8
prove
next
use axiom$11
prove
next
use axiom$14
use axiom$15
use axiom$16
use axiom$17
use axiom$18
use axiom$19
use axiom$20
use axiom$21
use axiom$22
use axiom$23
with normalization reduce
next
Axiomes de poids
Axiome de l'identifiant extraite du schéma
Vérification de la contrainte (2)
Axiomes de poids des attributs
Axiome de l'attribut all extraite du schéma
Vérification de la contrainte (3)
Axiomes de poids des attributs
Vérification de la contrainte (4)
Axiome du poids l'identifiant
Vérification de la contrainte (5)
Axiome du poids l'attribut all
Vérification de la contrainte (6)
Axiomes de distinction des attributs
Vérification de la contrainte (7)



Nous remarquons dans cette démarche que le démonstrateur Z\Eves vérifie les contraintes (7) et (8) deux fois. Cela s'explique par la dépendance entre ces contraintes et la stratégie de preuve de ce démonstrateur. Notons aussi que si un schéma ne satisfait pas un axiome, le démonstrateur de preuve Z/eves ne produit pas la commande *next* et le concepteur devrait revisiter son schéma au niveau du concept pertinent à l'axiome.

D'autre part, il est clair d'après l'exemple que la preuve suit les déclarations des contraintes dans le schéma *Hierarchie.* Cependant, pour un non expert en Z, connaître la commande à utiliser et retrouver les axiomes à appeler à chaque étape de la preuve pourrait être des tâches pénibles. En outre, à travers le traitement de plusieurs exemples, nous avons remarqué que les étapes de la preuve illustrée ci-dessus sont assez standard. Ce qui nous a motivé de proposer un générateur de preuve générique et automatique.

4.3 Génération automatique de preuve

Notre objectif est de créer un générateur des étapes de preuve, capable de chercher les axiomes nécessaires à partir du schéma traduit en Z, rédiger des axiomes supplémentaires dérivés à partir du schéma graphique, et introduire ces axiomes en fonction de l'avancement de la preuve dans Z\eves.

4.3.1 Recherche des axiomes à partir du schéma traduit en Z

Nous reprenons l'exemple précédent. La formalisation en Z de cet exemple sera stockée dans un fichier d'extension *.zev*. Ce type de fichier est structuré sous forme de balise,

contenant entre autre l'axiome box comprenant les différentes déclarations. La figure 5 présente cet axiome box dans son format lisible.

```
<axiomatic-box location="gui::61"><decl-part/>Immat, Modèle, Marque, all, Puissance:
AttDim
clas_Cons: NomH
T102, T103, T104, Fiesta, Clio, Golf, Ford, Renault, Volswagen, vall: Dom<ax-part/> Immat .
poids = ID
Marque . poids = Parametre
Modèle . poids = Parametre
all . poids = All
Puissance . poids = Faible
Determine  = {(Immat, Modèle), (Immat, Puissance), (Modèle, Marque)}
Immat &neq; Modèle
Immat &neq; Marque
Immat &neq; all
Immat &neq; Puissance
Modèle &neq; Marque
Modèle &neq; Puissance
Modèle &neq; all
Marque &neq; Puissance
Marque &neq; all
Puissance &neq; all
</axiomatic-box></textItem>
```

Fig. 5 - Format de l'axiome box de la hiérarchie de *Véhicule*

Nous remarquons depuis la figure 5 que nous pouvons localiser un axiome box à partir des deux balises de début « `<axiomatic-box` » et de fin « `</axiomatic-box>` ». La balise de séparation entre la partie déclarative et la partie prédicat d'un axiome box est « `<ax-part/>` ». De plus, dans cette partie, nous retrouvons le nom de la hiérarchie à vérifier. Ainsi, pour chaque hiérarchie, nous devons collecter les contraintes déclarées dans sa partie prédicat.

4.3.2. Edition des étapes de preuve

Une fois que nous avons collecté les axiomes à partir du schéma traduit en Z, nous pouvons entamer l'édition automatique des étapes de la preuve. Quant aux axiomes extraits du schéma graphique, ils seront définis au fur et à mesure de la phase d'édition des étapes de preuve.

Les étapes de vérification générées se composent de deux catégories de zones. La première catégorie contient les zones qui varient selon le schéma du concepteur : ce sont les axiomes retrouvés à partir de la base ou générés à partir de la hiérarchie dessinée. La deuxième catégorie reste toujours fixe, elle est indépendante de tous schémas : ces zones contiennent les commandes de l'outil du démonstrateur Z\Eves.

Soit une hiérarchie H composée de de n attributs :
L'algorithme de génération des étapes, appliqué sur H, produit les lignes de commande suivantes :

invoke	use *axiome de poids* a_n
prove by reduce	prove
cases	next
use *axiom$n° d'axiome de poids* a_1	use *axiome de distinction 1*
use *axiom$n° d'axiome de poids* a_2	use *axiome de distinction 2*
…	…
use *axiom$n° d'axiome de poids* a_n	use *axiome de distinction* $\sum_{1}^{n-1} i$
instantiate $x == a_1$	with normalization reduce
prove	next
next	use *axiom$n° d'axiome de dépendance fonctionnelle*
use *axiom$n° d'axiome de poids* a_1	apply *extensionality2 to predicate axiome de dependence fonctionnelle*
use *axiom$n° d'axiome de poids* a_2	with normalization reduce
…	next
use *axiom$n° d'axiome de poids* a_n	use *axiom$n° d'axiome de distinction 1*
instantiate $x == a_n$	use *axiom$n° d'axiome de distinction 2*
prove	…
next	use *axiom$n° d'axiome de distinction* $\sum_{1}^{n-1} i$
use *axiom$n° d'axiome de poids* a_1	with normalization reduce
use *axiom$n° d'axiome de poids* a_2	next
…	use *axiom$n° d'axiome de dépendance fonctionnelle*
use *axiom$n° d'axiome de poids* a_n	apply *extensionality2 to predicate axiome de dependence fonctionnelle*
with normalization reduce	with normalization reduce
next	next
use *axiome de poids* a_1	
prove	
next	

Nous avons implanté cet algorithme et nous l'avons testé sur plusieurs exemples. La Figure 6 illustre le résultat de cet algorithme pour l'exemple du schéma de la dimension *Véhicule*.

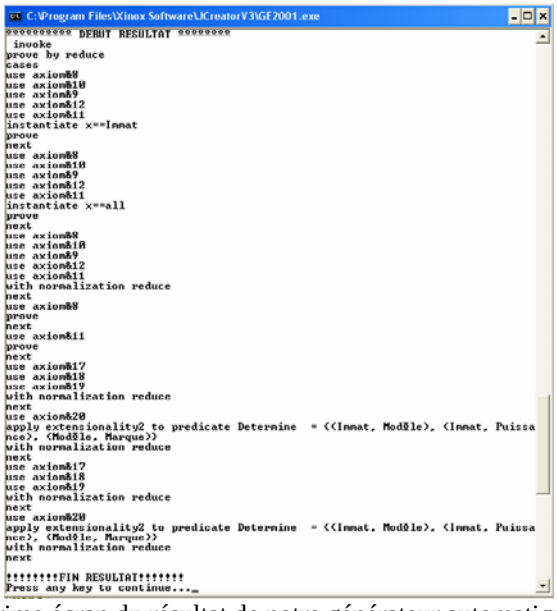


Fig. 6 - Imprime écran du résultat de notre générateur automatique de preuve

5 Conclusion

Dans ce papier, nous avons proposé une approche générique et automatique de vérification des hiérarchies des modèles multidimensionnels. Notre approche est basée sur une démarche formelle de vérification de schéma de dimension. Le langage formel utilisé dans notre démarche est le langage de spécification formelle Z et la vérification est assistée par l'outil Z\Eves. Nous sommes en court d'étendre notre approche pour traiter les contraintes relatives aux modèles multidimensionnels en constellation.

Référence

Abelló A., Samos J., Saltor F. (2006), "YAM2: a multidimensional conceptual model extending UML". Information Systems. Vol 31, p 541-567.

Carpani F., Ruggia R., (2001) "An Integrity Constraints Language for a Conceptual Multidimensional Data Model". Dans 13th International Conference on Software Engineering & Knowledge Engineering (SEKE'01), Argentina,.

Franconi E. and Kamble A., (2004) "The GMD Data Model and Algebra for Multidimensional Information" Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, Proceedings.

Ghozzi F., Ravat F., Teste O., Zurfluh G., (2003) "Modèle Dimensionnel à Contraintes". Dans Revue des Sciences et Technologies de l'Information, Série RIA- ECA, Hermes – Lavoisier, Vol. 17, N. 1-2-3, p.43-56.

Hurtado C.A., Mendelzon A.O., (2002) "OLAP Dimension Constraints". Dans 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02), Madison, USA, p. 169-179.

Kimball R., Ross M., (2002), "The Data Warehouse Toolkit", Wiley, New York,2nd edition.

Lechtenbörger J., G. Vossen, (2003),"Multidimensional normal forms for data warehouse design". Dans Revue Information Systems, Vol. 28, N. 5, p. 415-434.

Lujàn S, Trujillo J., Song, (2002) "Extending the UML for Multidimensional Modeling" The Unified Modeling Language : 5th International Conference, Dresden, Germany, September 30 - October 4, Proceedings, p 290- 304.

Saaltink M. (1999). The Z/EVES 2.0 User's Guide. ORA Canada, OneNicholas Street, Suite 1208, Ottawa (Ontario), K1N 7B7.

Salem A., Ghozzi F., Ben-Abdallah H. (2008), " MULTI-DIMENSIONAL MODELING - Formal Specification and Verification of the Hierarchy Concept. Databases and Information Systems Integration, ICEIS, p317- 322.

Spivey J.M. (1992)," The Z Notation: a Reference Manual". Prentice-Hall.

Les systèmes d'aide à la décision basés sur les entrepôts de données physiques/logiques

Madiha Bouainah *
Ali Mellit*
*Laboratoire LAMEL, Université de Jijel,
BP 98 Ouled Aissa, Jijel, 18000, Algérie
{mbouainah, ali_melit}@yahoo.fr

Résumé. Beaucoup de situations nécessitant une prise de décision rapide. Par contre, toute décision nécessite au moins la recherche et l'accès à des informations relatives au problème à résoudre et leur traitement dans certains cas. Pour atteindre cet objectif, des systèmes d'aide à la décision basés sur l'approche entrepôt de données ont vue le jour. Le but de tels systèmes est l'exploitation d'une masse de données importante résultant des systèmes d'informations des entreprises afin de comprendre le sens et de déceler les relations entre données pour offrir une aide à la décision. Dans cet article, nous décrivons les différentes architectures pour la création d'un système d'aide à la décision. Ensuite, nous présentons une approche hybride du modèle de conception d'un système décisionnel basé sur l'approche Data Warehouse.

1 Introduction

C'est dans les années 1990 que les entreprises comprennent que les données sont non seulement utiles dans le cadre d'une utilisation opérationnelle, mais qu'elles peuvent leur trouver une utilisation stratégique. L'exploitation directe des données des bases de production s'avère souvent inadaptée à leurs besoins décisionnels en raison du temps d'accès important. Face à ce problème ; les industriels ont progressivement mis en place des entrepôts de données, véritables interfaces entre les bases de données et les décideurs.

2 Data Warehouse

Ce n'est pas une usine à produire l'information ; mais plutôt un moyen de la mettre à disposition des utilisateurs de manière efficace et organisée.

3 Les composants de base d'un système décisionnel basé sur l'approche entrepôt de données

L'architecture des systèmes décisionnels met en jeu quatre éléments essentiels :

3.1 Les sources de données

Elles sont nombreuses, variées, distribuées et autonomes. Elles peuvent être internes (bases de production) ou externes (Internet) à l'entreprise.

3.2 L'entrepôt de données

C'est le lieu de stockage centralisé des informations. Il est utile pour les décideurs et met en commun les données provenant des différentes sources et conserve leurs évolutions.

3.3 Les magasins de données

Ils sont des extraits de l'entrepôt et sont orientés sujet. Les données sont organisées de manière adéquate pour permettre des analyses. Ils sont rapides à des fins de prise de décision.

3.4 Les outils d'analyse

Ils permettent de manipuler les données suivant des axes d'analyses. L'information est visualisée à travers d'interfaces interactives et fonctionnelles dédiées à des décideurs souvent non informaticiens (directeurs, chefs de services,).

4 La problématique de développement des systèmes d'aide à la décision

Les systèmes décisionnels comportent deux types d'espaces de stockages que sont les entrepôts de données et les magasins de données. Les travaux relatifs aux systèmes décisionnels peuvent être classés selon deux approches différentes.

4.1 L'approche 1 : « un SD avec un DW logique »

Des travaux abordent les systèmes décisionnels sans distinguer l'espace de stockage comme un entrepôt de données et des magasins de données. Ils représentent le système décisionnel comportant un unique espace de stockage appelé entrepôt de données logique.

Définition : Un entrepôt de données logique est une collection de données intégrées, orientées sujet, non volatiles et historisées pour la prise de décision (Inmon, 1992). L'entrepôt de données logique c'est tout simplement l'union des magasins de données qui le composent (Bolognini, 2002).

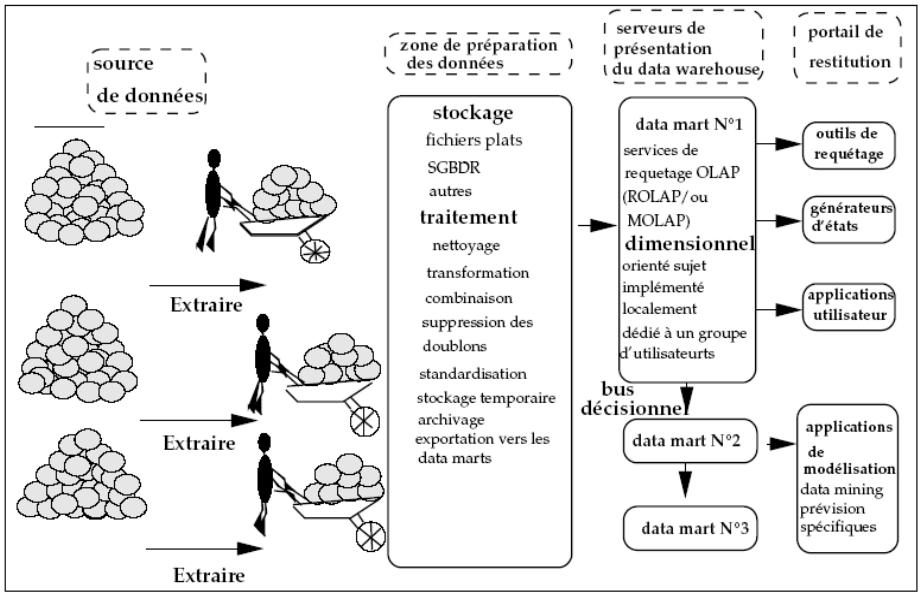


FIG. 1 – *Un système décisionnel avec un DW logique.*

4.2 L'approche 2 : « Un SD avec un DW physique »

D'autres travaux se basent sur la séparation de l'entrepôt de données physique et des magasins de données. L'entrepôt proprement dit est le lieu centralisé de toute l'information pertinente pour les utilisateurs tandis que le magasin de données est un extrait d'entrepôt dédié à un type d'utilisateurs et répondant à un besoin spécifique.

Définition : un entrepôt de données physique est le lieu de stockage centralisé d'un extrait des bases de production. Cet extrait concerne les données pertinentes pour l'aide à la décision. Elles sont intégrées et historisées. L'organisation des données est faite selon un modèle qui facilite la gestion efficace des données et leur historisation. (Teste, 2000).

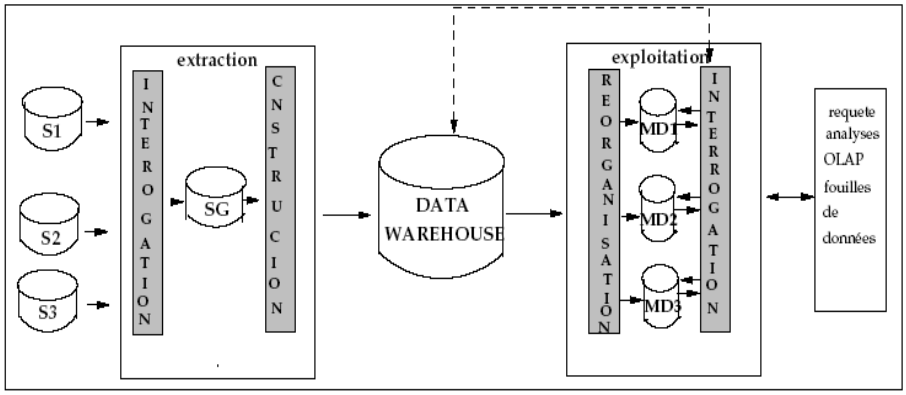


FIG. 2 – *Un système décisionnel avec un DW physique.*

5 Comparaison entre les deux approches

D'après notre étude sur les systèmes décisionnels, le tableau suivant résume la différence entre les deux approches.

L'approche N°1 : un SAD avec un DW logique	**L'approche N°2 : un SAD avec un DW physique**
l'entrepôt de données est une collections de données orientées sujets, intégrées, non volatiles et historisées, organisées pour le processus de décision	l'entrepôt de données est le lieu de stockage d'un extrait des bases de production. Cet extrait conserve les données pertinentes pour le support de décision
Le magasin de données est un sous ensemble logique de l'entrepôt de données	le magasin de données est un extrait de l'entrepôt de données. Les données extraites sont adaptées à une classe de décideurs ou à un usage particulier
les données de l'entrepôt sont modélisées en utilisant le modèle dimensionnel	la modélisation multidimensionnelle est utilisée seulement avec les magasins de données

TAB. 1 – *Comparaison entre un SAD avec un DW logique et un DW avec un DW physique.*

5.1 Comparaison du temps de recherche d'information

L'interrogation des données englobe toutes les activités qui consistent à demander une information à un magasin de données, la génération d'états, les applications complexes d'aide à la décision, les requêtes des applications de modélisation et le data mining.

5.1.1 Le temps d'interrogation avec l'approche logique

Soit

TR_i : le temps de recherche d'information dans le DM i
TP : le temps de passage entre magasins de données
P_i : la probabilité que l'information existe au niveau du DM i

Le temps d'interrogation = $TR_1+TP+ (1-P_1)TR_2+TP+(1-P_2)TR_3+\ldots\ldots+ +(1-P_{n-1})TR_n$

5.1.2 Le temps d'interrogation avec l'approche physique

Soit

TRW= le temps de recherche d'information dans un entrepôt de données.

$$TRW=\sum_{i=1..N} TRi$$

Le temps d'interrogation = TR1+TP+TRW

Résultat : Le temps moyenne de recherche d'information avec l'approche logique $\geq$ le temps moyenne de recherche d'information avec l'approche physique.

5.2 Comparaison de l'espace de stockage entre les deux approches :

Soit

SM : l'espace de stockage moyenne d'un data Mart
SW: l'espace de stockage d'un data warehouse = $\sum$ SM (i)
avec i=1...n

L'espace de stockage utilise avec l'approche N°1= SM (1) +…+SM (n)
L'espace de stockage utilisé avec l'approche N°2= SM (1) +…+SM (n)+SW

Résultat : l'espace de stockage utilisé avec l'approche physique=2*l'espace de stockage utilisé avec l'approche logique.

6 Proposition d'une solution hybride entre l'approche N°1 et l'approche N°2

Dans le but d'optimiser le temps de recherche d'information à partir des magasins de données et de minimiser l'espace de stockage utilisé pour la construction d'un système décisionnel, nous avons proposé une solution hybride basée sur la construction des magasins de données et d'entrepôts de données en même temps.

Avec cette architecture, les données sont stockées dans les magasins de données sous format détaillée ; par contre les données stockées dans l'entrepôt de données correspondant à une agrégation de certaines évolutions détaillées.

6.1 L'objet magasin

Un objet magasin modélise une entité extraite de la source globale. Il peut être constitué à partir d'un ou plusieurs objets source ou d'une partie d'objets sources.

Définition : Un objet magasin O est défini par un n-uplet (oid, S0, Histoire, Origine) où

1- oid est l'identifiant interne,

2- S0 est l'état courant correspondant à la dernière valeur extraite,

3- Histoire= (Sp1, Sp2,.., Spp) est un ensemble fini d'états passés correspondant au détail Des évolutions de valeur de l'objet magasin,

4- Origine=(s-oid1, s-oid2,.., s-oids) est un ensemble fini des identifiants des objets Source à partir desquels la valeur de l'objet magasin est obtenue.

6.2 L'objet entrepôt

Un objet entrepôt correspondant à une agrégation de certaines évolutions des objets magasins.

Définition : Un objet entrepôt O est défini par un n-uplet (oid, Archive, Origine) où

1- Oid est l'identifiant interne,

2- Archive= (Sa1, Sa2,.., Saa) est un ensemble fini d'états archivés correspondant à une Agrégation de certaines évolutions détaillées (états passés),

3-origine = {MDi}.

6.3 Comparaison entre l'architecture hybride et les deux autres solutions

6.3.1 Le temps d'interrogation avec l'approche hybride

TRWh= le temps de recherche d'information dans un entrepôt de données.

Remarque : TRWh<TRW

Le temps d'interrogation = TR1+TP+TRWh+TRi

Résultat : Le temps de recherche d'information de l'architecture physique ≤ Le temps de recherche d'information de l'architecture hybride≤ le temps de recherche d'information de l'architecture logique.

6.3.2 Calcul de l'espace de stockage avec la solution hybride

L'espace de stockage utilisé avec l'approche N°1= ∑ taille des magasins de données. L'espace de stockage utilisé avec l'approche N°2= 2* taille d'un entrepôt de données logique.

La taille d'un entrepôt de données hybride = **0.X** (taille d'un entrepôt de données logique) Avec X € {1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9}.

L'espace de stockage utilisé avec l'architecture hybride = taille d'un Entrepôt de données logique +0.X (taille d'un entrepôt de données Logique).

Résultat : L'espace de stockage utilisé avec l'architecture logique≤L'espace de stockage utilisé avec l'architecture hybride≤L'espace de stockage utilisé avec l'architecture physique.

7 Conclusion

Cet article décrit les différentes architectures pour la création d'un système d'aide à la décision. En plus nous avons exposé les solutions existantes selon le concept d'entrepôts de données (physique et logique), par la suite nous avons présenté deux architectures pour la création d'un système décisionnel, la première est basée sur un entrepôt de données logique, la deuxième utilise un entrepôt de données physique.

L'étude comparative des deux approches nous a permis de proposer une solution Hybride entre les deux approches.

Références

Bellatreche, L. (2000). *Techniques d'optimisation des requêtes dans les data warehouse* 16 : 86960 futuroscope-France.

Bessam, A. (2002). *Un système intelligent pour la planification de la production de l'énergie électrique basé sur le couplage des méthodes commonKads/SADT*. Mémoire de Magister, Université de Jijel.

Bolognini, N. R. (2002). *Étude pour la création d'un entrepôt de données dans le cadre de l'assurance vie et transformation des données en informations utiles en vue d'une prise de décision. 74 Pages*. École des hautes études commerciales -université de Lausanne.

Bret, F. et O. Teste (2001). *Construction graphique d'entrepôts et de magasins de données*. Vol. 18, IRIT équipe SIG, Université Paul Sabatier, Toulouse III, France.

Brobst, S. et J. Rarey (2001). *Les cinq étapes de l'évolution d'un entrepot de données actif*. Vol. 4. Accessible au site Web : www.taradata.com.

El Helou, G. et C. Abou Khalil (2004). *Data mining (techniques d'extraction des connaissances*. Université panthéon-assas, Paris II.

Garlatti, S. (1998). *Multimédia et système d'aide à la décision en situation complexe*. Ecole nationale supérieure des communications de Bretagne.

Garlatti, S. (2001). *Les systèmes interactifs d'aide à la décision en situations complexes*. Ecole nationale supérieure des communications de Bretagne.

Herznandez, H et P. Canarelli (2006). *Systèmes d'aide à la décision: les limites de l'information et le role des outils basés sur la connaissance*. Vol. 6 pages.

Inmon, B. (1992). *Building the data warehouse.* John Wiley.

Kimball R. (2005). *Le data warehouse guide de conduite de projet, volume 575 pages. 61,* Paris: Eyrolles.

Marhoumi, F. (2006). *Entrepôts de données XML: développement d'un outil extraction transformation load (ETL)*. Mémoire d'ingénieur civil informaticien, Université Libre de Bruxelles, 2006.

Morge, M. et P. Beaune (2000). *Conception Multi-agent d'un système d'aide à la décision collective justification automatique pour la confrontation des opinions.*

Mousseau, V. (2003). *Élicitation des préférences pour l'aide multicritère à la décision*. Université Paris Dauphine-U.F.R sciences des organisations.

Sdc (2005). *Support de cours. Informaticien de gestion /HES option d'école/base de données/informatique décisionnelle*. Vol. 49. Haute école économie.

Sdt (2000) *Système d'information décisionnel et data warehouse* 20, accessible au site Web : www. décisionnel.net.

Serres, E. (2004). *Le système d'information décisionnel ou comment piloter l'entreprise grâce au data warehouse*. École des mines de Paris.

Teste, O. (2000). *Modélisation et manipulation d'entrepots de données complexes et historisées*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France.

Teste, O. (2001) olivier teste. *Élaboration d'entrepôts de données complexes*. Université Paul Sabatier, Toulouse.

Toussaint, H. (2000). *Systèmes informatiques d'aide à la décision distribués (le modèle fédéraliste)*.

Volle, M. (2001). La place de l'aide à la décision dans le système d'information. *Club des maîtres d'ouvrages des systèmes d'information* 69 : 57-67.

Summary

Many situations require a fast decision making. However, any decision requires at least searching and accessing information related to the problem to be solved, and to process them in some cases. In order to achieve this goal, decision support systems that are based on the approach of data warehousing have emerged. Such systems aim at exploiting a great volume of data resulting from the enterprise information system, in order to understand their meaning and to detect the relationships between them and thus, help the managers in decision making. In this paper, we describe different architectures for the creation of a decision support system. Then we present a hybrid approach of a design model of a decision system based on the data warehouse approach.

Évaluation de la répartition d'un entrepôt de données

Tekaya Karima*, Abedellaziz Abdelatif**

**Faculté des sciences de Tunis, Département informatique,*
Campus universitaire, - 1060 Tunis
Karima.Tekaya@.isi.rnu.tn

***Faculté des sciences de Tunis, Département informatique,*
Campus universitaire, - 1060 Tunis
sql.expert@planet.tn

Résumé. A l'image des compagnies, la décentralisation du système d'information décisionnel est devenue de plus en plus accrue. De ce fait, une organisation centralisée de celui-ci risque de devenir inefficace. L'accès aux données à distance est moins souple et plus laborieux. Dans ce contexte, plusieurs techniques de répartition ont été proposées dans l'état de l'art, celles ci sont généralement inspirées des techniques développées dans les bases de données relationnelles. Or, de telles utilisations doivent être réévaluées avec les spécificités des entrepôts de données, tel que, étudier l'impact de la répartition sur le processus de chargement et de rafraîchissement des données et l'évaluation des accès inter site. Dans cet article, nous proposons un formalisme de modélisation et de suivi de la répartition géographique d'un entrepôt de données à base de matrices multidimensionnelles. Nous développons par la suite, des modèles de coûts pour l'évaluation du processus de chargement des données et des coûts des accès inter sites. Enfin, nous concrétisons notre travail par des expérimentations.

1 Introduction

La répartition d'un Entrepôt de Données (ED) a été proposée pour la première fois par Noaman, A.Y., Barker, K. (1997), ces auteurs ont mit en évidence l'importance de réorganiser les données lorsque les décideurs sont répartis géographiquement et que les sources sont elles-mêmes réparties sur plusieurs sites. La répartition d'un ED centralisé va se faire en fonction des besoins des utilisateurs. Les besoins des décideurs sont exprimés à travers les requêtes OLAP que nous allons utiliser comme critère de répartition. Les requêtes OLAP peuvent être classifiées selon deux catégories : des requêtes OLAP globales aux différents sites (telles que celles effectuées au niveau du siège) et des requêtes locales pour les besoins locaux de chaque site de la compagnie (par exemple un point de vente). Une gestion centralisée de l'ED pourra rendre plus souple l'exploitation des données agrégées. Cette solution a été jugée dans l'état de l'art difficile à mettre en place puisqu'elle nécessite une méta base assez compliquée et la gestion de celle-ci pourra devenir une tâche fastidieuse pour l'administrateur de l'entrepôt. Dans cet article, nous allons présenter dans la section 2, les travaux reliés à la répartition des ED. Dans la section 3, nous allons évoquer la probléma-

tique traitée. Dans la section 4, nous allons proposer une méthode d'évaluation de la répartition d'un ED par la proposition d'un ensemble de modèles de coûts. Dans la section 5, nous allons présenter un exemple d'application des concepts proposés. Dans la section 6, nous allons présenter des résultats expérimentaux pour la concrétisation de la démarche proposée et la validation des modèles de coût développés. Nous finissons et quelques perspectives.

2 Etat de l'art

Plusieurs travaux ont considéré la répartition comme l'une des techniques d'optimisation dans les ED. Bellatreche, L., Kamalakar K**.,** Mukesh M.(2002) ont traité le problème de la fragmentation horizontale primaire et dérivée dans un environnement d'ED centralisé. Puis, dans le même contexte, ils ont développé une technique de sélection d'un schéma de fragmentation en utilisant les algorithmes génétiques dans Bellatreche L., Boukhalfa, K. (2005). Le même travail a été développé avec Boukhalfa K., Bellatreche, L. (2006) en combinant des algorithmes génétiques et le recuit simulé pour la conception physique des entrepôts de données. Récemment, dans Boukhalfa, K., Bellatreche, L., Richard, P. (2008), Les auteurs ont proposé l'implémentation de la fragmentation horizontale primaire et dérivée sous ORACLE 10G.
Les travaux étudiés se sont orientés vers la fragmentation dans un contexte centralisé pour l'optimisation du temps d'exécution des requêtes OLAP. Par contre, peu de travaux ont montré l'intérêt de la fragmentation pour la répartition d'un ED dans un contexte géographiquement distribué. Quelques travaux ont proposé une architecture répartie pour un ED. Dans Noaman, A.Y., Barker, K. (1997) une architecture répartie d'un ED a été proposée. Ce travail se base essentiellement sur l'architecture ANSI/SPARC qui est composée de trois niveaux : le niveau interne, le niveau conceptuel et le niveau externe. Dans Noaman, A.Y., Barker, K. (1999) les auteurs ont proposé une architecture répartie d'un ED basée sur l'approche Top-Down. Ils ont par la suite, développé dans Noaman, A.Y., Barker, K. (1999)-2, un algorithme de fragmentation de la table des faits. Mais, le travail proposé n'a été ni implémenté, ni des expérimentations ont été présentées. Les mêmes auteurs ont développé dans Noaman, A.Y., Barker, K. (2000) une architecture hiérarchique d'un ED réparti.
Nous avons proposé dans Tekaya, K, Abdellatif, A. (2004) une démarche de modélisation de la répartition des données d'un ED et dans Tekaya, K., Abdellatif, A., (2005). Nous avons appliqué la même démarche sur un exemple concret. Furtado, P. (2006) a montré à travers des expérimentations que la répartition géographique d'un ED aura comme conséquence l'optimisation de l'organisation physique des données.

La répartition des entrepôts de données n'est plus en elle-même une problématique récente, puisque plusieurs chercheurs se sont investi au développement de technique de fragmentation permettant d'optimiser le temps de réponse des requêtes OLAP tel que Boukhalfa, K., Bellatreche, L., Richard, P. (2008), ou bien ont développé des techniques d'allocation assez avancées pour l'optimisation du stockage physique des données tel que Furtado, P. (2006).

De ce fait, pour valider les différents travaux proposés une méthode d'évaluation de la répartition est devenue de plus en plus nécessaire. Dans ce contexte, Chakravarthy, et al. (1992). Ont proposé une fonction objectif pour l'évaluation de la fragmentation verticale d'une base de données relationnelles. Ezeife, C.I., Zheng, J.. (1999) ont exploité la même fonction objectif et l'ont adapté pour l'évaluation de la fragmentation horizontale d'une base

de données orientée objet. Ezeife, C.I., Zheng, J. (2001) ont prouvé encore une fois l'efficacité de la fonction objectif et l'ont exploité pour le contrôle dynamique de la performance d'une base de données orientée objet. Ezeife, C.I., Pinakpani, D. (2003) ont développé le travail précèdent et ont développé une technique de fragmentation incrémentale basée sur une évaluation continue en utilisant la même fonction objectif. On a constaté donc, que cette fonction ne cesse de montrer son efficacité pour l'évaluation d'une certaine répartition.

3 Problématique

Dans un contexte réparti si les besoins diffèrent selon les sites, on procède à la construction de magasins de données indépendants. Chaque magasin de données pourra utiliser son propre formalisme, outils, et environnement de travail. Ce qui semble être une tâche plus simple et moins longue. D'autres part, les requêtes OLAP peuvent être classifiées selon deux catégories : des requêtes globales à l'entrepôt et des requêtes locales aux magasins de données. L'exécution de requêtes globales sur des magasins de données indépendants est une tâche très laborieuse qui nécessite une bonne connaissance de tous les magasins de données. De ce fait, un formalisme uniforme avec l'utilisation d'un SGBD réparti pourra rendre la tâche de l'administrateur plus facile et renforcer ainsi, l'exploitation de l'ED. La gestion de la méta base est centralisée : Pour alléger cette tâche, un bon formalisme de présentation de la globalité des métas donnés est nécessaire et peut venir en aide à l'administrateur pour une gestion plus souple des différentes tâches. Un ED peut être considéré à première vu comme une base de données relationnelle vu la structure relationnelle des tables et le respect de l'intégrité référentielle. Les travaux existants ont expérimenté plusieurs techniques de fragmentations et d'allocation et ont montré de fabuleux résultats surtout dans un contexte centralisé. Mais, peu de travaux ont traité la répartition d'un ED dans un contexte réparti. D'autres part, le point qui diffère un ED d'une base de données relationnelle c'est le processus de chargement des données. Celui-ci, ne peut en aucun cas être négligé tout au long du processus de répartition. Il sera intéressant d'étudier l'impact de la répartition d'un ED sur le coût de chargement ou de rafraîchissement des données à partir des sources vers les sites cibles. Il Faut aussi contrôler, suite à une certaine répartition, le coût des accès à distance et étudier les besoins de réplication. De ce fait, une méthode d'évaluation est nécessaire pour estimer si une certaine répartition est efficace.

4 Modélisation et suivi d'un ED réparti

Nous proposons un nouveau formalisme de modélisation basée sur un ensemble de matrices multidimensionnelles.

4.1 Concepts de base

Comme concepts de base, nous proposons des matrices multidimensionnelles pour modéliser les données correspondantes aux sources à l'ED, les critères de fragmentation, ainsi que les fragments engendrés et les sites où seront alloués ces fragments:

Matrice d'intégration des données sources. Englobe pour chaque attribut des tables dimensions les sources, les transformations subites et pour chaque mesure de la table des faits les fonctions d'agrégat utilisées ainsi que les attributs sources utilisés
Matrice de fragmentation horizontale primaire et dérivée. En utilisant la technique de fragmentation primaire et dérivée développée dans Noaman, A.Y., Barker, K. (1999)-2, cette matrice englobe pour chaque table dimension les fragments engendrés, ainsi que les prédicats de sélection utilisés. Pour la table des faits, on indique les fragments horizontaux dérivés, les tables dimensions concernées, ainsi que les conditions de semi jointure utilisées.
Matrice d'allocation. Englobe les sites dans lequel sera alloué chaque fragment engendré.
Matrice métabase. Cette matrice vient en déduction des matrices précédentes, elle englobe pour chaque fragment les sources utilisées, les transformations utilisées, ainsi que le site dans lequel il sera alloué.
Entrepôt de données réparti. Nous désignons par ce concept, un entrepôt réparti sur plusieurs sites. Pour le gérer on procède à l'utilisation d'un ETL distant dont la gestion est centralisée mais, le chargement et le rafraîchissement sont effectués à distance aux sites correspondants.
Magasin de Données (MD). Un ED réparti est un ensemble de plusieurs sous ED ayant des besoins spécifiques dans chaque site. On considère MD un sous ED répondant à des besoins bien spécifiques des décideurs d'un particulier.

4.2 Démarche de modélisation et suivi de la répartition d'un ED

Cette section présente les étapes à suivre pour la modélisation des différentes matrices proposées : (Figure 1)

- Etape 1 : Création de la matrice d'intégration des données sources ;
- Etape 2 : Création de la matrice de fragmentation horizontale primaire et dérivée ;
- Etape 3 : Création de la matrice d'allocation ;
- Etape 4 : Création de la matrice méta base.

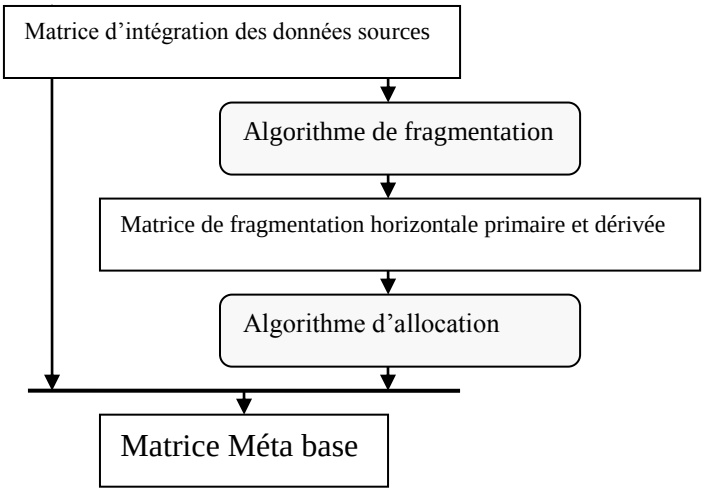


FIG. 1 – *Etape de modélisation*

Dans ce qui suit, nous allons détailler les différents concepts et leurs formalismes de présentation.

4.3 Formalisme

4.3.1 Matrice d'intégration des données sources (Tableau 1)

Pour modéliser le processus d'extraction, transformation et chargement des données sources, nous proposons comme formalisme la Matrice d'Intégration des Données Sources (*MIDS)* (Tableau1), pour chaque attribut A_{jL} ((1<=j<=k), (1<=L<=m) ; avec k le nombre de table dimension et m le nombre d'attribut par table dimension du modèle en étoile. On indique la source, c'est-à-dire, à partir de quelle table source (TSr (1<=r<=s ; s le nombre de tables sources utilisées pour alimenter l'ED et les différentes transformations subites. La Table des Fait (TF) est constituée par un ensemble de clés étrangères faisant référence aux clés primaires des tables dimension et d'un ensemble de mesure (M_1,…,M_p,…M_q) avec (1<=p<=q ; q le nombre de mesure de la table des faits). Une transformation peut être élémentaires ou composite :

1. Une transformation élémentaire est obtenue en calculant la fonction $f_{te(}A_{rz)}$, elle donne comme résultat un attribut a_{jL} qui sera intégré dans l'ED.
2. Une transformation peut être composites (TC), c'est-à-dire calculée en fonction de plusieurs attributs sources. Le résultat sera obtenu en appliquant la fonction $f_{tC(}A_{r1},...,A_{rx)}$.

Tables	Attributs	Sources de données																TC
		TS_1					TS_r					TS_s						
		A_{11}		A_{1z}		A_{1x}	A_{r1}		A_{rz}		A_{rx}	A_{s1}		A_{sz}		A_{sx}		
TD_1	a_{11}	$f_{te(}A_{11)}$																
	…																	
	a_{1L}	X				X	X					X						$f_{tC()}$
	…			$f_{te(}A_{1z)}$														
	A_{1m}																	
TD_j	a_{j1}					$f_{te(}A_{1x)}$												
	…																	
	$a_{j\,L}$								$f_{te(}A_{rz)}$									
	…																	
	a_{jm}																$f_{te(}A_{sx)}$	
TD_k	a_{k1}	X		X			X		X			X		X				$f_{tC()}$
	…										$f_{te(}A_{rx)}$							
	a_{k1}																	
	…			X		X			X		X							
	a_{km}																	
TF	m_1			X		X			X		X			x		X		$f_{tC()}$
	….																	
	m_p																	
	….																	
	m_q																	

TAB. 1– *Matrice Intégration des données sources*

Une transformation peut faire l'objet de l'application d'une certaine formule mathématique ou bien l'application d'une ou de plusieurs fonction d'agrégats :

1. Une formule mathématique répond généralement à un besoin en calcul spécifique à l'activité. Par exemple, le changement de la monnaie utilisée de l'Euro vers le Dinars Tunisien, ceci nécessite la multiplication du montant par le taux de change.
2. Une fonction d'agrégat consiste en l'application d'un opérateur algébrique tel que SUM(), AVG(), MAX(), etc .

Un attribut A_{rz}((1<=r<=s, 1<=z<=x ; avec x le nombre d'attribut pour chaque TS) contribue au chargement d'un attribut de la table dimension TD_J pour charger le contenu de a_{iL}, la valeur source peut être chargée telle qu'elle est, transformée ou bien une composante d'une opération de transformation composite.

4.3.2 Matrice de Fragmentation Horizontale Primaire et Dérivée(Tableau 2)

Une requête OLAP typique commence par l'application de l'opération algébrique de sélection sur les tables dimension et par la suite, applique les fonctions d'agrégats sur les mesures des tables des faits. Pour modéliser le processus de fragmentation des données logiques, nous proposons comme formalisme la matrice de fragmentation horizontale primaire et dérivée (MFHPD) (Tableau 2).

Un entrepôt de donnée relationnel peur être considéré comme une base de données relationnelles reliées par des contraintes d'intégrité référentielle. De ce fait, nous pouvons appliquer les techniques de fragmentation utilisées dans les bases de données relationnelles pour pouvoir fragmenter le modèle en étoile. La meilleure technique qui semble la plus adaptée à l'approche multidimensionnelle est celle de la fragmentation horizontale primaire et dérivée.

Dans ce qui, nous allons proposer les étapes à suivre pour remplir la matrice de fragmentation horizontale primaire et dérivée (Tableau 1) d'un modèle en étoile donné:

1. Appliquer l'algorithme de dénormalisation proposé par Noaman, A.Y., Barker, K. (1999)-2 permettant une réorganisation du schéma en flocon de neige selon les besoins du système décisionnel. L'algorithme de dénormalisation génère un modèle multidimensionnel dénormalisé en se basant sur des schémas de requêtes OLAP typique ;
2. Sélectionner les prédicats les plus pertinents, la sélection est basée sur une bonne connaissance de la base et de la sémantique des données ;
3. Optimiser la liste des prédicats en appliquant l'algorithme COMM_MIN proposé par Ozsu, M. T. , Valduriez, P.(1991). La liste optimisée permet de procéder à la phase de fragmentation primaire tout en assurant les contraintes de complétude et de disjonction. En sortie, l'algorithme nous fourni la liste des prédicats optimisée ;
4. Fragmenter la table des faits en appliquant l'algorithme de fragmentation horizontale primaire et dérivée proposé dans Noaman, A.Y., Barker, K. (1999)-2.
5. Optimiser le schéma de fragmentation abouti en sélectionnant le meilleur schéma de fragmentation. Dans ce sens, on pourra exploiter la technique d'optimisation à base d'algorithme génétique et de recuit simulé de Boukhalfa, K., Bellatreche, L. (2006).
6. La dernière étape consiste au remplissage de la MFHP pour toutes ces informations.

Un prédicat minterme MPipi (1< =i< =n, 1< =pi< =qi; avec n le nombre de site, q le nombre de requêtes par site) est une condition de restriction généralement appliquée sur les tables dimensions. Un prédicat minterme MPipi ne peut utiliser qu'une seule table dimension table TDJ (1< =j<v=k; avec k le nombre des tables dimensions du modèle en étoile et un ou plusieurs attributs ajLj (1< =l< =m; avec m le nombre d'attribut des la able dimension TDJ). Par exemple, un fragment primaire phf11 est généré par l'utilisation des attributs aJ, Lj

de la table TD1 par le prédicat minterme MP1,1. La liste des fragments horizontaux primaires résulte de l'application de la fonction *f*p qui prends en entrée la table dimension TDj, la liste des attributs concernés par la fragmentation {a1,l1,…,a1,L1} et les critères de fragmentation (FC) selon les prédicats minternes concernés. En sortie, le fragment horizontal primaire de la table dimension TDj. Un fragment horizontal primaire dérivé (dhf) résulte de l'application de la fonction *f*d qui prends en entrée la table des fait (FT), le fragment horizontal primaire (phf) et l'application de la conditionb de semi jointure (CSJ). En sortie, la liste des fragments horizontaux derivés de la table des faits. (Tableau2).

		Prédicats Mintermes	TABLES DIMENSIONS				
			$TD_1(a_{1,1},\dots a_{1,L1},\dots a_{1,m1})$		$TD_j(a_{j,1},\dots a_{jLj},\dots a_{j,mj})$		$TD_k(a_{k,1},\dots,a_{k,Lk},\dots a_{k,mk})$
SITES CIBLES	S_1	$PM_{1,1}$ … $PM_{1,p1}$ …. $PM_{1,q1}$	$Fhp_{11}=f_p(TD_1, \{a_{1,l1},.., a_{1,L1}\},CF)$ $Fhd_{11}=f_d(TF/TD, Fhp_{11}, CJ)$		…		…
	S_i	$PM_{i,1}$ … $PM_{i,pi}$ … $PM_{i,qi}$	$Fhp_{1i}=f_p(TD_1, \{a_{1,l1},.., a_{1,L1}\},CF)$ $Fhd_{11}=f_d(TF/TD, Fhp_{11}, CJ)$		$Fhp_{ji}=f_p(TD_j, \{a_{j,1},.., a_{j,Lj}\},CF)$ $Fhd_{ji}=f_d(TF/TD, Fhp_{ji}, CJ)$		…
	S_n	$PM_{n,1}$ … $PM_{n,pn}$ … $PM_{n,qn}$	…		…		$Fhp_{nk}=f_p(TD_k, a_{k,1},.., a_{k,Lk}\},CF)$ $Fhd_{nk}=f_d(TF/TD,$ $Fhp_{nk}, CJ)$

TAB. 2 – *Matrice Fragmentation horizontale primaire et dérivée*

4.3.3 Matrice d'Allocation (Tableau 3)

La matrice de fragmentation primaire et dérivée engendre un ensemble de données utilisées par les différents MD. Une données D_u (1<=u<=t ; avec t le nombre de données engendrées par la fragmentation) peut être une table des faits, une table dimension, un fragment dérivé ou un fragment primaire. Généralement dans un MD, une données (D_u) est utilisée exclusivement en consultation, toutes les opérations d'ajout, modification et suppression sont prises en charge par l'administrateur. Si une données est utilisée par un MD elle sera allouée à celui-ci et sera appelée Données Persistante (DP) si non, c'est une Données Absente (DA). La persistance d'une donnée dans un MD dépend de sa fréquence d'utilisation (F_u). Si celle-ci est égale à 0 à un instant donné, elle sera supprimée du site et passe à l'état DA. On déduit de ce fait, une matrice d'allocation (MA). (Tableau 3).

		TF / TD/ FHP/ FHD				
		D_1		D_u		D_t
DESTINATIONS	MD_1	$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD_1, t, F_u)$=DP/DA
	MD_i	$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD1, t, Fu)$=DP/DA
	MD_n	$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD_1, t, F_u)$=DP/DA		$f_a(D_1, MD_1, t, F_u)$=DP/DA

TAB. 3 – *Matrice Allocation*

4.3.4 Matrice Méta Base (Tableau 4)

La matrice Méta Base (MB) (Tableau 4) décrit pour chaque donnée allouée la source, les transformations subites et le MD concerné par l'allocation. Une transformation peut être élémentaire $f_{te(}\mathbf{A_{rz}}$ en utilisant un seul attribut source $\mathbf{A_{rz}}$ ou bien composite $f_{tc(}\mathbf{A_{1z}}, \mathbf{A_{1x}}, \mathbf{A_{r1}}, \mathbf{A_{s1)}}$ en utilisant plusieurs attributs sources.

Cette matrice est déduite à partir des matrices précédentes et fera l'objet d'un support d'aide à l'administrateur de l'ED réparti.

Destinations \ Sources			TS_1					TS_r					TS_s					TC
			A_{11}		A_{1z}		A_{1x}	A_{r1}		A_{rz}		A_{rx}	A_{s1}		A_{sz}		A_{sx}	
MD_1	$MLMD_1$	D_{11}	$f_{te(}A_{11)}$															
		D_{1u}	X				X	X					X					$f_{tc()}$
		D_{1t}																
MD_i	$MLMD_i$	D_{i1}																
		D_{iu}																
		D_{it}	X		X			X		X			X		X			$f_{tc()}$
MD_n	$MLMD_n$	D_{n1}																
		D_{nu}																
		D_{nt}																
TF		MI	X		X													
		Mp																
		Mq			$f_{te(}A_{1z)}$													

TAB. 4 – *Matrice Méta base*

5 Méthode d'évaluation d'un entrepôt de données réparti

Pour évaluer un ED réparti, nous allons proposer une méthode d'évaluation basée sur des modèles de coût. Pour ce faire, nous allons proposer une méthode d'évaluation, un ensemble de modèles de coût, un ensemble de concepts de base et un formalisme de présentation.

5.1 Modèles de coût

Les modèles de coût proposés sont les suivants :

Coût de chargement. Le coût de chargement est exprimé en terme de débit transféré sur le réseau pour le chargement ou le rafraîchissement des données de l'ED. La répartition d'un ED peut réduire le coût de chargement des données si les sources sont placées sur le site cible et risque d'augmenter les transferts sur le réseau si les sources des données deviennent plus éloignées aux MD. Dans ce cas, on préfère ne pas répartir et accéder à distance aux résultats ce qui pourra dans certains cas être plus rapide.
Coût des accès. Le coût est exprimé en terme d'entrée/sortie sur le réseau si un décideur consulte une donnée placée sur un MD distant ou même local.
Coût d'exécution des requêtes. Ce coût est exprimé par le temps d'exécution par seconde. Celui-ci permet de concrétiser l'efficacité de la répartition et valider les modèles de coût proposés par des résultats réels.

5.1.1 Coût de chargement des données

On dénote par TCh La taille de base d'un champ (TCh) dans une table des faits (par exemple à base de 4 Octets). Pour mesurer la taille d'une ligne chargée (TL), il suffit de multiplier la taille d'un champ par le nombre de champs (NBCh) qui est en faite, le nombre des mesures + le nombre de champs clés.

$$TL = TCh \text{ en Octet} * NBCh$$

Pour calculer la Débit Transféré (DT) pour un instant t, la taille d'une ligne de la table des faits est multipliée par le nombre de lignes chargées à l'instant t (NLCh) par l'ETL.

$$DT_t = TL_t * NLCh_t$$

Le débit transféré peut être ressenti sur le système selon le taux de transmission (Tx) sur le réseau exprimé en Bit/Seconde.

$$CC = DT_t * Tx$$

Le CC peut être réévalué selon les caractéristiques du réseau, un réseau donné peut être caractérisé par un Indicateur de Performance (IP), qui peut prendre la valeur de 1 pour un réseau en bon état, 2 pour un état moyen (Coupures assez fréquentes, pannes, etc.) et 3 pour un réseau en mauvaise état (Blocage très fréquents, saturation très fréquente, perte de données lors des transmissions, etc.). On pourra alors majorer le coût de chargement selon l'Indicateur de Performance du Réseau (IPR), Le coût de chargement devient :

$$CC = DT_t * Tx * IPR$$

Et pour tous les fragments de l'ED elle devient :

$$\mathbf{CC}= \sum_{f=1}^{F} (TL * NLCh * IPR) / Tx$$

5.1.2 Coût des accès

Nous allons appliquer la fonction objective proposée dans *Ezeife, C.I., Zheng, J.. (1999)* pour mesurer en terme d'entrée/Sortie les accès locaux ou à distances entre les différents MD. Pour les coûts locaux (CL) c'est-à-dire, les coûts d'accès aux données allouées localement, nous allons utiliser la formule suivante :

$$\mathbf{CL}= \sum_{f=1}^{F} \sum_{t=1}^{T} (FU * nlct * (1 - nlct / nltf)$$

Avec FU les fréquences d'utilisation des données, nlct le nombre de lignes consultées et nltf le nombre de ligne totale du fragment consulté. La fonction montre bien que le nombre de lignes consultées est majoré par (1-nlct/nltf) si le fragment alloué localement n'est pas consulté en totalité. Ce coût augmente à chaque fois qu'il y a des données allouées sur un site mais non utilisées par les décideurs. Pour les coûts distants (CD), nous allons utiliser la formule suivante :

$$\text{CD} = \sum_{f=1}^{F}\sum_{t=1}^{T}(FU * nlct * (nlct/nltf)$$

On remarque que le coût des accès à distance est majoré par (nlct/nltf) c'est-à-dire, si le nombre de lignes consultées à distance augmente le coût des accès à distance augmente.

5.1.3 Coût d'exécution des requêtes

Le coût d'exécution des requêtes locales et distantes permet de concrétiser les résultats obtenus par la démarche proposée. Le coût d'exécution des requêtes OLAP est exprimé par la somme d'exécution des requêtes OLAP locales et distantes.

$$\textbf{CL=} \sum_{f=1}^{F}\sum_{t=1}^{T}(Tempsdexécution)/s)$$

$$\textbf{CD =} \sum_{f=1}^{F}\sum_{t=1}^{T}(Tempsdexécution/s)$$

5.2 Méthode d'évaluation

Les besoins de répartition sont ressentis lorsque la compagnie est répartie et que les décideurs eux même sont répartis sur les différents sites. L'idée de la répartition va nous faire gagner énormément en terme d'un stockage différé, requêtes parallèles et chargement plus souples de l'ED. Certes, la répartition est déconseillée si le coût de chargement ou de rafraîchissement devient élevé et si les sources sont éloignées du lieu de l'entrepôt dans ce cas, on préfère centraliser l'entrepôt, procéder aux requêtes distantes et transférer seulement les résultats. Ceci sera largement plus intéressant que de transférer des milliers de lignes sur un réseau étendu. De ce fait, une évaluation est nécessaire avant de procéder à la répartition, parce que même si la compagnie est répartie géographiquement, on préfère centraliser l'ED de données si les chargements à distance vont nous coûter cher ou bien, que les coûts des accès aux nouveaux sites vont devenir plus élevés. Pour ce faire, nous allons proposer un ensemble de modèles de coûts pour évaluer le nouveau coût des chargements et les nouveaux coûts des accès à distances. Le temps d'exécution des requêtes reste toujours une bonne alternative pour concrétiser les modèles de coûts utilisés. Pour ce faire, nous proposons la méthode d'évaluation suivante :

- Etape 1 : Calcul des coûts de chargement et les coûts des accès aux données dans un cas centralisé et réparti.
- Etape 2 : Evaluation et comparaison des coûts. Ensuite, prise de décision
- Etape 3 : Si l'analyse a aboutie à de meilleurs coûts on lance le processus de fragmentation

– Etape 4 : Calcul et suivi des coûts d'exécution des requêtes OLAP et évaluation finale des résultats

5.3 Concepts de base

Pour faciliter le processus d'évaluation d'un ED réparti, nous allons proposer un nouveau formalisme de modélisation basée sur un ensemble de matrices multidimensionnelles.

Matrice des coûts des accès locaux et distants. Cette matrice englobe les coûts des accès locaux et distants à un fragment particulier. L'administrateur définit un seuil d'acceptation qu'on a choisi d'appelé 'seuil_coût_accès' permettant de juger si un coût d'accès à distance est acceptable ou non. Cette matrice est mise à jours à chaque fois qu'il y a des changements aux nombres de lignes consultées ou bien au niveau des calculs des coûts. Pour ce faire, on a établi un compteur_ligne(Fi) permettant de calculer le nombre de lignes d'un fragment Fi, et compteur_ligne_consultées(Fi, Rj) permettant de calculer le nombre de lignes consultées par la requête Rj pour le fragment Fi.

Matrice des coûts de chargement. Cette matrice englobe les coûts de chargement de chaque fragment. L'administrateur définit un seuil d'acceptation qu'on a choisi d'appelé 'seuil_coût_chargement' permettant de juger si un coût de chargement d'un fragment dans un site est acceptable ou non. Cette matrice est mise à jours à chaque fois qu'il y a des rafraîchissements aux nombres de lignes chargées ou bien au niveau des calculs des coûts. Pour ce faire, on calcule le cout_chargement(Fi,Sj) dans un fragment Fj. Le compteur_ligne(Fi) sera lancé et les mises à jours nécessaires au coût de chargement du fragment Fi seront établies. Dans le cas, où un coût de chargement dépasse le 'seuil_coût_chargement', l'administrateur doit réviser l'allocation des fragments aux différents sites.

Matrice des coûts d'exécution. C'est une matrice de suivi des différents calculs, elle permet de concrétiser la performance de l'allocation des fragments selon les modèles de coûts proposés. Pour ce faire, l'administrateur doit définir un seuil d'acceptation du temps d'attente d'une réponse appelé 'seuil_coût_d'exécution' généralement défini par l'administrateur en collaboration avec les décideurs. Si les temps de réponses sont inacceptables, une révision du schéma d'allocation est nécessaire et quelques réplications de fragments peuvent être envisagées.

Matrice Caractéristiques du Réseau. Cette matrice englobe un indicatif descriptif de l'état du réseau entre chaque deux sites du réseau. L'état du réseau est décrit pat un Indicateur de Performance (IP) qui caractérise chaque portion du réseau. Les différents IP sont regroupés dans la matrice des Caractéristiques du Réseau (CR).

Matrice Taux de Transmission. Cette matrice englobe un indicatif descriptif du taux de transmission entre chaque deux site du réseau. La consultation d'un MD à distance passe tout d'abord par le réseau de l'entreprise, le Taux de Transmission (TT) sur le réseau est nécessaire pour pouvoir anticiper les coûts de chargement à distance et le coût de réponse des requêtes OLAP à distance. Les Taux pour toutes les portions du réseau sont récapitulés dans cette matrice (Tableau 7). Une donnée au niveau d'un MD est chargée soit à partir du site local, soit à partir d'un site distant.

Matrice chargement. Cette matrice englobe le totale de chargement en terme de lignes chargées entre chaque deux sites du réseau. Elle décrit pour chaque MD, la source de chargement $Site_i$ et le nombre de lignes chargées.

5.4 Formalisme

Dans ce qui suit, nous allons proposer un formalisme de présentation des différents concepts de bases ainsi qu'aux différents coûts calculés à travers des matrices multidimensionnelles. Ces nouveaux concepts vont nous permettre de mieux exploiter les différents calculs pour l'évaluation de la répartition d'un ED.

5.4.1 Matrice des coûts des accès locaux et distants

Cette matrice englobe pour chaque données Du (1<=U<=t ; avec t le nombre de fragment), le nombre de lignes consultées localement qui est désigné par (NLCL), le nombre de lignes consultées à distance est désigné par NLCD. $(CL+CD)_{D1}$ désigne le coût des accès à D1 par tous les traitement d'un site Si et (CL+CD) $t_{1,1}$ désigne le coût total engendré par le traitement $t_{1,1}$ (Tableau 5)

	D_1	D_u	D_t	CL	CD	CT
Nombre lignes	NL_{D1}	NL_{D2}	NL_{Dt}			
$t_{i,1}$	NLCL	NLCL	NLCL			(CL+CD) $t_{1,1}$
$t_{i,pi}$	NLCL	NLCL	NLCL			(CL+CD)$t_{1,p1}$
$t_{i,qi}$	NLCD	NLCL	NLCD			(CL+CD)$t_{1,q1}$
CL						
CD						
CT	$(CL+CD)_{D1}$	$(CL+CD)_{Du}$	$(CL+CD)_{Dt}$			CFT

TAB. 5 – *Matrice coûts d'accès locaux et distants d'un site Si*

5.4.2 Matrice Caractéristiques du Réseau

La matrice Caractéristique du Réseau englobe les Indicateurs sur l'état du Réseau Local (IRL) et/ou les Indicateurs du Réseau Distant (IRD) ; elle permet de décrire le réseau de la compagnie en terme d'état et de performance. (Tableau 6)

5.4.3 Matrice Taux de Transmission

La matrice Taux de Transmission (TT) englobe les Taux de Transmission sur le réseau Local (TTL) et/ou les Taux de Transmission sur les réseaux Distant (TTD) ; elle permet de décrire l'état du réseau de la compagnie en terme de taux de transfert. (Tableau 7)

5.4.4 Matrice chargement

Cette matrice englobe le Nombre des Ligne Chargées Localement sur un site (NLChl) ou à Distance entre deux sites (NLChd). (Tableau 8).

5.4.5 Matrice des coûts de chargement.

Cette matrice englobe pour chaque site le Coût des Chargements Locaux (CCL), c'est-à-dire dont les sources sont locales au site. Et/ou Coût Chargement Distant (CCD). La colonne Coût de Chargement englobe les coûts de chargement Totaux locaux et distants pour chaque site. (Tableau 9)

	$Site_1$	$Site_i$	$Site_n$
$Site_1$	IRL	IRD	IRD
$Site_i$	IRD	IRL	IRD
$Site_n$	IRD	IRD	IRL

TAB. 6 – *Matrice CR*

	$Site_1$	$Site_i$	$Site_n$
$Site_1$	TTL	TTD	TTD
$Site_i$	TTD	TTL	TTD
$Site_n$	TTD	TTD	TTL

TAB. 7 – *Matrice TT*

	$Site_1$	$Site_i$	$Site_n$
S_1	NLChl	NLChd	NLChd
S_i	NLChd	NLChl	NLChd
S_n	NLChd	NLChd	NLCh1

TAB. 8 – *Matrice Chargement*

	$Site_1$	$Site_i$	$Site_n$	Coût de Chargement
$Site_1$	CCL	CCD	CCD	CCT du $Site_1$
$Site_i$	CCD	CCL	CCD	CCT du $Site_i$
$Site_n$	CCD	CCD	CCL	CCT du $Site_n$

TAB. 9 – *Matrice Coûts de Chargement locaux et distants*

5.4.6 Matrice des coûts d'exécution

Cette matrice englobe le coût d'exécution des requêtes sur chaque magasin de données, chaque requête utilise un ou plusieurs données et engendre un Coût d'Exécution (CE), cette matrice est très importante pour concrétiser les modèles de coûts utilisés, ainsi que pour le suivi des performances de l'exploitation de l'ED. (Tableau 10)

		TF / TD/ FHP/ FHD					CE
		D_1		D_u		D_t	
MD_1	Req_{11}						Temps d'exécution Req_{11}
	Req_{1l}						Temps d'exécution Req_{1l}
	Req_{1m}						Temps d'exécution Req_{1m}
MD_i	Req_{ij}						Temps d'exécution Req_{ij}
	Req_{jl}						Temps d'exécution Req_{jl}
	Req_{jm}						Temps d'exécution Req_{jm}
MD_n	Req_{1j}						Temps d'exécution Req_{1j}
	Req_{nl}						Temps d'exécution Req_{nl}
	Req_{nm}						Temps d'exécution Req_{nm}

TAB. 10 – *Matrice coût d'exécution*

6 Exemple d'application

Nous prenons comme exemple un ED pour « l'activité des ventes ». Les charges des requêtes, ainsi que les tables consultées et les données utilisées sont présentées dans les tableaux ci-dessous (Tableau11, Tableau12, Tableau13, Tableau14,Tableau 15). Dans le tableau 16 nous utilisons la matrice d'utilisation des données proposée dans Tekaya, K, Abdellatif, A. (2004), qui décrit pour chaque requête les fragments utilisés, ainsi que les fréquences d'utilisation.

	Site1	Site2	Site3	Tables	
R1	30	30	30	VENTE, Région, TEMPS	Analyse des ventes par jour
R2	4	30	30	VENTE, TEMPS, PRODUIT	Analyse des ventes par produit
R3	0	0	35	PRODUIT	Analyse lancement d'un nouveau produit
R4	1	4	4	CLIENT, VENTE, TEMPS	Analyse de l'évolution des clients
R5	40	2	2	VENTE, PRODUIT, REGION	Analyse des ventes par région

TAB. 11 – *Fréquences des requêtes et tables utilisées*

	$Site_1$	$Site_2$	$Site_3$
$Site_1$	1	2	3
$Site_2$	2	1	2
$Site_3$	3	2	1

TAB. 12 – Matrice IP

	$Site_1$	$Site_2$	$Site_3$
$Site_1$	200	100	100
$Site_2$	100	400	200
$Site_3$	100	200	400

TAB. 13 – Matrice TX

	$Site_1$	$Site_2$	$Site_3$
$Site_1$	300	390	3453
$Site_2$	600	3455	4353
$Site_3$	4000	234	3453

TAB. 14 – Matrice Chargement

		TABLES				
		Région	**Vente**	**Client**	**Temps**	**Produit**
SC	SITE1	Région	VENTE			PRODUIT
	SITE2	FP1, [CJ2.1 :N°in (1, 3)]	FD1,CJ2.1	FD3, [CJ2.2 :NR=2)	FD5, [CJ2.3 :date in FD1.IDT)	FD7,[CJ2.4: NR=2)]
	SITE3	FP2, [CJ3.1 :N°in (1, 2)]	FD2, CJ3.1	FD4, [CJ 3.2 :NR=3)	FD6, [CJ3.3 :date in FD2.IDT)	PRODUIT

TAB. 15 – *Matrice de fragmentation horizontale Primaire et dérivée*

			TABLES										
			Région	FP1	FP2	Vente	FD1	FD2	FD3	FD4	FD5	FD6	Produit	FD7
SITES CIBLES	S1	Req1	30	30	30	30	30	30	0	0	30	30	0	0
		Req2	0	0	0	4	4	4	0	0	4	4	4	4
		Req3	0	0	0	0	0	0	0	0	0	0	0	0
		Req4	0	0	0	1	1	1	1	1	1	1	0	0
		Req5	40	40	40	40	40	40	0	0	0	0	40	40
	Total des utilisations		70	70	70	75	75	75	1	1	35	35	44	44
	S2	Req1		30			30		0		30		0	
		Req2		0			30		0		30		30	
		Req3		0			0		0		0		0	
		Req4		0			4		4		4		0	
		Req5		2			2		0		0		2	
	Total des utilisations			32			66		4		64		32	
	S3	Req1			30			30		0		30	0	
		Req2			0			30		0		30	30	
		Req3			0			0		0		0	35	
		Req4			0			4		4		4	0	
		Req5			2			2		0		0	2	
	Total des utilisations				32			66		4		64	67	

TAB. 16 – *Matrice d'utilisation des données*

.

7 Résultats expérimentaux

Nous allons dans un premier temps, étudier les alternatives de répartition d'un ED et leur impacts sur les coûts calculés. Pour ce faire, nous allons traiter trois cas : le cas d'un ED centralisé, dupliqué dans chaque site et réparti géographiquement.

7.1 Cas d'un ED centralisé

On a constaté que la centralisation de l'ED a engendré un coût énorme au niveau de la table des faits, et les coûts distants au serveur central. La table des faits peut être énorme de point de vu taille. L'accès à la table des faits par les sites distants devient une opération laborieuse puisqu'il faut procéder à l'exécution d'une requête distante. (Figure2)

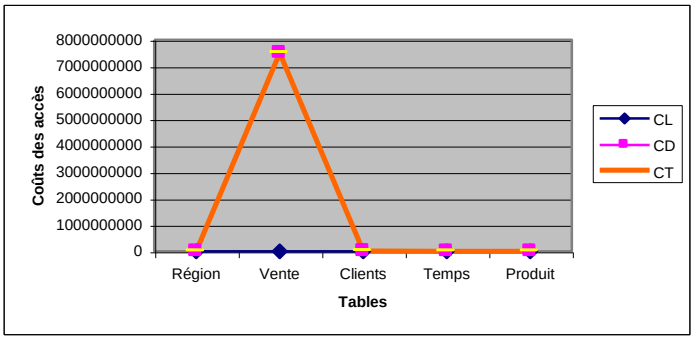


FIG 2 – *Coût de centralisation de l'ED dans le site 1*

7.2 Fragmentation de l'ED

On a remarqué que la fragmentation de l'ED selon les fréquences d'utilisation et les priorités des sites a engendré un coût total des accès négligeable par rapport aux autres cas. En conclusion, dans le cas des sociétés réparties géographiquement, une bonne fragmentation de l'ED pourra optimiser sa mise en place et son exploitation par la suite. Une telle fragmentation a minimisé les consultations à distance et aura comme conséquence une meilleure organisation physique des données dans le contexte réparti. (Figure 3)

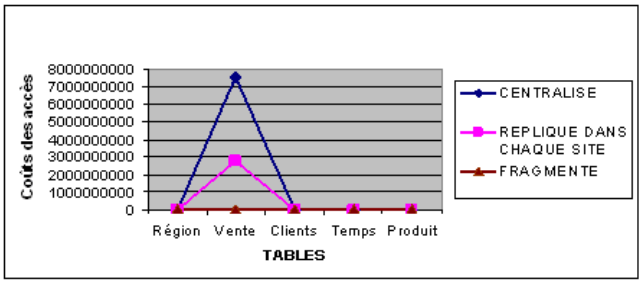


FIG. 3 – *Comparaison des trois approches*

8 Conclusion

Nous avons constaté à travers la littérature que la répartition d'un entrepôt de données en magasin de données, aura comme conséquence le rapprochement des données aux décideurs, l'optimisation du temps de réponse des requêtes décisionnelles et la répartition du stockage des données sur plusieurs serveurs. De ce fait, nous nous sommes fixés comme objectif, la proposition d'une approche de modélisation les composants de son système réparti. Par la suite, nous avons développé des modèles de coûts et nous avons amené des expérimentations et nous avons montré que la fragmentation peut perdre tous ces apports une fois les besoins des utilisateurs varient ou que de nouveaux besoins apparaissent sur le système. Ces modèles de coûts peuvent être très utiles à l'administrateur pour l'analyse de l'efficacité du schéma de répartition retenu. Comme perspectives, nous proposons l'automatisation d'un processus de contrôle basé sur ces modèles de coût permettant d'analyser et de détecter si un schéma de répartition entre dans une phase de dégradation. Nous pouvons aussi de point de vue logique étudier l'impact de la fragmentation de la table des faits sur la table d'agrégat au niveau du modèle en étoile.

9 Références

Bellatreche L., Boukhalfa, K. (2005. La fragmentation dans les entrepôts de données: une approche basée sur les algorithmes génétiques. Revue des Nouvelles Technologies de l'Information. 141-160.

Bellatreche, L., Kamalakar K., Mukesh M.(2002). Some issues in design of data warehousing systems. Data warehousing and web engineering. IRM Press Hershey, PA, United States. 22 – 76. 2002.

Boukhalfa K., Bellatreche, L. (2006). Combinaison des algorithmes génétiques et de recuit simulé pour la conception physique des entrepôts de données. INFORSID.

Boukhalfa, K., Bellatreche, L., Richard, P. (2008). Horizontal Partitioning in Data Warehouse: Hardness Study, Selection Algorithms and Validation on ORACLE10G. Dawak.

Chakravarthy, et al. (1992). An objective function for vertically partitioning relations in distributed databases and its analysis. Distributed and Parallel Databases.

Ezeife, C.I., Pinakpani, D. (2003). Incremental horizontal fragmentation of database class objects. 239-245

Ezeife, C.I., Zheng, J. (2001). Dynamic database object horizontal fragmentation. Natural science and engineering research council of Canada under an operating grant (OGP-0194134) and a University of Windsor grant.

Ezeife, C.I., Zheng, J. (1999). Measuring the performance of database object horizontal fragmentation schemes. Proceedings of the 3rd IEEE international database engineering and Applications Symposium IEEE press.

Furtado, P. (2006): Efficient and Robust Node-Partitioned Data Warehouses. In Data Warehouses and Olap: Concepts, Architectures and Solutions by Robert Wrembel (Editor), Christian Koncilia (Editor.). ISBN-13: 978-1599043647.

Ladjel B., Schneider, M., Lorinquer, H., Mohenia, M. (2004). Bringing together partitioning materialized views and indexes to optimize performance of relational data warehouse. Proceeding of the international conference Data Warehousing and Knowledge Discovery. 15-25.

Noaman, A.Y., Barker, K. (1997). Distributed data warehouse architectures. Journal of Data Warehousing.

Noaman, A.Y., Barker, K. (1999). A Horizontal Fragmentation Algorithm for the fact relation in a Distributed Data Warehouse. Proceeding of the Eigth International Conference on Information and Knowledge Management 154-161.

Noaman, A.Y., Barker, K. (1999)-2. Distributed data warehouse architectures and design. Fourteenth Internationnal Symposium on Computer and Information Sciences.

Noaman, A.Y., Barker, K. (2000). Hierarchically Distributed Data warehouse.

Ozsu, M. T., Valduriez, P. (1991): Principles of Distributed Database Systems. Prentice Hall.

Tekaya, K, Abdellatif, A. (2004). Modélisation de la répartition des données d'un data warehouse. Eighth Maghrebian Conference on Software Engineering and Artifical Intelligence.

Tekaya, K., Abdellatif, A., (2005). Modélisation de la répartition des données d'un data warehouse'. Revue Permanente en ligne des utilisateurs des Technologies de l'Information et de la Communication.

Summary

Usually, data warehouse are built and owned by centrally coordinated organizations. A data warehouse stores large volumes of data which are used frequently by decision support applications. Data warehouse are usually dedicated to the processing of data analysis and decision support queries (OLAP queries). These queries are much more complex, consequently the response time is much higher. A lot of work has been done to speed up the OLAP query processing in data warehouse like materialized view and advanced indexes. Another techniques witch is proving its efficacy in theses last years is the data fragmentation technique, starting with some fragmentation criterions, the data warehouse is partitioned into many data marts. We choose the context of distributed data warehouse. Theses criterions are done according to the decisional request, their frequencies of use and according to the network characteristics. In this paper, we propose some basic model to apply the fragmentation methodology in a data warehouse star schema. We propose an analytical cost model to evaluate a distributed data warehouse efficacy. Finally, we conduct some experiments to concretize the utility of fragmentation to distribute a data warehouse into data marts.

Sécurité des entrepôts de données - état de l'art -

Nouria Harbi , Maaroufi Ghanem, Omar Boussaid

Laboratoire ERIC, Université Lumière – Lyon 2
5 avenue Pierre Mendès-France
69676 Bron Cedex
{nouria.harbi, omar.boussaid}@univ-lyon2.fr, ghaska@hotmail.fr

Résumé.
Les entrepôts de données, de par leur nature même créent un conflit au niveau de la sécurité. D'une part, l'objectif de chaque entrepôt de données est de rendre disponibles, compréhensibles, et faciles d'accès des données pertinentes. La plupart des organisations ont non seulement investi dans d'importants réseaux pour relier des centaines, voir des milliers d'ordinateurs qui ont accès à des entrepôts de données, mais ont également multiplié les liens avec internet. D'autre part, le monde semble être rempli de hakers, crakers, spooks et industriels qui ne cessent de monter des attaques sur les réseaux, les systèmes matériels, et en définitif, les entrepôts de données. L'utilisation des réseaux informatiques lors d'un processus d'entreposage (acquisition, stockage et accès) augmente les risques d'attaques et rend les systèmes décisionnels de plus en plus vulnérables. Nous constatons que la sécurité reste le parent pauvre dans un projet d'entreposage de données, celle-ci se limite dans la plus part des cas à la sécurité des infrastructures, de la logistique et aux dispositifs disponibles au niveau des logiciels utilisés. Ainsi, la sécurité dans un entrepôt de données lors de sa conception passe souvent à travers les mailles du filet et demeure très insuffisante pour garantir l'intégrité, la confidentialité et la disponibilité des entrepôts de données.
L'objectif de cet article est de réaliser une synthèse des travaux de recherche dans le domaine de la sécurité des entrepôts de données, de présenter une étude comparative de ces travaux et d'aborder l'offre de sécurité au niveau des outils existants dans le domaine des systèmes d'information décisionnels (SID).

1 Introduction

L'entrepôt de données sert à produire de nouvelles informations sur le pilotage de l'entreprise et sur ses clients. Nous constatons que la plupart des responsables se préoccupent plus des aspects marketing des SID, et moins des problèmes liées à la sécurité. Certains dirigeants d'entreprise pensent que la sécurité se limite à celle des réseaux et se contentent des dispositifs disponibles au niveau des logiciels utilisés. Il est d'autant plus difficile de se détourner de ces avantages pour se pencher sur les risques et sur les vulnérabilités.
Sans être alarmiste, le monde des entrepôts de données a jusqu'à présent éludé les problèmes de sécurité. La plus part des entrepôts de données sont construits avec peu ou pas de considération accordée à la sécurité durant la phase de développement Slemo Warigon [SW97].

Le rôle du responsable d'un entrepôt de données est quelque peu paradoxal : il doit à la fois diffuser les informations et les protéger. D'une part, il sera jugé sur la facilité avec laquelle les utilisateurs accèdent aux données ; d'autre part, il porte la responsabilité en cas de pertes de données ou si des données sensibles tombent entre de mauvaises mains. Le responsable de l'entrepôt de données est le gardien des « joyaux de la couronne » de l'entreprise (les données bien sûr) et sera tenu pour responsable s'ils sont perdus, volés Torsten Priebe et al, [TP00].

L'objectif de cet article est de montrer la problématique du conflit entre les spécificités du SID et les exigences de sécurité à travers une étude bibliographique des travaux de recherche dans ce domaine ainsi que des outils commerciaux existants.

Cet article est organisé de la façon suivante. Nous commençons par la présentation des spécificités d'un entrepôt de données dans la section 2. Puis, dans la section 3, nous aborderons la sécurité des entrepôts de données et de leurs environnements. Nous présenterons ensuite un panorama des travaux de recherche dans ce domaine ainsi que des outils utilisés suivi d'une étude comparative. Nous terminerons par une conclusion et des perspectives.

2 Entrepôts de données

L'entrepôt de données, *(ou Data Warehouse),* est un concept spécifique de l'informatique décisionnelle, issu du constat suivant : les données de production également appelée « données transactionnelles », notamment les progiciels de gestion ne se prêtent pas à une exploitation dans un cadre d'analyse décisionnelle. Les systèmes de production, où sont générés les données de production, sont en effet construits dans le but de traiter des opérations qui peuvent impliquer différents métiers de l'entreprise et surtout, ne se préoccupent pas de leurs historisation dans le temps. À l'inverse, les systèmes décisionnels doivent permettre l'analyse par métiers ou par sujets et le suivi dans le temps d'indicateurs de performance d'une activité. Il est donc souvent indispensable de séparer ces deux mondes et de repenser les schémas de données, ce qui implique l'unification des différents gisements de données de l'entreprise en un entrepôt de données global ou un magasin de données *(datamart)* contenant des données dédiées à un sujet (ou à un métier).

La constitution d'entrepôt de données est une réponse au problème de l'intégration d'une grande quantité de données variées, relatives à un certain domaine d'application, et stockées physiquement dans différentes sources de données. L'entrepôt de données regroupe, sous une forme exploitable par des traitements utiles pour l'aide à la décision, les informations extraites de ces sources et qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine. En plus de dix ans d'existence, les entrepôts de données se sont imposés comme une solution rentable pour faire face aux besoins des entreprises en termes de capitalisation de connaissances et d'aide à la décision.

2. 1 Caractéristiques

Un entrepôt est défini comme "une collection de données intégrées, orientées sujet, non volatiles, historiées, résumées et disponibles pour l'interrogation et l'analyse" Bill Inmon [BI96]. L'entrepôt de données stocke des données nécessaires à la prise de décision. Il est alimenté et mis à jour via des extractions de données portant sur les bases de production qui sont considérées dans la chaîne décisionnelle comme les "sources de données" (Fig.1).

Les données de l'entrepôt proviennent de différentes sources éventuellement hétérogènes. Dans le contexte d'un entrepôt de données, "décrire une donnée" consiste principalement à indiquer comment l'obtenir à partir des sources. Les métadonnées jouent un rôle important dans les mécanismes d'extraction, de rafraîchissement et d'intégration, et également dans la présentation d'une vision globale des données à l'utilisateur. L'intégration des données nécessite des assertions de correspondance entre les tables des sources et les vues de l'entrepôt.
L'entrepôt a pour objectif final l'analyse des données en vue de la prise de décision. Différents types d'analyse peuvent être réalisés : analyses statistiques, fouille de données, etc. Il existe également l'analyse en ligne OLAP. Il s'agit en l'occurrence d'une navigation dans les données. Cette analyse peut être qualifiée d'exploratoire. Le principe général est d'arriver au cours de la navigation à détecter des points intéressants qu'on essaye de décrire, d'expliquer en naviguant par exemple en allant chercher davantage de détails. Le rôle de l'utilisateur est ici central puisque c'est lui qui réalise la navigation. Celle-ci nécessite une connaissance du domaine afin d'être en mesure de savoir si les valeurs des mesures sont intéressantes ou non.

2.2 Composants d'un processus d'entreposage de données

Le processus d'entreposage (Fig. 1) est composé de plusieurs zones :
- **La zone de préparation ou ETL** (extraction/transformation/chargement) est une zone qui est à la fois une zone de préparation des donnés et de stockage. Un point très important est d'interdire aux utilisateurs l'accès à la zone de préparation des données qui ne fournit aucun service de requête ou de présentation

Extraction : C'est une opération qui consiste à lire et interpréter les données sources et de les copier dans la zone de préparation en vue de manipulations ultérieures.
Transformation : une fois dans la zone de préparation, les données peuvent être transformées de nombreuses manières, par exemple nettoyées (correction orthographique, résolution de conflits de domaine, traitement de problèmes liés aux éléments manquants, conversion vers des formats standards), combinées à partir de sources multiples, dédoublées s'il y a lieu et pourvues de clés propres à l'entrepôt de données. Ces transformations sont le prélude au chargement dans la zone de présentation.
Chargement : C'est une opération de chargement des données dans l'entrepôt qui est théoriquement la destination ultime des données.
- **La zone de présentation des données :** cette zone est l'entrepôt de données tel qu'il est perçu par la communauté des utilisateurs. C'est le lieu où les données sont organisées, stockées et consultées sous forme de schémas dimensionnels. Si la zone de présentation utilise une base de données relationnelle, les données sont dans des tables. Si elle utilise une base de données multidimensionnelle utilisant une technologie OLAP, les données sont alors stockées dans des cubes.
- **Les outils d'accès aux données :** nous appliquons ce terme général à un ensemble de moyens fournis aux utilisateurs pour exploiter les données de la zone de présentation en vue de prendre des décisions basées sur des analyses. Par définition, tous les outils d'accès aux données font des requêtes sur les données de la zone de présentation.

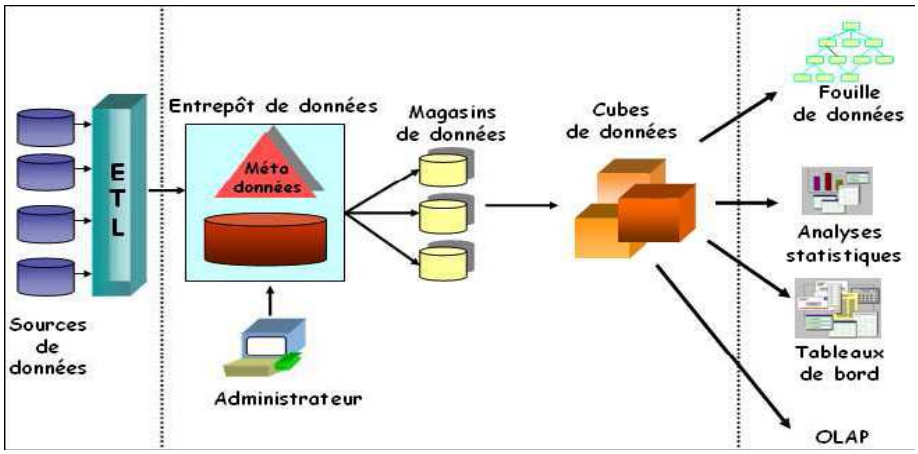


Fig. 1 - Architecture décisionnelle

Un entrepôt de données est généralement construit selon l'architecture suivante :

- Un serveur d'entrepôt (serveur de données)
- Un serveur OLAP (de type HOLAP, MOLAP ou ROLAP)
- Un client
- Un outil pour l'exécution des requêtes
- Un outil pour l'analyse des données

3 Sécurité des entrepôts de données et de leur environnement

Nous constatons que la pertinence des entrepôts de données et de l'analyse en ligne (OLAP) pour une organisation du système d'aide à la décision a rapidement augmenté au cours des dernières années. Et qu'en même temps, une sensibilité à la sécurité de l'information et de la vie privée a évolué. Cependant, il n'y a pas beaucoup de démarches pour traiter ces deux champs ensemble.

Les entrepôts de données, de par leur nature même, créent un conflit au niveau de la sécurité. D'une part l'objectif de chaque entrepôt de données est de rendre disponibles, compréhensibles, et facile d'accès des données précieuses, d'autres part l'utilisation via les réseaux informatiques augmente les risques d'attaques sur les réseaux, la messagerie et les SID.

Ces menaces sont provoquées par plusieurs types d'individus : des hackers, crackers, des industriels, la concurrence, un jeune qui considère l'intrusion dans un système informatique comme un jeu, les adultes tentent de manipuler des informations commerciales frauduleuses, de simples curieux, des employés qui ne peuvent pas résister à la tentation d'explorer…

Jusqu'à présent, les équipes chargées de la mise en place d'un entrepôt de données s'occupent plus du repérage des données et du choix de matériels et de logiciels alors qu'un plan de sécurité global de base n'est jamais fait. Ainsi, la sécurité de l'entrepôt de données lors de sa conception passe souvent à travers les mailles du filet et demeure très insuffisante pour garantir l'intégrité, la confidentialité et la disponibilité des entrepôts de données.

Nous constatons alors que la sécurité reste le parent pauvre dans un projet d'entreposage de données, celle-ci se limite dans la plus part des cas à la sécurité des infrastructures, de la logistique et aux dispositifs disponibles au niveau des logiciels utilisés. L'utilisation des

réseaux informatiques, d'Internet lors d'un processus d'entreposage (acquisition, stockage et accès) augmente les risques d'attaques et rend les systèmes décisionnels de plus en plus vulnérables.

3. 1 Exigences de sécurité

L'organisation internationale de normalisation (International Organization for Standardization), ou ISO n'a pas défini des normes spécifiques pour la sécurité des entrepôts de données. Cependant elle préconise pour la sécurité des systèmes d'informations classiques les sept services suivants : (1) l'authentification de l'entité homologue, (2) le contrôle d'accès, (3) la confidentialité des données, (4) le secret des flux, (5) l'intégrité des données, (6) l'authentification de l'origine, (7) la non répudiation.
Nous avons fait le choix de nous limiter à quatre services pour la sécurisation des entrepôts de données, ceux qui permettent de garantir la sécurité et la fiabilité des données du SID :
- ***La confidentialité*** : caractère de données dont la diffusion doit être limitée aux seules personnes autorisées et qui ont accès aux ressources du système.
- ***L'intégrité*** : propriété associée aux données que, pendant leurs traitements, leur conservation en mémoire ou leur transport par voix électronique ne subissent aucune altération ou destruction volontaire ou accidentelle. L'intégrité garantie que les ressources de données ne soient pas corrompues.
- ***La disponibilité*** : faire en sorte que les données soient accessibles en permanence avec un temps d'attente raisonnable.
- ***L'authenticité*** : caractère d'une information dont l'origine et l'intégrité soient garanties. Ce caractère permet de gérer correctement les accès aux données sensibles.

3.2 Sécurité de l'environnement de l'entrepôt de données

La sécurité est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité contre les menaces accidentelles ou intentionnelles.
Généralement comme pour tout système d'information et plus particulièrement dans un contexte d'entrepôt de données, il est indispensable d'assurer la sécurité de son environnement. Elle porte sur la sécurité des infrastructures et de la logistique en plus de la sécurité des applicatifs et des droits d'accès. L'ampleur des problèmes de sécurité est étonnante et sous estimée, Dés octobre 1996, Ernst & Young déclaraient, dans Information Week, que « 78% des responsables de la sécurité informatique font état de pertes financières subies par leurs entreprises à cause de brèches (failles) dans la sécurité de leur système d'information. Plus de 25% annoncent des pertes supérieures à 25 000 dollars, et les infractions internes sont responsables de prés de 32% des pertes ». En 1997, l'Aberdeen Group, utilisant des unités de mesures légèrement différentes, faisait état des principales causes de mise en danger des informations, Ralph Kimball [RK07] :

- erreur humaine : 35%
- omission humaine : 25%
- employés mecontents : 15%
- intervenants externes : 10%
- incendie : 7%

- inondation : 5%
- autres catastrophes naturelles : 3%

D'après ces conclusions, les sources de vulnérabilité sont variées : la menace vient pour une bonne part de l'intérieur et n'est pas exclusivement le fait de « hackers de l'Internet ». Charles Pfleeger, dans son livre Security in Computing (Prentice- Hall, 1989), divise le groupe des employés mecontents et des intervenants externes cités plus haut en trois catégories : amateurs, crackers et Criminels de carrière, Ralph Kimball [RK07].

Nous avions l'habitude de penser que nos bâtiments et nos ordinateurs, étaient intrinsèquement sûrs parce qu'ils sont grands, importants et visibles. Ce mythe a été démoli. Au contraire, ce type de bâtiments et d'ordinateurs sont les plus vulnérables. L'attaque dévastatrice sur des infrastructures américaines s'est produite à un moment où les entrepôts de données ont évolué aux Etats-Unis, jusqu'à acquérir un statut proche de celui de la production. L'entrepôt de données pilote maintenait la GRC (Gestion des Relations Clients) et assure le suivi, quasiment en temps réel, des commandes, des livraisons et des règlements. Voici une liste de menaces pouvant entrainer un arrêt catastrophique et de longue durée d'un entrepôt de données suivie d'une liste de solutions pratiques éventuelles.

Menace : Destruction du bâtiment, sabotage délibéré par une menace interne, guerre cybernétique, défaillances de maillons uniques (provoquées ou non)…

Solutions : Systèmes fortement distribués, voies de communication parallèles, les réseaux SAN (Storage-Area Networks), sauvegardes quotidiennes sur supports amovibles stockés en lieu sûr…

4 Etat de l'art des travaux de recherche

4.1 Présentation des travaux de recherche

Depuis quelques années, des travaux de recherche dans le domaine de la sécurité des entrepôts de données sont apparus. Nous allons les présenter selon d'une part les diverses exigences de sécurité (Confidentialité, Intégrité, Disponibilité, Authenticité) et d'autre part selon les composants ou les niveaux de l'entrepôt de données (Acquisition (ETL), Stockage, Accès (analyse)). La plus part des travaux existants s'intéressent et proposent des démarches pour quelques problématiques spécifiques, mais très peu de travaux proposent une solution globale de sécurité telle que celle suggérée par R. Kimball [RK97] ou S. Warigon [SW97].

Torsten Priebe et al, [TP00], dans le cadre du projet GOAL (Geographic Information Online Analysis) qui vise à étudier l'intégration de systèmes informatiques géographiques (GIS) et la technologie d'entrepôt de données, donnent un aperçu de la sécurité OLAP et la nécessité de disposer de mécanismes de contrôles d'accès adéquats pour assurer la confidentialité des données sensibles. Il introduit une méthodologie de la sécurité OLAP en fonction des différents niveaux d'accès de manière à réduire à la fois l'apprentissage pour les administrateurs et les coûts des niveaux logiciels. Les auteurs présentent aussi une étude comparative de quelques produits commerciaux OLAP qui propose des solutions pour faire face aux exigences de sécurité. Nous verrons plus en détail ces produits dans la prochaine section. La comparaison est basée à la fois d'une part sur les informations des produits (l'approche générale, les composantes d'architecture, les politiques, les caractéristiques et l'administration de la sécurité) et d'autre part selon le niveau de complexité sur des exigences fondamentales ou de base et des exigences avancées (Fig.2).

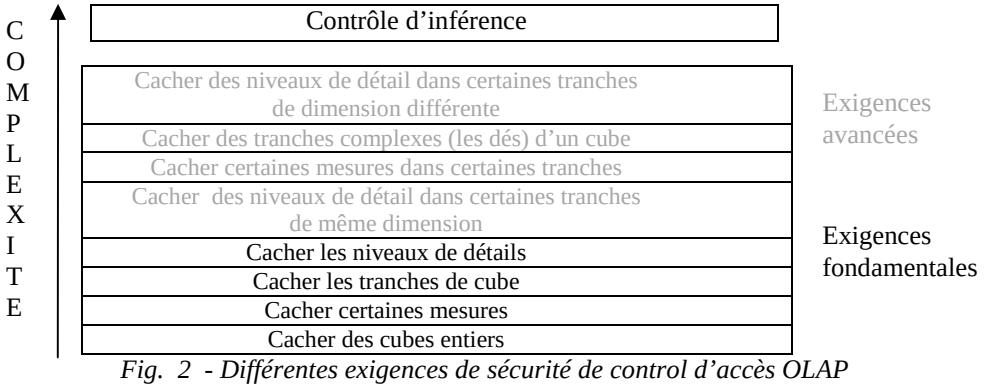


Fig. 2 - Différentes exigences de sécurité de control d'accès OLAP

Paulraj Ponniah [PP01] suppose que la sécurité des données dans un environnement d'entrepôt de données est similaire à la sécurité en OLTP Systems (OnLine Transaction Processing). Les autorisations et les restrictions d'accès aux données peuvent être accédées selon les rôles et les responsabilités. Après avoir créé les rôles, les utilisateurs seront rattachés à des rôles appropriés. (TAB.1).

ROLES	RESPONSABILITES	ACCES PRIVILEGES
UTILISATEUR FINAL	Exécuter des requêtes et des rapports contre l'entrepôt de données de tableaux	Système : non; Admin base de données : non; Tableaux et Vues : sélectionnez
ANALYSTE	Exécuter des requêtes ad hoc complexes, conception et de générer des rapports	Système : non; Admin base de données: non; Tableaux et Vues: tous
HELPDESK	Aide aux des utilisateurs avec des requêtes et des rapports, analyser et expliquer	Système : non; Admin base de donnée s: non; Tableaux et Vue s: tous
SPECIALISTE OUTILS REQUETTES	Installer utilisateur final et outils OLAP	Système : non; Admin base de données : non; Tableaux et Vues : tous
ADMINISTRATEUR SECURITE	surveiller l'utilisation des accès privilèges	Système : oui; Admin base de données : oui; Tableaux et Vues : tous
ADMINISTRATEUR RESEAU	Installer et maintenir l'exploitation systèmes et réseaux	Système : oui; Admin base de données : non; Tableaux et Vues : non
ADMINISTRATEUR ENTREPOT DE DONNEES	Installer et entretenir les bases de données et fournir un plan de sauvegarde et de récupération	Système : oui; Admin base de données : oui; Tableaux et Vues : tous

TAB. 1 - Echantillon des rôles, des responsabilités et des privilèges

Achref Karray et al, [AK06], présentent des travaux portant sur la sécurité et la confidentialité. Ils montrent les problématiques soulevées en ce qui concerne les données et les méthodes d'extraction de connaissance où les données et le code de fouille appartiennent à des propriétaires différents et qui ne se font pas nécessairement confiance. Ils décrivent l'application permettant le partage d'information entre les compagnies aériennes de l'Union Européenne et le CBP américain (bureau de la douane et de la protection des frontières), ils mettent aussi en évidence l'apport à cette problématique d'une architecture distribuée à base de Java Cards. S'appuyant sur cette expérience, les auteurs affirment qu'il est possible de

passer rapidement à l'échelle en termes d'une part de la quantité de données traitées et d'autre part de la complexité des traitements effectués. La plate-forme qui a permis de réaliser cette expérience est constituée de 2 machines interconnectées. Chaque machine comprend un PC de pilotage sous Linux, 16 lecteurs de cartes à puce connectés au PC pilote par des hubs USB. Les deux machines sont connectées entre elles par un réseau filaire classique et chaque machine dispose d'une connexion Wifi vers l'extérieur. Les perspectives associées à cette problématique sont nombreuses et concernent le développement de nouvelles applications, le déploiement automatique des codes, l'augmentation significative de la taille de la plate-forme pour accroître la capacité de calcul et la gestion de swap sécurisé off card pour accroître la taille des données manipulables.

Arnon Rosenthal et Edward Sciore [ARES00], présentent un travail intéressant sur le problème de sécurité des entrepôts de données. Ils proposent une nouvelle théorie qui permet l'inférence automatique de plusieurs droits pour l'entrepôt de données. Cette théorie est en fait une extension naturelle de modèle des accords SQL standards pour les systèmes avec les données redondantes et dérivées. Presque toute la capacité de la théorie vient de l'adaptation de mécanismes générales pour la vue sécurité, et d'exploitation des technologies présentes pour génération des requêtes équivalentes. Cette théorie aide à minimiser le temps d'administration du système et à implémenter les nouveaux logiciels.
Leurs contributions sont nombreuses : l'inférence automatique de droits d'accès sur les informations et de droit d'accès physique, la théorie de requêtes « témoins » équivalent, le droit « de vue ». Cependant, il reste encore plusieurs problèmes ouverts à résoudre à savoir : comment produire un régénérateur de requêtes, la capacité d'inférence des droits pour les opérations ETL, un algorithme efficace pour inférer des droits…

Pays Gupta et al, [PG08], comme beaucoup d'autres chercheurs de la communauté de la fouille de données qui s'intéressent à la détection de fraudes dans les réseaux, proposent une nouvelle approche de détection basée sur un paramétrage automatique du comportement des requêtes normales et de la distribution de valeurs d'attributs. Ils ont prouvé que les connaissances extraites représentées sous la forme de signatures statistiques peuvent être utilisées pour chercher efficacement des comportements malveillants dans le flot de requêtes. Etant donné que chaque serveur Web possède ses propres caractéristiques ou ses propres usages, les auteurs déterminent pour un serveur les caractéristiques qui correspondent à des requêtes valides, et modélisent les données valides sous la forme d'attributs qui pourront être utilisés pour générer des signatures. Afin de prendre en compte les nouveaux services et minimiser les fausses alarmes, les signatures sont maintenues d'une façon incrémentale. L'approche proposée peut être utilisée pour détecter les attaques de n'importe quel serveur mais le jeu d'apprentissage d'un serveur ne peut pas être utilisé pour tester les requêtes d'un autre serveur. Elle peut également mettre à jour les signatures pour prendre en compte les évolutions du site. C'est une technique qui a été développée pour détecter les nouvelles attaques plutôt que celles qui sont anciennes ou connues et qui peuvent être détectées par de nombreux SDI (Système Détection Intrusion). Les expérimentations ont montré que le système proposé est capable de reconnaitre 99.98% des requêtes valides et plus de 90% de requêtes invalides.

Ralph Kimball [RK97], propose de mettre en place un plan de sécurité. Selon Kimball, l'entrepôt de données doit avoir une équipe d'architecte de sécurité dont la tâche est de prévoir un plan de sécurité basé sur la sécurité des rôles, des technologies et de l'administration.

La deuxième étape de son plan pour obtenir la sécurité est de commencer à utiliser sérieusement la technologie de sécurité. Kimball affirme que la bonne technologie (IDS, routeur pare-feu, cryptage, serveur d'authentification…), permet d'assembler un système de sécurité pour un entrepôt de données qui offre l'équilibre entre les besoins des utilisateurs pour accéder aux données d'organisation et les besoins de garder les données confidentielles. L'idée architecturale, c'est que la sécurité de l'entrepôt est mise en dehors du SGBD (Système de Gestion de Base de Données) relationnel. Ce SGBD applique certainement les privilèges d'accès à la lecture / écriture, mais l'auteur ne donne pas de réponse à un bon nombre de questions importantes qui sont traitées en amont par le serveur d'authentification et le contrôle d'accès du serveur. Cinq ans plus tard, et pour répondre à ces questions, *Ralph kimball et Margy Ross [RKMR02]*, proposent une solution pour le contrôle d'accès par identification et reconnaissance centralisées des rôles. La solution appropriée est un serveur LDAP (Light-weight Directory Access Protocol) contrôlant tous les accès à l'entrepôt de données en dehors de la passerelle. Le serveur LDAP permet l'identification de manière uniforme de tous les utilisateurs demandeurs, qu'ils soient internes ou qu'ils arrivent par Internet depuis un lieu distant. Après l'identification, il associe l'utilisateur à un profil. Le serveur décide alors, si l'utilisateur, compte tenu de son rôle, peut voir les informations. Cette architecture présente des avantages significatifs dans l'hypothèse où les entrepôts de données se développent pour servir des milliers d'utilisateurs dans une centaines de rôles différents. *Ralph Kimball et al, [RK07]*, arrivent enfin à mettre en place l'architecture d'un serveur type d'un entrepôt de données dont les caractéristiques principales sont : l'interdiction des connexions directes des utilisateurs internes, ces derniers doivent passer par une interface d'accès par jeton, l'utilisation des communications cryptées via le protocole VPN (Virtual Protocol Network), d'un serveur d'annuaire LDAP, d'un serveur d'application OLAP (OnLine Analytical Processing) ou ROLAP (Relational OLAP).

Slemo Warigon [SW97], propose de mettre en place, un plan complet en sept étapes pour la sécurité des entrepôts de données :
1 : identifier les données stockées dans l'entrepôt à savoir toutes les tables, les colonnes, les lignes, les types de données, qui utilisent les données et à quelle fréquence.
2 : classer les données en fonction de leur sensibilité à la divulgation, la modification et la destruction (public 'moins sensibles', confidentielle 'modérément sensibles' et top secret 'sensibles' ou 'extrêmement sensibles'). L'objectif essentiel de cette classification des données par rapport au niveau de sensibilité est de prévoir les différentes mesures de protection à mettre en œuvre pour chaque catégorie. Le classement des données en fonction de différentes catégories n'est pas aussi simple. Certaines données présentent un mélange de deux ou de plusieurs catégories selon le contexte d'utilisation.
3 : quantifier leurs valeurs. La valeur des données est mesurable par le coût de sa reconstruction en cas de perte, de la restauration de l'intégrité des données corrompues, de son déni de service, de l'indemnité financière, des pertes de revenus causées par la fuite de secrets …
4 : identifier les vulnérabilités liées à l'environnement de l'entrepôt de données telles que facteurs humains, facteurs naturels, facteurs utilitaires…
5 : élaborer des mesures de protection (contrôles d'accès, contrôles d'intégrité, cryptage des données….)
6 : estimer les coûts des mesures de protection, en effet toutes mesures de sécurité impliquent des dépenses, c'est l'objectif de la sixième étape du plan. Les dépenses doivent être justifiées. Dans tous les cas, les coûts de la protection des données à risque ne doivent pas dépasser la valeur des données, d'où la nécessité d'une sélection coût-efficacité.

7 : évaluer l'efficacité des mesures de sécurité. Cette évaluation doit être régulière pour déterminer si les mesures sont toujours pertinentes.
Le plan proposé par S. Warigon permet à la fois la prévention des problèmes de sécurités des entrepôts de données et aussi de prendre des mesures correctives dans une période de crise, c'est une stratégie gagnante pour assurer un heureux mariage entre l'idéalisme de l'entrepôt de données basé sur le pouvoir d'information de traitement et la philosophie pragmatisme d'une sécurité proactive basée sur la sécurité des pratiques prudentes dans l'environnement informatique.

4.2 Etude comparative des travaux de recherche

L'étude comparative des travaux de recherche est présentée d'une part selon les quatre exigences de sécurité identifiées pour des entrepôts de données :
Confidentialité, Intégrité, Disponibilité et Authenticité et d'autre part selon les phases de processus d'entreposage à savoir : Acquisition (ETL), Stockage, et Analyse (TAB.3).

Phases d'entreposage / ***Auteurs***	***ETL (Acquisition)***	***Stockage***	***Analyse***
Kimball R. (1997)			**Confidentialité**
Warigon S. (1997)		**Confidentialité** **Intégrité** **Disponibilité** **Authenticité**	**Confidentialité** **Intégrité** **Disponibilité** **Authenticité**
Priebe T. et al (2000)	**Confidentialité**		**Confidentialité** **Disponibilité** **Authenticité**
Rosenthal A. et al (2000)	**Disponibilité**		**Confidentialité** **Authenticité**
Ponniah P. (2001)	**Disponibilité**	**Disponibilité**	**Intégrité** **Authenticité**
Kimball R. et al (2002)			**Intégrité** **Authenticité**
Laurent A. et al (2004)	**Intégrité**	**Disponibilité**	**Intégrité**
Karray A. et al (2006)			**Confidentialité** **Intégrité** **Disponibilité**
Eden C. et al (2006)			**Confidentialité** **Intégrité** **Disponibilité** **Authenticité**
Cuppens F. et al (2007)	**Intégrité** **Disponibilité** **Authenticité**	**Confidentialité**	**Confidentialité** **Intégrité** **Disponibilité** **Authenticité**
Kimball R. et al (2007)			**Intégrité** **Disponibilité** **Authenticité**
Gupta P. et al, (2008)			**Confidentialité** **Intégrité**

TAB. 3 - Travaux de recherches portant sur les exigences de sécurité des différents niveaux processus d'entreposage

4.3 Bilan concernant les travaux de recherche

D'après notre étude bibliographique portant sur le thème de la sécurité des entrepôts de données et à partir de notre étude comparative que nous venons de présenter, nous avons constaté que : très peu de travaux considèrent que la sécurité physique (infrastructure, réseaux…) est suffisante pour garantir la sécurité de l'entrepôt de données.
Les travaux existants se situent par moment sur un ou plusieurs processus d'entreposage et concernent un ou plusieurs objectifs de sécurité.
Beaucoup de travaux portent sur l'aspect analyse notamment la gestion de la confidentialité mais très peu s'intéressent aux niveaux acquisition (ETL) et stockage.
Aucune proposition concernant un plan de sécurité global couvrant toutes les phases du processus d'entreposage [NH08].

5 Outils commerciaux

5.1 Présentation de l'offre sécurité de quelques produits

En matière d'entrepôt de données, il existe toute une série de produits. Parmi les plus utilisés, on peut citer *Teradata, SAS ou Cognos* qui offrent tous des possibilités différentes. *Teradata* est un entrepôt de données pur et sur ce point, il est un parmi quelques produits qui dominent le marché. *SAS* offre des outils d'analyse très puissants et c'est dans ce domaine qu'il est *leader. Cognos* offre des outils d'agrégation et de résumé très avancés.
La plus part des outils se concentrent essentiellement sur les aspects de l'interrogation.
Plusieurs requêteurs (*Business Objects, Impromptu, Discover2000)* et outils *OLAP (Powerplay, Discover, Essbase, Express, Mercury)* sont proposés. Les requêteurs facilitent la construction de requêtes SQL (jointures masquées, alias de noms d'attributs,…). Les outils OLAP se concentrent sur l'analyse multidimensionnelle en proposant des objets graphiques (tableaux croisés,…). Ces outils utilisent les données d'entrepôts ou des magasins, mais ne permettent pas leur construction. Peu d'outils sont disponibles pour assister l'administrateur dans la construction des entrepôts et des magasins de données *(Data Mart Builder, SAS/Warehouse Administrator, Warehouse Manager).* Avec ces outils, le schéma de l'entrepôt doit être construit au préalable. L'administrateur a la charge d'associer chaque attribut cible de ce schéma à un attribut d'une source de données. Cette phase reste complexe et demande une connaissance importante des structures sources.
On distingue généralement deux catégories de produits commerciaux : les produits qui proposent une solution complète d'entrepôsage y compris la sécurité bien évidement, d'autres produits plus nombreux sont spécifiques à un aspect de processus d'entreposage spécifique tel que les outils sécurité OLAP.
La ligne de produits *Statistica* offre la gamme la plus riche du marché en termes de procédures d'analyse, de gestion et de représentation des données, ainsi que de solutions de data mining. *Statistica Data Warehouse* intègre un système de sécurité très performant et sophistiqué, garantissant la confidentialité des connaissances et de l'intelligence par rapport à toute tentative d'intrusion non autorisée. Le système *Statistica Data Warehouse* va sans doute s'imposer comme le système incontournable d'information décisionnelle au sein de toute organisation.

C'est la raison pour laquelle le système de sécurité a une importance particulière, afin de prévenir toute intrusion non autorisée au niveau des ressources les plus sensibles. *Statistica Data Warehouse* intègre le niveau de sécurité le plus élevé en permettant de définir des groupes d'utilisateurs associés à des droits d'accès différents (qui vont à leur tour spécifier l'information qui est accessible, et les opérations qui sont réalisables), les mots de passe doivent être mis à jour périodiquement pour accroître encore la sécurité. Diverses méthodes spécialisées sont également proposées pour assurer la détection et la protection contre toute tentative d'intrusion électronique systématique (les "hackers"). *Statistica Gestion Documentaire* permet de respecter les politiques de gestion documentaire ou réglementaires en matière de sécurité des documents, traçabilité et signatures/authentifications.

Quelques produits commerciaux OLAP proposent des solutions pour assurer la sécurité des entrepôts de données [TP00]. Ces outils se diffèrent (TAB.4) d'une part par les informations sur les produits (l'approche générale, les composantes d'architecture, les politiques, les caractéristiques et l'administration de la sécurité) et d'autre part par le niveau de complexité des exigences fondamentales ou de base (Cacher les caractéristiques cube, cacher certaines mesures, cacher des tranches du cube ou cacher des niveaux de détails) et des exigences avancées de sécurité (Cacher les niveaux de détails de certaines tranches de même dimension, cacher certaines mesures dans certaines tranches, cacher les tranches complexes du cube ou cacher les niveaux détails de certaines tranches de dimensions différentes).

Dans les projets avec ROLAP des instruments relationnels sont utilisés s'il n'y a pas de contrôle d'accès suffisant pour le produit au niveau de serveur OLAP.
SQL serveur 2000 améliore le modèle de sécurité de services pour inclure en plus de la sécurité de niveau de cellule, une caractéristique de sécurité de dimension qui permet aux métas donnés et au membre de dimension pénétrant de fournir la transparence d'utilisateur final. Avec les mesures de sécurité de dimension, les niveaux de hiérarchie et les membres de dimension (c'est-à-dire les tranches) peuvent être cachés.
MicroStrategy7 fournit deux moyens de contrôle d'accès. D'abord, une liste de contrôle d'accès est maintenue pour tous les objets de méta données, en incluant des attributs (c'est-à-dire les niveaux de hiérarchie de dimension) et la métrique (les mesures), mais pénètre aussi, les gabarits et les rapports prédéterminés. Le propriétaire ou l'administrateur décident qui peut accéder à l'objet. La deuxième façon est par soi-disant filtre de sécurité qui est une construction qui représente fondamentalement la coupe d'une requête d'OLAP. Ceux-ci empêchent des utilisateurs de voir certaines données dans la base de données.
Dans *Cognios PowerPlay*, la sécurité est respectée par l'instrument front fin en utilisant les éléments de sécurité (encrypté) d'autorisation des dossiers.
Le contrôle d'accès est défini sur la catégorie (la terminologie de *Cognios* pour le membre de dimension), la mesure, ou le niveau de dimension, en utilisant une approche de vue multidimensionnelle.
Oracle Express, très flexible pour la spécification des contraintes même les plus complexes. D'autre part, comme les programmes de permission fournissent seulement (le fait) la filtration de données, en quittant les membres de dimension et les métas données inchangées, il n'est pas possible de cacher l'existence de données sensibles (la transparence pour l'utilisateur final).

	Produits	Produits ROLAP	Microsoft SQL Server 2000	MicroStrategy 7	Cognos PowerPlay	Oracle Express
Info	**Libération évaluée**	N/A	8.0 BETA	7.0 BETA	6.0	6.2
	Caractéristiques de sécurité soutenues	SQL views	Cell-level et dimension security	Access control list and security filtres	User class et dimension views	Permission programs
	Sécurité faisant respecter la composante d'architecture	DBMS	OLAP server ou OLAP front-end	OLAP server ou OLAP front- end	OLAP front-end	OLAP server
	Approche générale	View	Hybrid[3]	View	View	Rule
	Politiques de sécurité	Closed World	Open World	Open World	Open World	Open World
	Administration sécurité	Ownership	Administrator	Ownership	Administrator	Administrator
Exigences	**Cacher les caractéristiques cube**	•	•	•	•	•
	Cacher certaines mesures	•	•	•	•	•
	Cacher des tranches du cube	•	•	•	•	•
	Cacher des niveaux de détails	•	•	•	•	•
	Cacher les niveaux de détails de certaines tranches de même dimension	•	•	◦	•	•
	Cacher certaines mesures dans certaines tranches	•	•	◦	-	•
	Cacher les tranches complexes du cube	•	•	•	-	•
	Cacher les niveaux détails de certaines tranches de dimensions différentes	◦	•	◦	-	•
	Dynamique / donnée contrainte conduit	◦	◦	◦	-	◦
	Inférence control	-	-	-	-	-

TAB. 4 - Produits commerciaux sécurité OLAP

5.2 Discussion et critiques

Nous constatons qu'une classification des produits commerciaux peut se faire, en effet, on distingue les produits qui proposent une solution complète d'entreposage de données, d'autres produits plus nombreux sont spécifiques à un aspect de processus d'entreposage tel que les outils de sécurité OLAP. On peut également classer ces produits selon leur type, à savoir, les produits propriétaires, les suites décisionnelles libres tel que BIRT, *Pentaho ou Insight Strategy BEE 2.1…* et les produits ETL tels que *Talend Open Studio1, Berlios Clover.ETL 2.0….* Les outils des éditeurs propriétaires : *Business Objects, Cognos, Hyperion, SAS, Teradata…* coûtent cher, car ils sont souvent facturés au nombre d'utilisateurs ou de connexions simultanées. Les entreprises se tournent donc vers des outils open source pour réduire les coûts de licence de 20 à 50 % du coût total d'un projet décisionnel. Il faut signaler également que la personnalisation d'une solution open source demande un certain engage-

ment que les utilisateurs ne sont plus forcément prêts à faire vu qu'ils ont pris l'habitude avec des solutions déjà packagées.
De point de vu sécurité, le risque technique est trop important puisque les outils open sources intègrent moins d'assistants, moins de stabilité et de performances que les outils propriétaires. D'après l'étude comparative des différents produits commerciaux, il est clair qu'il y a quelques systèmes qui fournissent quelques mécanismes pour s'occuper des exigences de sécurité notamment la gestion des droits d'accès, mais Il n'y a aucune proposition standard pour le contrôle d'accès dans le monde multidimensionnel.

6 Conclusion et perspectives

Il est claire que d'une part, le décisionnel est un secteur en plein développement au vu de la croissance exponentielle des données manipulées par les entreprises. Et que d'autre part, nous constatons, à travers les travaux de recherche étudiés et les offres sécurité de quelques outils existants, que les différents acteurs commencent à prendre conscience de la nécessité de poursuivre le développement de ce secteur d'une façon générale et de la sécurisation des entrepôts de données d'une façon particulière.
Une proposition qui nous semble intéressante est celle de combiner les résultats des travaux de Ralph Kimball [RK07] pour sécuriser l'environnement de l'entrepôt de données avec le plan de Slemon Warigon [SW97] qui vise à sécuriser les accès et contrôler l'intégrité des données pour garantir une meilleure sécurité.
Concernant nos perspectives, les discussions que nous avons menées sur notre travail ont fait émerger de nombreuses idées. Nous comptons alors :
- établir une étude complète sur les différentes vulnérabilités des entrepôts de données
- s'inspirer du domaine de la sécurité des systèmes d'informations de production et de l'énorme travail déjà réalisé dans ce domaine et l'adapter aux spécificités des entrepôts de données notamment dans la gestion des droits d'accès.
- développer un outil de gestion automatique des droits d'accès à partir des profils des utilisateurs et des spécificités des entrepôts de données.
- réfléchir sur le développement de la sécurité des composants acquisition (ETL) et stockage de l'entreposage vu que la majorité des travaux traitent le composant analyse.
- proposer un plan de sécurité global pour l'entreposage des données.

Références

- [AK06] Achraf Karray, Serge Chaumette et Damien Sauveron. Gestion de la Sécurité pour l'Extraction Parallèle Distribuée Des Connaissances. 2006

- [ARES00] Arnon Rosenthal et Edward Sciore. View Security as the Basis for Data Warehouse Security. Proceedings of the Intemational Workshop on Design and Management of Data Warehouse (DMDW'2000), Manfred A. Jeusfeld, Hua Shu, Karlstad Martin Staudt, Gottfried Vossen (Editor), Sweden, June 2000.

- [BI96] Bill Inmon W.H. Building the Data Warehouse. John Wiley & Sons, INC. Second Edition. ISBN n° 0471-14161-5. 1996.

- [NH08] Nouria Harbi. Cours : sécurité des entrepôts de données, Master 2 ECD, Université Lumière Lyon2, Janvier 2008

- [PG08] Pays Gupta, Chedy Raïssi, Gérard Dray, Pascal Poncelet et Johan Brissaud. Détection d'intrusions : de l'utilisation de signatures statistiques. 2008.

- [PP01] Paulraj Ponniah. Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals. Copyright © 2001 John Wiley & Sons, Inc. ISBNs: 0-471-41254-6 (Hardback); 0-471-22162-7 (Electronic). 2001.

- [RK07] Ralph Kimball, Laura Reeves, Margy Ross et Thornthwaite Warren, Le data Warehouse Guide de conduite de projet. 3ème tirage 2007édition EYROLLES. 2007.

- [RK97] Ralph Kimball.: Hackers, Crackers, and Spooks; ensuring that your data warehouse is secure. In DBMS Magazine; April 1997.

- [RKMR02] Kimball Ralph et Ross Margy, entrepôts de données Guide pratique de modélisation dimensionnelle. Traduction de Claude Raimond. 2ème édition 2002. . l'édition originale de ce livre a été publié aux États-Unis par John Wiley & Sons, Inc., 605 Third Avenue, New York, 10159, sous le titre : The data warehouse Toolkit-Second Edition. 2002.

- [SW97] Slemo Warigon, Data Warehouse Control and Security CISA, MBA Association of College and University Auditors LEDGER, Vol. 41, No. 2, April 1997; pp. 3-7. Copyright © 1997 The Ledger -- All rights reserved. 1997.

- [TP00] Torsten Priebe, Günther Pernul. Towards OLAP Security Design - Survey and Research Issues. Proc. of the 3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP 2000), Washington, DC, November 10, 2000.

Intégration automatique des données semi-structurées dans un entrepôt cellulaire

Fawzia Zohra Abdelouhab, Baghdad Atmani

Département Informatique, Faculté des Sciences, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie
fzabdelouhab@yahoo.fr, atmani.baghdad@univ-oran.dz

Résumé. Pour enclencher un processus d'extraction des connaissances à partir de données complexes, il faut intégrer puis représenter les données complexes sous une forme adaptée aux techniques d'analyse en ligne ou de fouille de données. La problématique d'intégration, de modélisation, de structuration et d'extraction de connaissances à partir de données complexes nécessite une méthodologie et des outils génériques adaptés. D'un autre côté, l'automate cellulaire est un système dynamique discret capable de simuler un comportement global complexe en utilisant des règles de transition simples et locales. En effet, malgré la simplicité des règles qui le définisse, il fait apparaître de nombreux phénomènes imprévisibles, qui sont *à priori* difficilement obtenues par des méthodes analytiques classiques. L'idée d'utiliser le formalisme des automates cellulaires pour résoudre le problème d'intégration dans les entrepôts de données, représente le point de départ de notre réflexion sur un nouveau principe cellulaire capable d'alimenter automatiquement un entrepôt de données semi-structurées.

Partant de ce constat, nous proposons dans cet article notre problématique sur la conception d'un outil cellulaire d'intégration de données semi-structurées dans un entrepôt de données.

Mots clés: Entrepôts de données, Intégration des données hétérogènes, Automate cellulaire, Similarité et Appariement des graphes, Algorithmes de Matching et de Mappings.

1 Introduction

Avec l'apparition de l'internet, nous assistons, aujourd'hui, à une explosion d'outils et de systèmes informationnels engendrant des données, diverses, complexes et fortement distribuées. Ceci complique considérablement l'intégration de ces données dites hétérogènes au sein d'une même structure de données qui est l'entrepôt.

La notion d'hétérogénéité est inhérente à chaque approche même si elle présente, à priori, plusieurs analogies. Certains travaux tels que Beneventano et al. (2002), Kim et Park (2003) et Maibaum et al. (2005) qualifient les données de différentes catégories comme étant hétérogènes. Alors que dans d'autres Saccol et Heuser (2002), l'hétérogénéité traite des données de même catégorie mais avec des modélisations différentes. On trouve même des travaux, désignant des données de même catégorie avec la même modélisation qui parlent de données hétérogènes Da Silva et al. (2002).

Si l'on considère les différentes représentations et formats des documents à traiter, les données peuvent être classées selon Hamdoun et al. (2007) en plusieurs catégories : structurées (données relationnelles, données objets), semi-structurées (HTML, XML, graphes) ou non structurées (textes, images, sons). L'hétérogénéité des sources d'un entrepôt de données est illustrée par la figure 1.

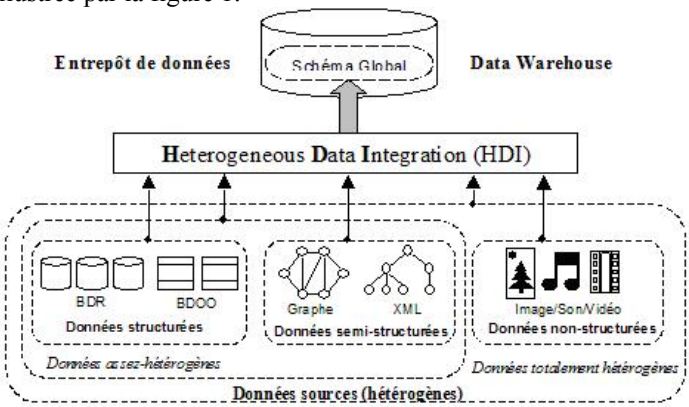


FIG. 1 – *L'hétérogénéité des sources d'un entrepôt de données.*

Cette ambiguïté d'hétérogénéité des données est due au fait qu'aucun consensus n'ai été établi afin d'élaborer un modèle unique de représentation de données à intégrer. Le traitement de données complètement hétérogènes structurées, semi-structurées et non structurées s'avère donc un volet de recherche récent et assez peu exploré. Dans un tel contexte, le besoin d'intégration devient un concept incontournable mais en même temps compliqué car il se voit contraint de composer avec la répartition des sources, l'hétérogénéité de leurs structures et la complexité de leurs données (voir Boussaid et al., 2006).

D'autres parts, un entrepôt de données, qu'il soit homogène ou hétérogène, nécessite d'être maintenu. Il doit aussi évoluer en fonction de l'évolution des données sources aussi bien au niveau des structures qu'au niveau des données**.** Le problème de la maintenance est également très complexe. Les algorithmes proposés dans la littérature traitent essentiellement de données sources homogènes (voir Laurent et al., 2001). Par contre, concernant les données sources hétérogènes la question reste à étudier...

Notre contribution consiste à exploiter les performances confirmées et les techniques mathématiques formelles de la machine cellulaire CASI (Atmani et Beldjilali, 2007) afin de concevoir, dans un premier temps, un processus dynamique de construction d'un entrepôt de données dit cellulaire par une intégration automatique des données semi-structurées des sources hétérogènes. Et dans un deuxième temps, le soumettre aux techniques des fouilles de données pour d'éventuelles extractions des connaissances. Dans cet article seul l'intégration automatique des données semi-structurées sera présentée.

Cet article est structuré comme suit. Le paragraphe 2 traite de la construction des entrepôts de données semi-structurées. Le principe de la machine cellulaire est présenté dans le paragraphe 3. La problématique qui est le système cellulaire d'intégration des données hétérogènes semi-structurées est abordée dans la section 4. Nos travaux futurs seront donnés en guise de conclusion.

2 Construction des Entrepôts de données

2.1 Travaux similaires

Le rôle des systèmes d'intégration de données est de répondre aux besoins des utilisateurs à travers des interfaces d'accès uniformes à ces sources de données, Zerdazi (2007). Le défi de l'intégration de sources de données est de faire cohabiter ces sources hétérogènes, de plus en plus nombreuses, souvent réparties et indépendantes, dans un seul système uniforme, appelé système d'intégration, sans contraindre le comportement ni l'autonomie de chacune d'elles.

Dans les premières solutions proposées, les systèmes d'intégration se répartissent en deux composants fonctionnels à savoir les adaptateurs (wrappers en anglais) et les médiateurs, Zerdazi (2007). Les adaptateurs sont des composants dédiés aux sources qui permettent d'abstraire et de cacher l'hétérogénéité aux médiateurs. Quant aux médiateurs, ils permettent de concilier les données provenant des différents adaptateurs à travers un langage de requêtes et un modèle de données commun. Le projet TSIMMIS, Garcia-Mollina et al. (1995), le projet DISCO, Tomasic et al. (1997) et le projet YAT, Siméon (2000) en sont de réels exemples. Le temps a prouvé que ces travaux n'ont pas pu apporter de solutions génériques au problème de l'intégration mais ils ont mis en avant une approche, un modèle et un algorithme pour la résolution d'un problème particulier de l'intégration et souvent dans un domaine d'application spécifique. Une troisième approche a vu le jour, c'est l'approche matérialisée (ou par entrepôt) Widom (1995), approche asynchrone, dans laquelle l'utilisateur interroge un référentiel contenant une copie des données issues de différentes sources de données. Le projet XYLEM, Sorlin et Solnon (2004) est un système d'entrepôt dynamique ayant pour but de stocker et d'intégrer de manière semi automatique toutes les ressources XML du Web. Ce stockage permet à l'utilisateur final d'avoir un accès unique et transparent à toutes les données hétérogènes. L'utilisation d'un système à base d'arbre, Delobel et al. (2003), contribue à faire de XYLEM un système efficace pour l'évaluation des requêtes, l'intégration des données et leur maintenance.

L'adaptation d'un schéma source par rapport à un autre schéma cible passe par deux étapes importantes :

1. **Le Matching :** est un processus qui effectue des correspondances sémantiques entre les éléments et les attributs des schémas, et retourne comme résultat les valeurs de similarités sémantiques entre les deux schémas. Nous parlons d'un appariement structurel.
2. **Le Mapping :** se sont des expressions décrivant le moyen dont les instances du schéma cible (final) sont dérivées à partir des instances de schéma source (initial). elles décrivent la correspondance sémantique entre les instances de schémas en complémentarité avec le Matching. Et là nous parlons d'un appariement ontologique qui dépend du domaine d'application à traiter.

Nous nous sommes, donc, intéressés aux processus de Matching et de Mapping des schémas XML. Ces deux processus qui se suivent sont des pré-requis à l'intégration et à la transformation des schémas XML. Avec ces deux descriptions nous pouvons restructurer les données d'une représentation à une autre. C'est-à-dire passer d'un schéma source à un schéma cible et vis-et-versa.

2.2 Pourquoi le schéma XML?

Tout d'abord le standard XML n'est pas un langage à proprement dit. Il représente une *famille de langages* ayant en commun le respect de certaines règles qui le rendent, à la fois, extensible, rigoureux, ouvert et universel. Un schéma XML représente la structure arborescente d'un document XML. Cette arborescence comporte une racine unique, des branches et des feuilles. L'élément-racine indique la base du document `XML`. Il est unique et englobe *tous* les autres éléments. Les éléments forment la structure même du document : ce sont les branches et les feuilles de l'arborescence. Ils peuvent contenir du texte, ou bien d'autres éléments, qui sont alors appelés "éléments enfants", l'élément contenant étant quant à lui appelé logiquement "élément parent". Tous les éléments peuvent contenir un ou plusieurs attributs. Chaque élément ne peut contenir qu'une fois le même attribut. Un attribut est composé d'un nom et d'une valeur.

Le choix du schéma XML porte essentiellement sur sa puissance d'expression et surtout sur sa normalisation définie par W3C comme étant une infrastructure de base pour la description du type et de la structure des documents XML. Cette description suit un format commun que tous les navigateurs Web, les applications et les clients utilisant XML peuvent reconnaître. En d'autres termes les schémas XML définissent les règles qu'un document de données XML doit respecter. De ce fait, les méthodes de Matching, basées sur les schémas c'est-à-dire sur les noms, les types et les méta-données, permettent d'obtenir un ensemble de règles d'associations entre les éléments des schémas XML. Un autre avantage d'utiliser le schéma XML est qu'il permet la définition de modèles de données structurés, typés et ayant de puissantes propriétés de validation à travers les fonctionnalités additives suivantes :

- Le typage des données qui permet la gestion de booléens, d'entiers, d'intervalles de temps... Il est même possible de créer de nouveaux types à partir de types existants.
- La notion d'héritage. Les éléments peuvent hériter du contenu et des attributs d'un autre élément. C'est sans aucun doute l'innovation la plus intéressante de XML Schéma.
- Le support des espaces de nom.
- Les indicateurs d'occurrences des éléments peuvent être tout nombre non négatif.
- Les schémas sont très facilement concevables par modules.

2.3 Les étapes conceptuelles de notre projet

Rappelons que l'objectif de notre travail est de traiter l'hétérogénéité des données par la construction d'un entrepôt de données cellulaire, en incorporant, à partir des couches les plus basses du système, le principe de la représentation binaire adopté par la machine cellulaire CASI dont le schéma conceptuel est donné par la figure 2. La conception de notre outil cellulaire se traduit par ce qui suit :

1. Il est nécessaire dans un premier temps d'utiliser un modèle commun pour assurer une bonne compréhension et une bonne intégration des données échangées. Ce modèle devra être indépendant des sources et permettre de construire une vue commune sur ces différentes sources. Le but d'une telle représentation logique est de capturer les caractéristiques structurelles et sémantiques des schémas de départ pour faciliter leur compréhension tout en

offrant plus de souplesse et plus de flexibilité lors de leur comparaison et de leur intégration. Le choix s'est porté sur les schémas XML.

2. Trouver les correspondances entre les différents schémas de données, dans un mode automatique (le Matching ou appariement). C'est-à-dire rechercher des mises en correspondances entre concepts puis entre propriétés et enfin entre relations afin d'établir la correspondance des éléments des schémas et leur transformation d'un schéma à un autre. Cette correspondance permet de spécifier l'ensemble des assomptions heuristiques, ainsi que l'ensemble des informations d'entrée nécessaires permettant de positionner un cadre formel pour l'appariement inter-schémas. Selon Limam (2007) c'est une tâche extrêmement complexe à réaliser car elle relève du domaine cognitif. Le problème peut s'exprimé comme suit « comment réconcilier deux sources de données hétérogènes».
3. Rendre ces correspondances opérationnelles (Création de vues) : en utilisant la machine CASI pour la découverte automatique de mapping. Cette découverte se fait par comparaison des graphes au moyen des règles de production.

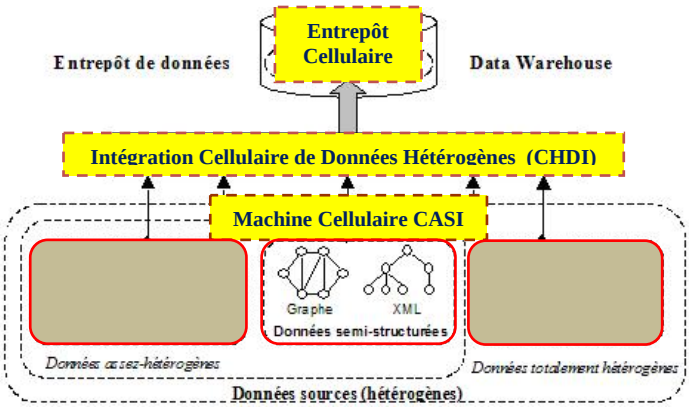


FIG. 2 – *Intégration cellulaire des données semi-structurées*.

Pour cela, nous distinguons deux étapes indépendantes ; la première qui consiste à décrire l'aspect structurel des schémas sources se résumant à la définition de trois types d'éléments de bases : concept, relation, et propriété. La deuxième étape qui tient compte de l'aspect sémantique des schémas consistant en l'enrichissement sémantique de ces éléments via des métaconnaissances sémantiques (entre autres, spécification des concepts pertinents, découverte des catégories de relations, identifications des propriétés clés, etc.).

Pour cela, et c'est là l'originalité du projet, notre solution au problème consiste à proposer une formalisation graphique des schémas unifiés et une pondération binaire du degré de similarité à l'aide de la machine cellulaire CASI.

3 La machine cellulaire CASI

CASI (Automate Cellulaire pour les Système d'Inférence), issue des travaux de Atmani et Beldjilali (2007) est un automate cellulaire qui simule le principe de fonctionnement de base d'un Moteur d'Inférences en utilisant deux couches finies d'automates finis. La première couche, CELFACT, pour la base des faits et, la deuxième couche, CELRULE, pour la base de règles. Chaque cellule au temps t+1 ne dépend que de l'état des ses voisines et du sien au temps t. Dans chaque couche, le contenu d'une cellule détermine si et comment elle participe à chaque étape d'inférence : à chaque étape, une cellule peut être active (1) ou passive (0), c'est-à-dire participe ou non à l'inférence. Le principe adopté est simple :

– Toute cellule i de la première couche CELFACT est considérée comme fait établi si sa valeur est 1, sinon, elle est considérée comme fait à établir. Elle se présente sous trois états : état d'entrée (EF), état interne (IF) et état de sortie (SF)

– Toute cellule j de la deuxième couche CELRULE est considérée comme une règle candidate si sa valeur est 1, sinon, elle est considérée comme une règle qui ne doit pas participer à l'inférence. Elle se présente sous trois états : état d'entrée (ER), état interne (IR) et état de sortie (SR). Les matrices d'incidence RE et RS représentent la relation entrée/sortie des Faits et sont utilisées en chaînage avant et en chainage arrière en inversant leur ordre.

La dynamique de la machine CASI, pour simuler le fonctionnement d'un Moteur d'Inférences, selon Atmani et Beldjilali (2007) utilise deux fonctions de transitions δ_{fact} et δ_{rule}, où δ_{fact} correspond à la phase d'évaluation, de sélection et de filtrage, et δ_{rule} correspond à la phase d'exécution.

- La fonction de transition δ_{fact} :
δ_{fact} (EF, IF, SF, ER, IR, SR) = (EF, IF, EF, ER+(R_E^T.EF), IR, SR)
- La fonction de transition δ_{rule} :
δ_{rule} (EF, IF, SF, ER, IR, SR) = (EF+(RS.ER), IF, SF, ER, IR, ^ER), où la matrice R_E^T désigne la transposée de R_E et ^ER désigne la négation du vecteur booléen ER.

Dans la section qui suit nous verrons comment utiliser les principes de la machine CASI à travers un exemple.

4 Intégration cellulaire d'un document XML

Le modèle de données de l'entrepôt est basé sur le concept Objet. En se basant sur les schémas XML et en s'inspirant des principes de fonctionnement de la machine CASI, nous engendrons des règles de construction de notre entrepôt de données dit cellulaire. Etant donné qu'un entrepôt est conçu spécialement pour un domaine d'application particulier, il peut être vu comme un contenant d'une ontologie c'est-à-dire d'une description explicite et déclarative de la sémantique du domaine d'application. Cette description est représentée par un schéma global constitué d'un ensemble de classes, un ensemble fini de super classes, un ensemble d'instances et une fonction de construction (le Mapping). Chaque classe extraite du

Web est représentée dans l'entrepôt comme un objet entrepôt de base ayant un unique identifiant, un nom, un type définissant la structure et le comportement des instances de cette classe.

La fonction de construction est un ensemble de règles de transformations permettant de spécifier les dérivations mises en jeu pour intégrer la classe dans le schéma global :

- Dériver les structures de la source pour définir celle des classes entrepôt.
- Augmenter les classes entrepôt en recopiant uniquement l'information utile.
- Transformer certaines structures de données extraites pour redéfinir des structures pertinentes pour l'entrepôt.
- Organiser la hiérarchie d'héritage des classes entrepôt afin de structurer efficacement le schéma des classes de l'entrepôt.

Cette fonction est une composition de fonctions de base qui peuvent être classées comme suit :

- les fonctions de structuration pour intégrer la structure des classes entrepôt (attributs et relation). Elles permettent de créer un schéma vide contenant la classe générique de l'entrepôt avec deux éléments vides ; les éléments simples et les éléments complexes.
- les fonctions d'adjonction pour intégrer les objets sources à partir desquels l'extension des classes entrepôt est calculée. Elles permettent de créer de nouvelles propriétés qui sont soit des attributs calculés soit des propriétés spécifiques.
- Les fonctions d'héritages pour organiser la hiérarchie d'héritage dans l'entrepôt en créant des super classes et des sous classes.

Exemple d'application : considérons une ontologie 'Bibliographie' à partir de laquelle nous souhaiterons créer un entrepôt. Le fichier XML BIBLIO représentant cette bibliographie est illustrée par la figure 3. A partir du fichier de la figure 3 on extrait la grammaire XML correspondante donnée par la figure 4. Un arbre T représentant le schéma XML est un ensemble tel que T={N, A, AT, ATN} où :

- N est l'ensemble fini des nœuds appartenant au schéma associé ;
- A est l'ensemble fini des couples (N,N) représentant des arêtes entre deux nœuds non vides. Ils correspondent aux liens de compositions entre éléments du schéma ;
- AT est l'ensemble fini des attributs du schéma où chaque attribut est lié à un élément de l'ensemble N ;
- ATN est l'ensemble fini des couples (ei,<at1,at2,….atn>) tel que i $\in${1, n}. si un élément ex possède deux attributs aty et atz alors le couple correspondant dans ATN sera (ex,<aty,atz>).

En représentant la structure de l'arbre T sur notre schéma XML BIBLIO nous aurons les ensembles définis comme suit :

- N={BIBLIO, ISBN, TRADUCTEUR, AUTEUR, EDITEUR }
- A={(BIBLIO, ISBN), (BIBLIO, TRADUCTEUR), (ISBN, AUTEUR), (ISBN, EDITEUR)}
- AT={TITRE, DATEPUB, NOM, PRENOM, PREFIX, PLACE}
- ATN={(ISBN, <TITRE, DATEPUB>), (AUTEUR, <NOM, PRENOM>), (TRADUCTEUR, <PREFIX, NOM, PRENOM>), (EDITEUR, <NOM, PLACE>) }

XML BIBLIO comme suit:

```
<?XML VERSION="1.0" ENCODING="ISO-8859-1"?>
<BIBLIO SUJET="XML">
        <LIVRE ISBN="9782212090819" LANG="FR" SUJET="APPLICATIONS">
                <AUTEUR>
                        <NOM> BERNADAC </NOM>
                        <PRENOM> JEAN-CHRISTOPHE </PRENOM>
                </AUTEUR>
                <AUTEUR>
                        <NOM> KNAB </NOM>
                        <PRENOM> FRANÇOIS </PRENOM>
                </AUTEUR>
                <TITRE>CONSTRUIRE UNE APPLICATION XML</TITRE>
                <EDITEUR>
                        <NOM>EYROLLES</NOM>
                        <PLACE>PARIS</PLACE>
                </EDITEUR>
                <DATEPUB>1999</DATEPUB>
        </LIVRE>
        <LIVRE ISBN="9782212090529" LANG="FR" SUJET="GENERAL">
                <AUTEUR>
                        <NOM> MICHARD </NOM>
                        <PRENOM> ALAIN </PRENOM>
                </AUTEUR>
                <TITRE>XML, LANGAGE ET APPLICATIONS</TITRE>
                <EDITEUR>
                        <NOM>EYROLLES</NOM>
                        <PLACE>PARIS</PLACE>
                </EDITEUR>
                <DATEPUB>1998</DATEPUB>
        </LIVRE>
        <LIVRE ISBN="9782840825685" LANG="FR" SUJET="APPLICATIONS">
                <AUTEUR>
                        <NOM> PARDI </NOM>
                        <PRENOM> WILLIAM J. </PRENOM>
                </AUTEUR>
                <TRADUCTEUR PREFIX="ADAPTE DE L'ANGLAIS PAR">
                        <NOM> GUERIN </NOM>
                        <PRENOM> JAMES </PRENOM>
                </TRADUCTEUR>
                <TITRE>XML EN ACTION</TITRE>
                <EDITEUR>
                        <NOM>MICROSOFT PRESS</NOM>
                        <PLACE>PARIS</PLACE>
                </EDITEUR>
                <DATEPUB>1999</DATEPUB>
        </LIVRE>
</BIBLIO>
```

FIG. 3 – *Fichier XML correspondant à Bibliographie*.

Le schéma XML BIBLIO peut être représenté sous forme d'un arbre comme indiqué sur la figure 5. De cet arbre nous allons déduire la base de connaissances suivante :

R1 : SI Biblio ALORS At1= Lang, At2=Sujet, Isbn, Traducteur
R2 : SI Isbn ALORS At3=Titre, At4=DatePub, Auteur, Editeur
R3 : SI Auteur ALORS At5=Nom, At6=Prénom

R4 : SI Editeur ALORS At5=Nom, At7=Place
R5 : SI Traducteur ALORS At8=Prefix, At5=Nom, At6=Prénom

Notation :

- CELFACT : EF(i)=1 ; il s'agit d'un fait considéré établi
  EF(i)=0 ; il s'agit d'un fait à établir
  IF(i)=1 ; il s'agit d'un fait de type attribut=valeur
  IF(i)=0 ; il s'agit d'un fait de type sommet
- CELRULE : l'état interne IR peut être considéré comme coefficient de probabilité.

```
<!ELEMENT  BIBLIOSUJET(LIVREISBN+, TRADUCTEUR ?)>
        <!ATTLIST BIBLIOSUJET
                LANG #PCDATA #Required
                SUJET #PCDATA #Required>
<!ELEMENT LIVREISBN (AUTEUR+, EDITEUR )>
        <!ATTLIST LIVREISBN
                TITRE #PCDATA #Required
                DATEPUB #Required>
<!ELEMENT AUTEUR   NMTOKEN #Required)>
<!ATTLIST AUTEUR
                NOM #PCDATA #Required
                PRENOM #PCDATA #Required>
<!ELEMENT EDITEUR   NMTOKEN #Required)>
<!ATTLIST EDITEUR
                NOM #PCDATA #Required
                PLACE #PCDATA #Required>
<!ELEMENT LANG (#PCDATA)>
<!ELEMENT SUJET (#PCDATA)>
<!ELEMENT TRADUCTEUR  NMTOKEN #Required)>
<!ATTLIST TRADUCTEUR
                PREFIX #Required
                NOM #PCDATA #Required
                PRENOM #PCDATA #Required>
```

FIG. 4 – *Grammaire XML BIBLIO.*

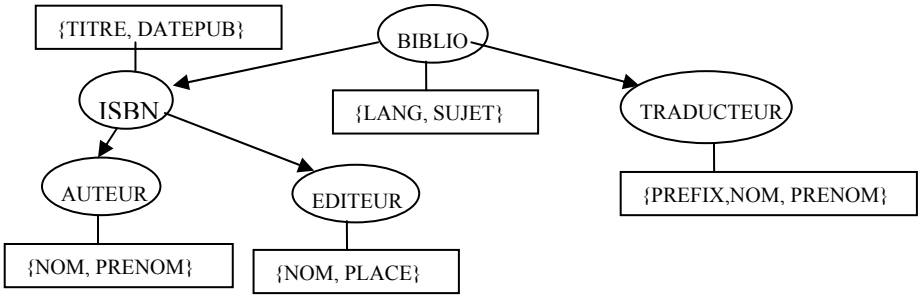


FIG. 5 – *Arbre T représentant le fichier XML BIBLIO.*

La configuration initiale de la machine CASI, représentant l'état courant de l'entrepôt cellulaire, est illustrée par les figures suivantes. Initialement, toutes les entrées des cellules dans la couche CELFACT sont passives (EF = 0), excepté ceux qui représentent la base des faits initiale (EF(1) = 1).

CELFACT(Faits)	*EF*	*IF*	*SF*
Biblio	1	0	0
At1=Lang	0	1	0
At2=Sujet	0	1	0
Isbn	0	0	0
Traducteur	0	0	0
At3=Titre	0	1	0
At4=DatePub	0	1	0
Auteur	0	0	0
Editeur	0	0	0
At5=Nom	0	1	0
At6=Prénom	0	1	0
At7=Place	0	1	0
At8=Prefix	0	1	0

CELRULE (Règles)	*ER*	*IR*	*SR*
R_1	0	1	1
R_2	0	1	1
R_3	0	1	1
R_4	0	1	1
R_5	0	1	1

FIG. 6 – *Les couches CELFACT et CELRULE.*

Le voisinage est exprimé par les matrices d'incidences Entrée/Sortie (voir figure 7) :

a. La relation d'entrée R_E : si (le fait i ∈ la prémisse de la Règle j) Alors $R_E(i,j) \leftarrow 1$
b. La relation de sortie R_S : si (le fait i ∈ la conclusion de la Règle j) Alors $R_S(i,j) \leftarrow 1$

R_E	R_1	R_2	R_3	R_4	R_5
Biblio	1				
At1=Lang					
At2=Sujet					
Isbn		1			
Traducteur					1
At3=Titre					
At4=DatePub					
Auteur			1		
Editeur				1	
At5=Nom					
At6=Prénom					
At7=Place	1				
At8=Prefix					

R_S	R_1	R_2	R_3	R_4	R_5
Biblio					
At1=Lang	1				
At2=Sujet	1				
Isbn	1				
Traducteur	1				
At3=Titre		1			
At4=DatePub		1			
Auteur		1			
Editeur		1			
At5=Nom			1	1	1
At6=Prénom			1		1
At7=Place					
At8=Prefix	1				

FIG. 7 – *Les matrices d'Entrée et de Sortie (Faits/Règles).*

Le cycle de base d'un moteur d'inférence, pour établir un fait F en chaînage avant, fonctionne traditionnellement comme suit :

1. Recherche des règles applicables (évaluation et sélection) ;
2. Choisir une parmi ces règles, par exemple R (filtrage) ;
3. Appliquer et ajouter la partie conclusion de R à la base des faits (exécution).

Le cycle est répété jusqu'à ce que le fait F soit ajouté à la base des faits, ou s'arrête lorsqu'aucune règle n'est applicable. Nous considérons G_0 la configuration initiale de notre automate cellulaire et, $\Delta=\delta_{fact}o\delta_{rule}$ la fonction de transition globale : $\Delta(G_0)=G_1$ si $\delta_{fact}(G_0)=G'_0$ et $\delta_{rule}(G'_0)=G_1$. Supposons que $G = \{G_0, G_1,\ldots, G_q\}$ est l'ensemble des configurations de notre machine cellulaire. L'évolution discrète de l'automate, d'une génération à une autre, est définie par la séquence $G_0, G_1,..., G_q$, où $G_{i+1} = \Delta(G_i)$.

Essayons, par exemple, d'augmenter la classe BIBLIO par un nouvel élément qui est l'adresse de l'auteur. Ceci se traduit par l'intégration d'un nouveau schéma représenté aussi

sous forme d'arbre pour lequel nous génèrerons la matrice d'incidence d'entrée et de sortie. Pour notre machine ceci revient à établir un nouveau fait 'Adresse' avec notre base de connaissances. En utilisant le principe cellulaire de la machine CASI, nous obtenons un état global des deux couches, CELFACT et CELRULE, après évaluation, sélection et filtrage en mode synchrone : application de la première loi de transition δ_{fact}. De même, après l'application de la seconde loi de transition, δ_{rule}, nous obtenons la configuration G_1.

La fonction Δ constitue une loi de transition globale en chaînage avant, qui transforme itérativement, notre automate cellulaire d'une configuration initiale G_0 en une autre configuration finale en passant par une configuration intermédiaire.

5 Conclusion

D'une manière générale, les données diffusées sur le web sont tellement hétérogènes et complexes dans leurs diversités que le problème de leur intégration et de leur homogénéisation devient prioritaire et suscite l'intérêt de plusieurs chercheurs. En effet, l'étude bibliographique montre que beaucoup de travaux on été réalisés en vue d'apporter des solutions sûres et même parfois palliatives.

C'est pour ce fait, que notre souci, pour l'instant, est la structuration et l'intégration des données hétérogènes au sein d'un même entrepôt. Notre solution est d'exploiter la richesse de la machine cellulaire CASI, pour garantir, d'abords, une intégration cellulaire au contexte multidimensionnel des données d'un entrepôt, et ensuite, appliquer des requêtes d'analyse OLAP pour l'extraction des connaissances et la fouille des données.

Références

Atmani, B., B. Beldjilali (2007). *Knowledge Discovery in Database: Induction Graph and Cellular Automaton*. Computing and Informatics Journal, Vol.26, N°2 171-197.

Beneventano, D., S. Bergamaschi,, S. Castano, V. De Antonellis, A. Ferrara, F. Guerra, F. Mandreoli, G.C. Ornetti, et M. Vincini (2002). *Semantic Integration and query optimization of heterogeneous data sources*. LNCS 2426, pp 154-165.

Boussaid O., R. Ben Messaoud, R. Choquet et S. Anthoard (2006). *Conception et construction d'entrepôts en XML*. Dans la RNTI correnspondant à la 2ième journée francophone sur les entrepôts de données et l'analyse en ligne EDA'06 Versaille 19.

Da Silva, A.S., I.M.R. Evangelista Filha, A.H.F Laender et D.W Embley (2002). *Representing and querying semistructured Web Data Using Nested Tables With structural Variants*. LNCS-2503 : 21st International conference on conceptual modelling ER, pp 135-151, Octobre, Tampere Finland.

Delobel, C., C.Reynaud, M.C Rousset, J.P Sirot et D. Vodislav (2003). *Semantic integration in Xyleme : A uniform tree-based approach*. Data and Knowledge Engineering 44, pp 267-298.

Garcia-Molina, H., Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. sagiv, J. Ullman et J. Widom.(1995). *The STIMMIS approach to mediation : Data Models and Languages* . CiteSeer.

Hamdoun, S., F. Boufarès et M. Badri (2007). *Construction et maintenance des entrepôts de données hétérogènes*.. e-TI - la revue électronique des technologies d'information, Numéro 4, 23juin , http://www.revue-eti.netdocument.php?id=1331.

Kim, H.H. et S.S Park (2003). *Building a Web-enabled Multimedia Data warehouse*». LNCS 2713, pp 594-600.

Laurent, D., J. Lenchtenboer-Ger, N. Spyratos et G. Vossen (2001). *Monotonic Complements for Independent Data Warehouses.* The International Journal of Very Large Data Base VLDB, volume 10 issue 4, pp 295-315.

Limam, L. (2007). *Découverte Automatique de Mappings fondée sur les Requêtes dans un environnement P2P.* Laboratoire d'InfoRmatique en Image et Systèmes d'information.

Maibaum, M., L. Zamboulis, G. Rimon, C. Orengo, N. Martin, et A. Poulovassilis (2005). *Cluster based Integration of Heterogeneous Biological Databases using the AutoMed toolkit*. In Proceedings of DILS'05.

Saccol, D.d.B. et C.A Heuser (2002). *Integration of XML Data*, LNCS 2590, pp 68-80.

Siméon (2000). *Data Integration with XML : A Solution for Modern Web Applications*. Lecture at Temple University, March (2000)

Sorlin, S. et C. Solnon (2004). *Mesurer la similarité de graphes pour la recherche d'informations* . Dans http://www-clips.imag.fr/mrim/User/catherine.berrut

Tomasic,A., L. Rashid et P. Valduriez (1997). *A data model and query processing techniques for scaling access to distributed heterogeneous databases in Disco.* IEEE Transactions on computers, special issues on Distributed Computing Systems.

Widom, J (1995). *Research Problems in Data Warehousing*. In Proceedings of the 1995 International Conference on Information and Knowledge Management (CIKM), Baltimore, Maryland.

Zerdazi (2007). *Cadre formel pour l'appariement de schémas XML pour l'intégration de données*. Thèse de Doctorat, Université Paris VIII.

Summary

To initiate a knowledge extraction process from complex data must be integrated and then represent complex data in a form suited to analysis techniques online or data mining. The issue of integration, modeling, structuring and knowledge extraction from data requires a complex methodology and generic tools adapted. On the other hand, the cellular automaton is a discreet dynamic system capable of simulating an overall behaviour using simple and local transition rules. Indeed, despite the simplicity of the rules that define, it shows many unpredictable phenomena, which are a priori difficult obtained by conventional analytical methods. The idea of using the formalism of cellular automata to solve the problem of integration in data warehouses, is the starting point of our thinking on a new principle cell capable of automatically feed a data semi-structured in warehouse.

Based on this observation, we propose in this article our problem on designing a tool cellular of integration of semi-structured data in a data warehouse.

Entreposage et analyse de données complexes: le cas DBLP

Doulkifli Boukraa*
Riadh Ben messaoud**
Omar Boussaid***

*Département d'informatique, faculté des sciences de l'ingénieur
Université de Jijel, BP 98 Ouled Aissa, Jijel, 18000, Algérie
d_boukraa@mail.univ-jijel.dz
**Université de Nabeul, Tunisie
rbenmessaoud@chirouble.univ-lyon2.fr
***Laboratoire ERIC, Université Lumière Lyon2
5 avenue Pierre Mendès-France, 69676 Bron Cedex, France
omar.boussaid@univ-lyon2.fr

Résumé. Nous présentons dans cet article un ensemble d'outils pour l'entreposage et l'analyse de données complexes appliqués à la base des références bibliographiques DBLP. Le premier est un outil ETC (Extraction, Transformation, Chargement) qui permet d'alimenter un entrepôt de données XML (XDBLP) à partir d'une source de données XML. Le deuxième outil permet le stockage natif et relationnel des données XML de l'entrepôt et d'y appliquer une charge de requête afin de mesurer la performance des requêtes pour chaque solution. Le troisième outil permet de générer des cubes XML pour répondre à des besoins spécifiques d'analyse.

1 Introduction

Les outils associés aux entrepôts de données classiques permettent de nos jours de satisfaire les différents besoins de décideurs sur plusieurs plans : performance des requêtes, interfaces conviviales d'analyse en ligne, etc. Cependant, ces outils ne sont plus adéquats lorsqu'il s'agit de tenir compte des différents types de données qui existent dans l'entreprise. Nous qualifions ces données de *données complexes*. L'entreposage et l'analyse de données complexes s'inscrit dans les nouveaux défis et perspectives des entrepôts de données tant sur le plan théorique que pratique. En particulier, la modélisation de l'entrepôt de données complexes qui est un élément important du processus décisionnel doit permettre de capturer et conserver la complexité des données. En revanche, les outils associés aux entrepôts de données complexes doivent être au même niveau de convivialité d'utilisation que les outils d'entrepôts classiques.

L'objet de cet article est de présenter un ensemble d'outils pour l'entreposage et l'analyse de données complexes associés au modèle d'entrepôts de données complexes que nous avons proposé dans (Boukraâ et al. 2007) et (Boukraâ et al., 2008). L'objectif de notre travail est de

valider notre modèle à travers son implémentation sur un cas pratique, cela d'une part. D'autre part, nous tentons, à travers les résultats d'utilisation de ces outils d'améliorer davantage le modèle, notamment sur le plan du stockage physique, de l'optimisation des requêtes décisionnelles et de l'analyse en ligne. Cet article est structuré comme suit. En section 2, nous présentons le concept de données complexe et les problèmes liés à leur entreposage ainsi que les travaux de recherches associés. En section 3, nous présentons le domaine d'application et le modèle d'entrepôt de données complexes appliqué à ce domaine. Par la suite, nous décrivons les différents travaux d'entreposage associés à l'entrepôt XDBLP. En section 4, nous présentons les outils développés à travers des captures d'écran. Nous terminons par une conclusion et des travaux futurs.

2 Entreposage des données de données complexes

Dans cette section, nous présentons le concept de données complexes, et des problèmes liées à leur entreposage. Par la suite, nous exposons des travaux de recherches liés exclusivement à un type d'entrepôt de données complexes, à savoir les entrepôts de données XML. Aussi, nous classons ces travaux selon quatre aspects du processus d'entreposage.

2.1 Données complexes et problématique d'entreposage

Le concept de donnée complexes varie selon les chercheurs, y compris au sein d'une même communauté. Les données sont qualifiées de complexes si elles sont multiformats, multistructures, multisources, multimodales ou multiversions. L'entreposage des données complexes découle du besoin d'intégrer, dans le processus décisionnel, des données de quelconque nature qu'elle soit, d'où l'émergence de nouvelles générations d'entrepôts de données : entrepôt de données XML, entrepôts de données spatio-temproels, etc. Dans ce travail, nous traitons exclusivement les entrepôts de données XML.

A l'heure actuelle, il n'existe pas de définition communément acceptée d'un entrepôt de données XML. Cela est dû au fait que XML aie été utilisé différemment dans les travaux de recherche: format d'origine des données sources, formalisme d'intégration de données hétérogènes, formalisme logique de l'entrepôt de données ou format d'échange entre des entrepôts de données hétérogènes. Dans cet article, nous classons ces différents travaux en quatre classes : la modélisation de l'entrepôt XML, le processus ETC (extraction, transformation, chargement), le stockage physique et l'analyse en ligne OLAP/XML.

2.2 Travaux sur la modélisation d'entrepôts XML

Il existe différentes approches de modélisation d'un entrepôt de données XML. Pokorný (2001) part d'un ensemble de *DTD-core* qui décrivent le noyau d'un ensemble de DTD sources. Il utilise un mécanisme de vues afin de définir un schéma en étoile XML avec des hiérarchies multiples. Le mécanisme de vues est également présent dans les travaux de (Baril et Bellahsène, 2003) où les auteurs spécifient des vues matérialisées sur un ensemble de sources de données XML. L'entrepôt de données XML est alors décrit par un document XML qui fait références aux vues matérialisées et aux sources de données. Dans (Golfarelli et al., 2001) et (Vrdoljak et al., 2003), les auteurs présentent des approches similaires pour la

conception de magasins de données ou entrepôts Web respectivement à partir de DTD et de Schémas.

Leurs approches comportent la création de graphe de DTD ou de Schéma XML, le choix des faits et la détermination des dimensions et des mesures pour chaque fait. Nassis et al. (2004) utilisent UML pour décrire un contexte d'analyse appelé fait significatif (xFACT). Les dimensions ne sont pas matérialisées mais exprimées sous forme de vues sur xFACT et de fait, sont qualifiées de dimensions virtuelles. Dans (Boussaid et al. 2006), les auteurs utilisent XML comme formalisme logique d'un entrepôt de données XML. Ils définissent un fait comme un élément XML où les mesures sont modélisées sous forme d'attributs et les dimensions sous forme d'éléments XML imbriqués. Rusu et al. (2005) basent leur approche pour la construction d'un entrepôt XML sur le langage XQuery. L'interrogation de sources de données XML permet alors d'obtenir un document XML représentant le fait et plusieurs documents XML représentant les dimensions.

Dans (Boukraâ et Boussaid, 2007) et (Boukraâ et al., 2008), nous avons proposé un modèle d'entrepôt de données complexes adapté à des sources XML et décrit au niveau logique en XML également. Ledit modèle répond à un ensemble de besoins de modélisation non pris en charge par les modèles d'entrepôts existants. Le modèle est centré autour des concepts d'*objet complexe*, de *relation complexe*, de *hiérarchie d'objets* et de *hiérarchie d'attributs*.

2.3 Travaux sur l'ETC

Les travaux sur le processus ETC dans un entrepôt de données XML ne sont pas nombreux, et les outils proposés sous ce nom sont souvent des outils d'intégration de données, qui s'adaptent plus ou moins à des sources de données XML. Dans un contexte d'entreposage, nous citons principalement deux travaux. Le premier travail est basé sur le langage XQuery lequel est utilisé pour l'interrogation de documents XML sources, leur transformation et la production de documents XML conformes à un schéma XML en étoile. Le deuxième travail figure dans (Tsheke Shele, 2007). Il consiste à appliquer un fichier XSLT sur une source XML pour extraire et transformer les données. En plus, l'auteur propose un générateur automatique de fichier XSLT en tenant compte des schémas XML source et destination. Ce concept de générateur automatique de fichier XSLT rend l'ETC utilisable tant dans les entrepôts de données XML que dans d'autres domaines.

2.4 Travaux sur le stockage physique

Un entrepôt de données est caractérisé, entre autres, par son grand volume de données ce qui influe sur la performance des requêtes. La conception du stockage physique est de fait une activité du processus d'entreposage à ne pas négliger. Dans le contexte des entrepôts de données XML, peu travaux détaillent la conception physique. En l'occurrence, dans (Xylème, 2002), les auteurs proposent un système orienté stockage de grands volumes de documents XML dans un entrepôt. La performance de requêtes est alors optimisée via l'utilisation d'un index appelé XyIndex. Le système DAWAx (Baril et Bellahsène, 2003) pour sa part permet de stocker des données XML dans une base de données relationnelle. Des règles de correspondance sont alors définies entre le modèle de l'entrepôt XML et les concepts du modèle relationnel.

entreposage et analyse de données complexes

2.5 Travaux sur l'analyse OLAP/XML

A la différence des outils OLAP traditionnels, les outils XML/OLAP ne sont pas simples à réaliser. En effet, étant donné la nature flexible de données XML, les opérateurs OLAP conventionnels, notamment l'agrégation, ne sont plus appropriés aux données XML et sont, de ce fait, à redéfinir. Des travaux de définition de tels opérateurs figurent dans (Wang et al, 2005) et (Wiwatwattana et al, 2007).

D'autres techniques utilisent des médiateurs pour effectuer des opérations OLAP sur des données XML. Par exemple, Niemi et al. (2002) proposent d'analyser les données XML d'un entrepôt XML via un *serveur de collection* qui est un médiateur entre les entrepôts et un serveur OLAP basé sur la technologie MDX de Microsoft. Un travail similaire figure dans (Jensen et al, 2001) avec la différence que les données à analyser peuvent être en XML ou en relationnel. Dans le même contexte, Li et An (2005) proposent d'intégrer virtuellement des données XML sources en les transformant en diagrammes UML pour ensuite les interroger.

3 Entreposage des données de DBLP

Dans cette section, nous présentons la base DBLP avec des exemples de besoins d'analyse nécessitant sa restructuration. Nous détaillons par la suite les éléments qui composent le processus d'entreposage propre à DBLP.

3.1 Présentation de la base DBLP

DBLP[1] est un serveur Web qui recense les références aux publications scientifiques d'informatique. C'était à l'origine un serveur bibliographique spécialisé en une partie de l'informatique, à savoir *les systèmes de base de données et la programmation logique.* L'abréviation DBLP signifiait alors DatadBase Systems and Logic Programming. Par la suite, la base de données a été étendue à d'autres champs de l'informatique. On peut lire maintenant DBLP comme Digital Bibliography & Library Project.

3.2 Besoins d'entreposage en DBLP

La base DBLP est accessible à travers une interface Web qui permet d'effectuer des recherches de références selon plusieurs critères, notamment par auteur, par conférence et par journal. La base est également disponible en format XML sous la forme d'un seul fichier (dblp.xml)[2] mis à jour régulièrement.

La base DBLP est riche en informations et peut être exploitée à des fins décisionnelles. Un exemple de besoin décisionnel concerne l'intégration de chercheurs dans un laboratoire de recherche sur la base de l'analyse du nombre de leurs publications et/ou de la qualité de leurs publications. Un autre exemple concerne l'évaluation d'une conférence, d'un journal ou d'une publication individuelle par rapport à ses auteurs.

[1] La base DBLP est accessible à http://dblp.uni-trier.de

[2] Le fichier peut être téléchargé à partir de l'URI : http://dblp.uni-trier.de/xml/dblp.xml

Cependant, ni l'interface Web ni le fichier dblp.xml ne s'accommodent à la satisfaction de tels besoins d'analyse. C'est dans ce contexte que nous proposons un ensemble d'outils d'entreposage des données de DBLP faisant partie d'un même processus d'entreposage (figure 1) et centrés autour du modèle d'entrepôt de données complexe que nous avons proposé dans (Boukraâ et Boussaid, 2007) et (Boukraâ et al., 2008). Dans la suite de cet article, nous résumons l'essentiel des fonctions assurées par les outils.

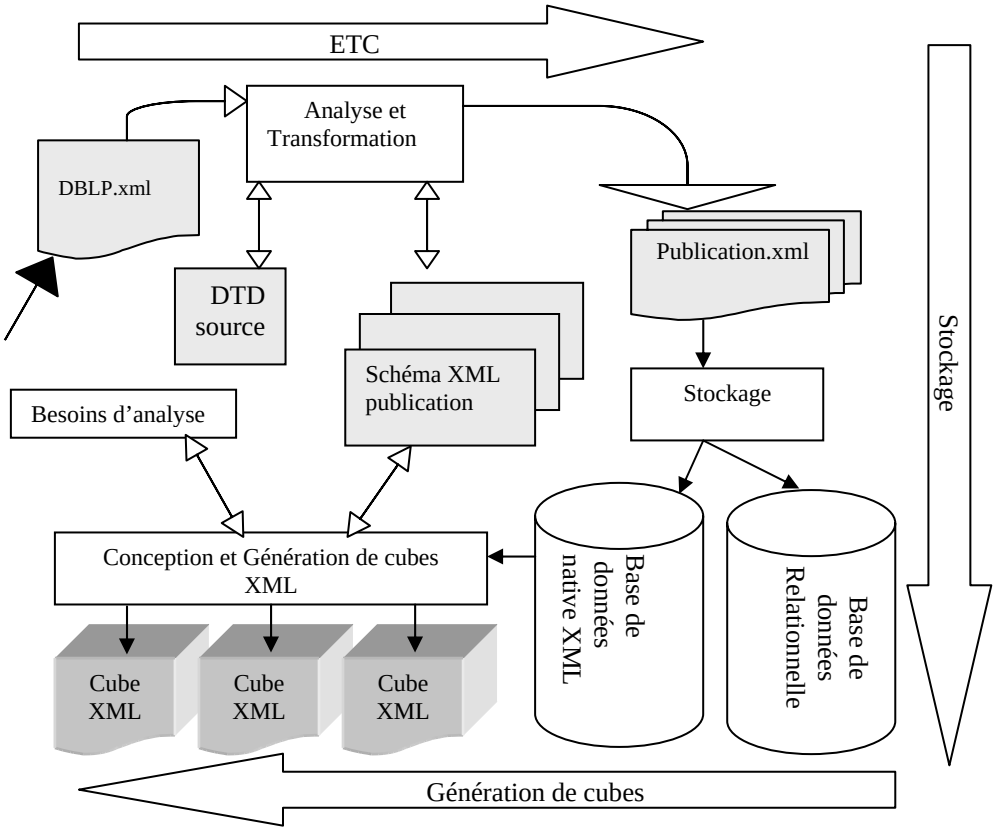


FIG. 1 – *Processus d'entreposage de DBLP.*

3.3 Modélisation de l'entrepôt de données complexes XDBLP

3.3.1 Rappel du modèle d'entrepôt de données complexes

Dans (Boukraâ et al. 2007) et (Boukraâ et al., 2008), nous avons proposé un modèle conceptuel pour les entrepôts de données complexes. Le concept-clé du modèle est l'*objet complexe*. Un objet complexe résulte de la conceptualisation d'une entité possédant une structure complexe (document, images, etc), et qui est au centre d'intérêt des décideurs, soit en tant que sujet d'analyse ou axe d'analyse. Dans le modèle que nous proposons, nous limi-

tons la description de la structure interne d'un objet complexe aux notions de clé et d'attributs. Hormis la clé d'un objet complexe, les autres attributs peuvent être multivalués, et sont organisés de différentes manières. Parmi ces organisations, nous nous sommes intéressé à l'organisation hiérarchiques d'attributs, en nous avons définis la notion de *hiérarchie d'attributs*. Les objets complexes de même nature appartiennent à une classe. Nous avons défini le concept de *relation complexe* qui permet de modéliser différents liens sémantiques entre des pairs de classes d'objets. Ces liens peuvent être des liens d'associations, d'héritage, etc. Enfin, nous avons défini le concept de *hiérarchie d'objets complexes* pour modéliser l'organisation hiérarchique des classes d'objets.

Le schéma (multi)dimensionnel est alors constitué d'un ensemble de classes d'objets complexes, d'un ensemble de relations entre les objets complexes, d'un ensemble de hiérarchies d'attributs et d'un ensemble de hiérarchies d'objets.

Notons que dans le schéma dimensionnel, nous n'évoquons pas le concept de fait. Autrement dit, le modèle n'est *a priori* composé que de dimensions. Le concept de fait intervient au moment de l'analyse, où l'on procède à la désignation d'une classe particulière pour jouer le rôle de fait, et d'un sous-ensemble de classes pour jouer le rôle de dimensions.

En outre, l'explicitation des hiérarchies d'attributs et des hiérarchies d'objets permet de donner le choix au concepteur de cube complexe quant aux hiérarchies à impliquer dans l'analyse. Cela résulte en un cube *sur mesure*, i.e. répondant aux besoins d'analyse.

3.3.2 Le schéma conceptuel de XDBLP

L'analyse détaillée de DBLP a permis de dégager plusieurs classes d'objets complexes, des relations complexes, des hiérarchies d'objets et aussi des hiérarchies d'attributs, lesquelles nous semblé les plus intéressantes pour l'activité d'analyse. Les objets complexes que nous avons identifiés sont[3] *Publication*, *Conference*, *Proceedings*, *Journal*, *Journal_volume*, *Journal_author*, *Time*. Les relations complexes sont *Authored_by* qui lie *Publication* à *Author* et *Date_pub* qui lie *Publication* à *Time*. Nous avons identifié au sein de l'objet Time une hiérarchie d'attribut composée des attributs *day* <*month* < *year* < *All*. Pour les hiérarchies d'objets, nous en avons identifié deux: la hiérarchie *h_ PubConf* composée de *Publication* < *Proceedings* < *Conference* < *All* et la hiérarchie *h_PubJour* composée de *Publication* < *Journal_Number* < *Journal_Volume* < *Journal* < *All*.

La figure 2 illustre le schéma conceptuel de l'entrepôt XDBLP, formalisé en UML. Dans ce schéma, nous regroupons les classes qui rentrent dans la définition de chaque objet complexe, telles que modélisé dans (Boussaid et al, 2003) en un paquetage UML pour chaque classe d'objet complexe.

[3] Nous utilisons des dénominations anglaises pour rester conformes au contenu de DBLP

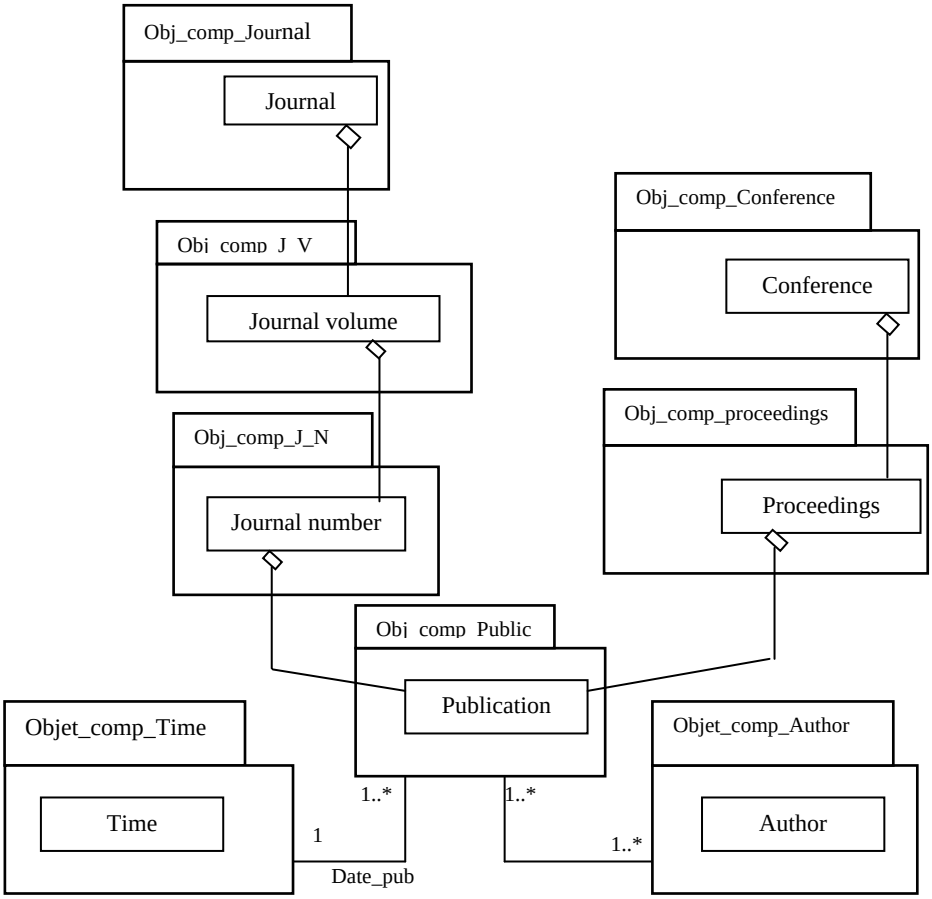


FIG. 2 – *Schéma conceptuel de l'entrepôt XDBLP.*

3.3.3 Le schéma logique de XDBLP

Le schéma logique de l'entrepôt XDBLP est issu de l'application de règles de correspondance définies dans (Boukraâ et Boussaid, 2007). Ainsi, chaque classe d'objets complexes (resp. de relations) sera décrite par un document XML regroupant l'ensemble des instances d'objets de la même classe (resp. des instances de la même relation). Par contre, les hiérarchies d'objets ou d'attributs ne sont décrites de manière indépendante mais plutôt à l'intérieur de la définition des objets ou attributs impliqués dans les hiérarchies.

3.4 La phase ETC dans XDBLP

La phase ETC permet d'alimenter l'entrepôt XDBLP à partir d'une source de données à travers un ensemble de transformations. Nous décrivons ci-après les principaux éléments de cette phase.

3.4.1 La source de données

La source de données de l'entrepôt XDBLP est constituée d'un seul document dblp.xml, décrit par une DTD. Le fichier recense huit types de publications : *Article*, *Book*, *Incollection*, *Mastersthesis*, *Inproceedings*, *Proceedings*, *Phdthesis* et *www*. Chaque publication correspond à un élément XML imbriqué directement sous la racine, et possèdant deux attributs : *key* qui représente la clé de la publication et *mdate* qui représente la date de sa dernière modification.

3.4.2 Les différences entre le schéma source et le schéma de l'entrepôt

Afin de mener à bien la phase d'ETC, nous avons étudié les principales différences entre le schéma source (la DTD du fichier dblp.xml) et les schémas cibles (les Schémas XML des différentes classes d'objets et relations). Les principales différences se résument comme suit :

- Le typage de données : étant donné que le fichier source est décrit par une DTD, les types de données sont limités aux chaînes de caractères. Par contre, les Schéma XML de l'entrepôt permettent d'attribuer des types de données adéquats notamment pour les dates ;
- Les clés : au niveau de la source de données, les seuls éléments possédant une clé sont ceux qui représentent les publications (ex : article, inproceedings, …). Par contre, au niveau du schéma cible, et conformément au modèle d'entrepôt que nous avons proposé, tout objet possède une clé (publication, author, date, …) ;
- La structuration XML des concepts du domaine : au niveau du fichier source, le type de la publication correspond à un élément XML imbriqué directement sous l'élément racine. Par contre, au niveau du schéma cible, toutes les publications possèdent le même nom d'élément XML (*Publication*) ; le type de publication est alors représenté comme un sous-élément appelé « type » et imbriqué sous l'élément *Publication*.

3.4.3 Le processus ETC de l'entrepôt XDBLP

Eu égard des différences entre le schéma de la source de données et le schéma de l'entrepôt, nous avons dressé la liste de toutes les transformations à effectuer au cours du processus d'ETC. Le processus ETC consiste à lire séquentiellement le fichier source de données. Pour chaque élément-fils de la racine, on extraits les informations qui le décrivent, en particulier sa clé. Ces informations sont disséminées sur les différents documents XML qui constituent l'entrepôt XDBLP. Avant l'ajout d'un élément, on en teste l'existence préalable dans le document correspondant dans l'entrepôt. La recherche se fait sur la base de la clé pour la publication puisque cette dernière possède une clé dans le fichier source. Par

contre, pour les autres classes d'objets, la recherche se fera sur la base de plusieurs informations combinées, et si l'élément n'existe pas, il y aura une génération de clé. En outre, pour les classes d'objets (resp. les classes d'attributs) entrant dans la composition d'une hiérarchie d'objets (resp. d'attributs), on créé la hiérarchie au fur et à mesure de la création des objets (resp. des attributs). L'exception à ce processus réside dans le sous-élement *Proceedings* qui est décrit au même niveau que les autres publications. A la lecture de chaque élément XML *Proceedings*, il n'y aura pas création de publications mais seulement d'objets *Proceedings* ainsi que des autres objets qui lui sont associés dans les hiérarchies s'ils n'existent pas encore.

3.5 Le stockage physique

La phase ETC résulte en la génération d'un ensemble de fichiers XML conformes au modèle de l'entrepôt XDBLP. Nous avons choisi de ne pas stocker ces fichiers directement au niveau d'une base de données afin de pouvoir étudier plusieurs solutions de stockage. En effet, nous avons opté pour deux solutions : le stockage natif XML et le stockage relationnel. Le stockage natif XML étant pris en charge par le système de base de données choisi, nous avons conçu le schéma relationnel qui correspond aux éléments du modèle d'entrepôt de données complexes. Le schéma est décrit comme suit :

- Chaque classe d'objet est décrite par une table relationnelle, qui porte alors le nom de la classe. Chaque table est identifiée par une clé primaire qui correspond à la clé de l'objet complexe ;
- Chaque attribut décrivant un objet complexe est représenté sous forme de colonne de la table relationnelle. Dans le cas précis de l'entrepôt XDBLP, ces colonnes ne possèdent pas une structure complexe. Donc, nous n'avons pas eu recours à l'utilisation de types de données relationnels adéquat aux type XML (ex : XMLType ou CLOB) ;
- Chaque relation complexe correspond à une table relationnelle portant le nom de la relation et identifiée par une clé primaire composée des clés des objets complexes ;
- Les hiérarchies d'objet sont décrites par une table relationnelle appelée ObjHierarchy, identifiée par le nom de la hiérarchie. L'appartenance d'un objet complexe à une hiérarchie est matérialisée par une table relationnelle identifiée par la clé de la hiérarchie et la clé de l'objet. Cette table contient trois attributs : rollsup, level et value;
- Les hiérarchies d'attributs sont décrites par une table relationnelle appelée AttHierarchy et identifiée par le nom de la hiérarchie. Dans le cas précis de XDBLP, la seule hiérarchie d'attribut étant la hiérarchie *day* < *month* < *year*, nous avons du disséminer les composants de la hiérarchie en trois tables relationnelles *Day*, *Month* et *Year*. L'appartenance de chaque attribut membre de la hiérarchie à cette hiérarchie est alors matérialisée par une table relationnelle identifiée par la clé de la hiérarchie et la clé de l'attribut membre de la hiérarchie.

3.6 La génération de cubes complexes

3.6.1 Processus de conception du cube

C'est au moment de la conception d'un cube XML que l'on sélectionne la classe d'objet servant de fait à analyser et les classes d'objets servant de dimensions d'analyse. Le concepteur du cube sélectionne une classe d'objets complexes parmi la liste des classes de l'entrepôt XDBLP et le désigne comme fait. Cette sélection entraine la réduction de la liste des relations complexes pour ne contenir que les relations où figure la classe *fait*. Le concepteur peut ainsi sélectionner les relations qui l'intéressent pour l'analyse. Cela entraîne la réduction de la liste des classes d'objets complexes pour ne contenir que les classes liées à la classe *fait* à travers les relations sélectionnées. Ces classes sont alors considérées comme des classes *axes*. Une fois la liste des classes d'objets *fait* et *axes* sélectionnées, la liste des hiérarchies de classes d'objets où figurant ces classes objets devient disponible pour la sélection. Aussi, la sélection d'une hiérarchie entraîne l'ajout des classes d'objets entrant dans sa composition. Une fois ces choix effectués, il devient possible de générer le cube XML. Cette génération passe par des transformations au niveau du document XML qui représente l'objet fait. La figure 3 illustre le diagramme d'activité en UML de la conception d'un cube XML.

4 Présentation des outils associés

Les outils d'entreposage des données de DBLP ont été développés au sein d'un même environnement afin de faciliter leur intégration dans un outil global. Dans cette section, nous présentons l'environnement et décrivons les trois outils développés.

4.1 Environnement de développement

Les outils pour l'entreposage et analyse de données ont été développés sous Oracle Jdeveloper studio 10.1.3.0 Pour l'analyse des documents XML, nous avons utilisé le parseur SAX lequel est mieux adapté à l'analyse de document XML volumineux. Nous avons opté pour la base de données Open Source eXist pour le stockage des documents XML de l'entrepôt XDBLP en format natif XML. En ce qui concerne le stockage relationnel, nous avons choisi l'édition *express edition* du SGBDR Oracle dans sa version 10g.

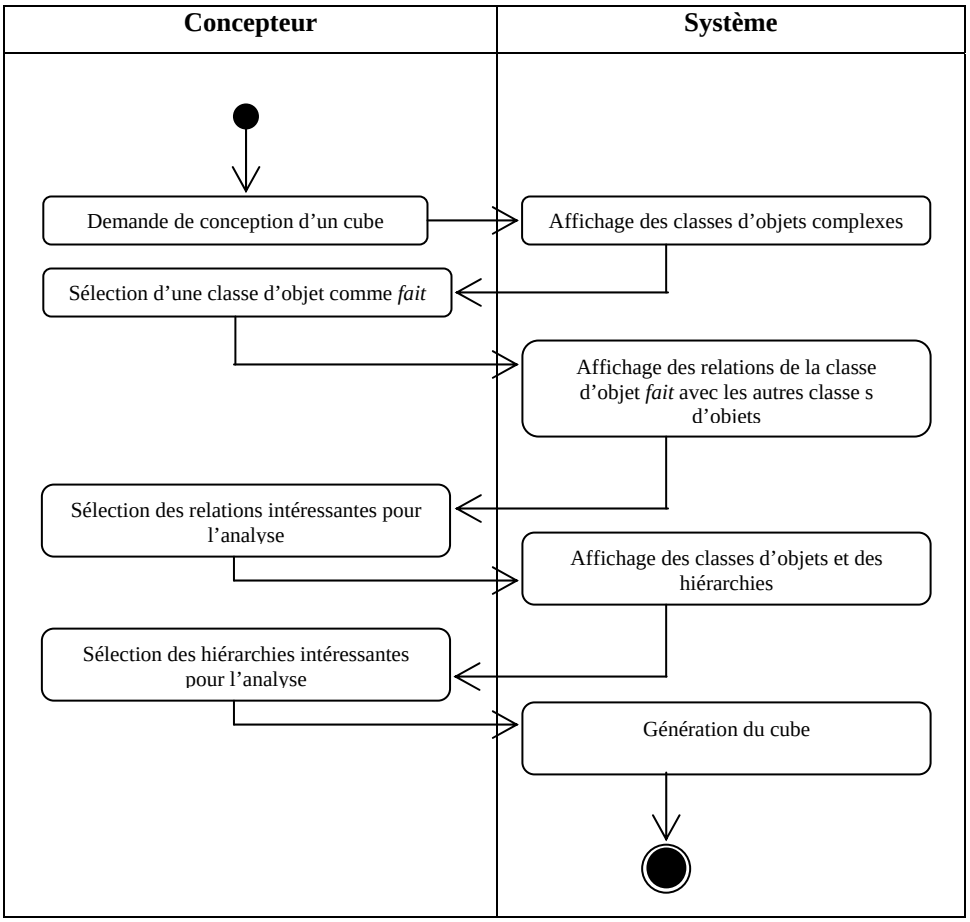


FIG. 3 – *Le processus de conception et de génération d'un cube XML dans XDBLP.*

4.2 Outil ETC

L'outil ETC permet de charger et de valider le fichier source de l'entrepôt XDBLP (dblp.xml) et de lancer le chargement initial. Une barre de progression témoigne du déroulement de l'opération (figure 3). Au stade actuel du développement, l'outil ne permet pas d'effectuer des opérations de rafraichissement périodique.

entreposage et analyse de données complexes

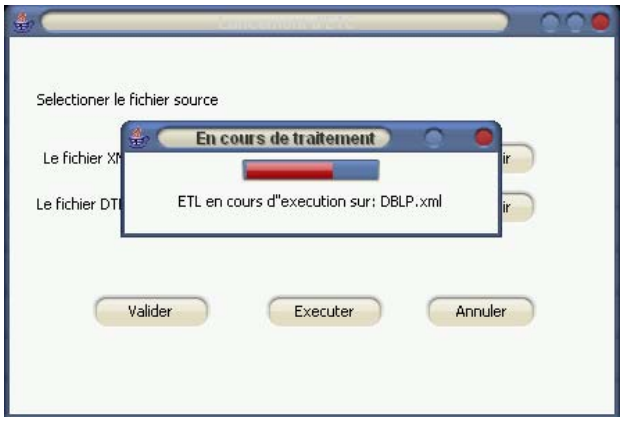


FIG. 4 – *Interface de progression de l'ETC dans XDBLP.*

4.3 Outil de stockage physique avec étude de performance

L'outil de stockage physique permet de charger les documents XML de l'entrepôt XDBLP issus de la phase ETC dans la base de données eXist et dans la base de données Oracle. Une fois que les documents sont stockés, l'entrepôt XDBLP peut être interrogé à travers une interface graphique qui permet de choisir une requête dans une liste de 7 requêtes décisionnelles. La requête est affichée en langue naturelle et peut être exécutées en XQuery dans eXist ou en SQL dans Oracle. A la fin de cette étape, il est possible de visualiser les temps de réponses des requêtes sur les deux bases de données. L'outil nous a permis de constater la performance du stockage relationnel par rapport au stockage XML natif, ce qui est naturel du fait que les bases de données XML natives ne sont pas aussi matures sur le plan de la performance des requêtes que les bases de données relationnelles.

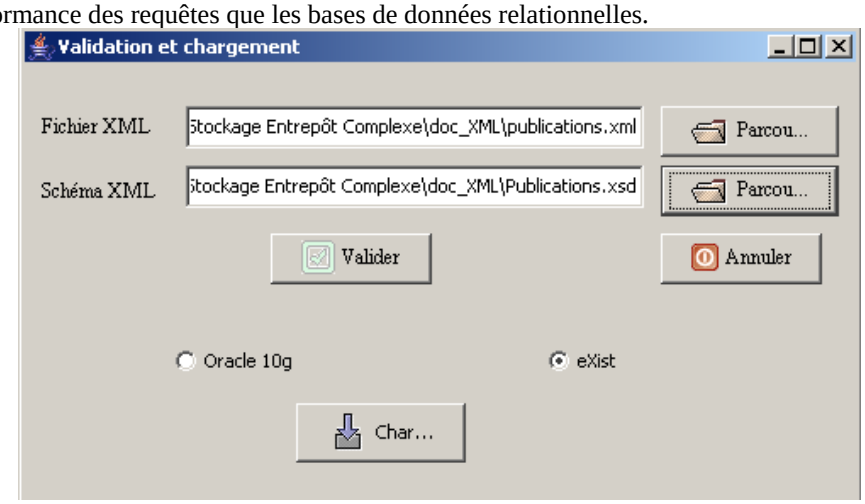


FIG. 5 – *Interface permettant le chargement et stockage des documents XML de l'entrepôt XDBLP dans les bases de données eXist et Oracle.*

4.4 Outil de génération de cubes XML

L'outil de génération de cube offre une interface dynamique qui permet de filtrer les listes de classes d'objets, de relations et de hiérarchies selon le besoin d'analyse (figure 6). En outre, une interface graphique permet d'interroger le contenu du cube en exécutant un ensemble de requêtes décisionnelles. Au stade actuel du développement, le cube ne permet pas d'effectuer des opérations similaires aux opérations OLAP, ce qui peut être inscrit comme travaux futurs.

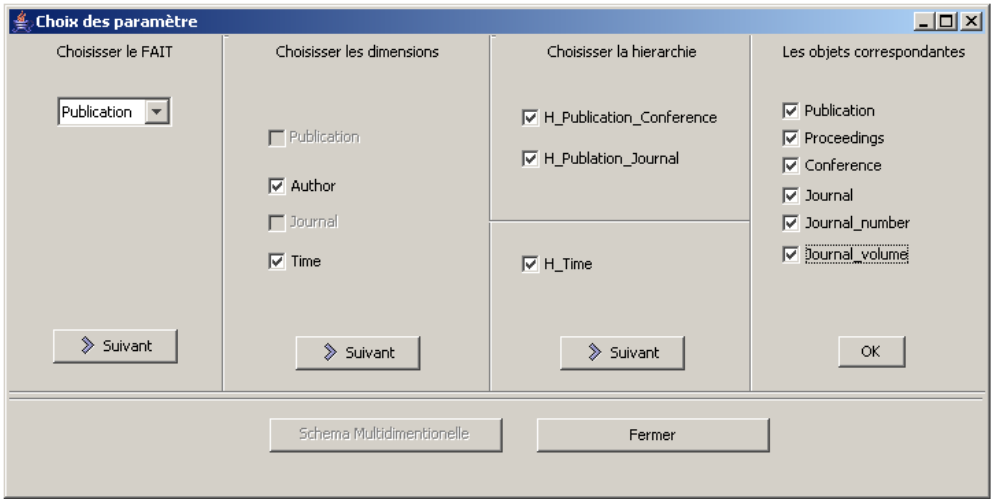


FIG. 6 – *Interface de conception d'un cube XML à partir de l'entrepôt XDBLP.*

5 Conclusion et directions futures

Dans cet article, nous avons présenté un ensemble d'outils pour un entrepôt de données XML, appelé *XDBLP*. Ces outils gravitent autour d'un modèle d'entrepôt de données complexes lequel a été appliqué à la base des références bibliographiques DBLP. Ces outils restent ouverts à plusieurs améliorations et extensions. L'outil ETC doit être étendu à la fonction de rafraîchissement périodique de l'entrepôt XDBLP, mais aussi à la possibilité de travailler directement sur le fichier dblp.xml localisé sur le serveur d'origine du projet DBLP, sans devoir le télécharger à chaque fois. Le développement de l'outil de stockage physique en relationnel doit être étendu pour la prise en compte de structures plus complexes d'objets, notamment par l'exploitation de types de données adéquats comme CLOB et XMLtype. Aussi, la mise en œuvre de nouveaux modèles de stockage physique nous permettra à travers cet outil, d'observer les améliorations des temps de réponses des requêtes XQuery. Enfin, l'outil de génération de cube nécessite l'intégration d'une interface OLAP pour l'interrogation des données du cube. La définition d'opérateurs OLAP et d'algèbre associée à notre modèle d'entrepôt de données complexes étant nécessaire avant de passer à leur implémentation.

entreposage et analyse de données complexes

6 Remerciements

Le développement des différents outils de l'entrepôt XDBLP fut confié à des étudiants de cycle ingéniorat dans le cadre de leurs projets de fin d'étude. Nous tenons à remercier MM. B. Bouraoui et B. Mebirouk (outil ETC), M[elles] N. Sellamna et F. Boukaka (outil de stockage physique) et M[elles] A. Salhi et S. Bencharif (outil de génération des cubes XML).

Références

Baril, X. et Z. Bellahsène (2003). Designing And Managing An XML Warehouse. In *XML Data Management: Native XML and XML-Enabled Database Systems.* : Addison Wesley Professional 455-474.

Boukraâ, D. et O. Boussaid (2007). Complex Data Warehouse Modelling**.** In *Deuxième Atelier des Systèmes Décisionnels (ASD'2007), Sousse, Tunisie.*

Boukraâ, D., R. B. Messaoud, et O. Boussaid (2008). Modeling XML warehouses for complex data: the new issues. In *Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies.* Accepted.

Boussaid, O., R. B. Messaoud, R. Choquet, et S. Anthoard (2006). Conception et construction d'entrepôts XML. In *2ème journée francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA'2006)*, Revue des Nouvelles Technologies de l'Information, Versailles, France. Cépaduès 3–21.

Boussaid, O., F. Bentayeb, J. Darmont et S. Rabaseda (2003). Vers l'entreposage des données complexes : structuration, intégration et analyse. *Ingénierie des Systèmes d'Information (RSTI série ISI) 8*(5-6), 79–107.

Golfarelli, M., S. Rizzi, et B. Vrdoljak (2001). Data Warehouse Design from XML Sources. In *Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001)*, Atlanta, Georgia, USA, pp. 40–47. ACM Press.

Jensen, M. R., T. H., Møller et T. B. Pedersen (2001). Specifying OLAP Cubes on XML Data. *Journal of Intelligent Information Systems* 17(2-3): 255–280.

Li, Y., et A. An (2005). Representing UML Snowflake Diagram from Integrating XML Data Using XML Schema. In *Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC'2005)*, Tokyo, Japan, pp. 103–111. IEEE Computer Society.

Marhoumi. F-Z. (2005). *Entrepôts de données XML: développement d'un outil Extraction Transformation Load (ETL)*. Mémoire d'Ingénieur civil informaticien, Université libre de Bruxelles.

Nassis, V., R. Rajugan, T. S. Dillon, et W. Rahayu (2004). Conceptual Design of XML Document Warehouses. In *Proceedings of the 6th International Conference Data Warehous-ing and Knowledge Discovery (DaWaK 2004)*, Zaragoza, Spain, pp. 1–14. Springer.

Niemi, T., M. Niinimäki, J. Nummenmaa, et P. Thanisch (2002). Constructing an OLAP cube from distributed XML data. In *5th International Workshop on Data Warehousing and OLAP (DOLAP 02), McLean, USA*, pp. 22–27. ACM.

Pokorný, J. (2001). Modelling Stars Using XML. In *Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001)*, Atlanta, Georgia, USA, pp. 24–31. ACM Press.

Rusu, L. I., J.W. Rahayu et D. Taniar (2004). On Building XML DataWarehouses. In *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004)*, Exeter, UK, pp. 293–299. Springer.

Vrdoljak, B., M., Banek et S. Rizzi (2003). Designing Web Warehouses from XML Schemas. In *Proceedings of the 5th International Conference on Data Warehousing and-Knowledge Discovery (DaWaK 2003)*, Prague, Czech Republic, pp. 89–98. Springer.

Tsheke Shele, J. (2007). *Le processus d'Extraction, Tranformation et Load (ETL) dans des entrepôts de données XML*. Mémoire d'ingénieur civil informaticien, Académie universitaire, Wallonie, Bruxelles.

Wang H., J. Li, Z.He et H. Gao. OLAP for XML data (2005). In *Proceedings of the Fifth International Conference on Computer and Information Technology (CIT 2005)*, 21-23, *Shanghai, China. IEEE Computer Society.*

Wiwatwattana N., H. Jagadish V., V.S.Lakshmanan Laks et D. Srivastava (2007). X^3: A Cube Operator for XML OLAP. In *23rd Intl. Conf. on Data Engineering* (*ICDE*), IEEE Computer Society, pp. 916–925.

Xylème (2001). A Dynamic Warehouse for XML Data of the Web. IEEE Data Engineering Bulletin, Vol. 24, No. 2, 2001, pp. 40-47.

Summary

In this paper, we present a set of tools for warehousing and analysing complex data, applied to the bibliography database DBLP. The first one is an ETC tool that allows feeding an XML data warehouse (XDBLP) from an XML data source. The second tool allows the storage of the XML data warehouse data in a native XML database in one hand and in a relational database in the other hand. Also, it allows querying these data in order to measure the query performance for each storage solution. The third tool allows generating XML cubes in order to answer some specific analysis needs.

Conceptual and Logical design for XML Warehouse Methodology and Tool

Zoubir Ouaret*, Ladjel Bellatreche**, Omar boussaid***

* National Institute of Informatics, Algeria
z_ouaret@ini.dz,
** Poitiers University - LISI/ENSMA, France²
bellatreche@ensma.fr
*** ERIC, University of Lyon, France
Omar.Boussaid@univ-lyon2.fr

Abstract. Nowadays, we attend an evolution of information systems and Internet which generates a large number of heterogeneous data sources (structured, semi structured, not structured). XML was largely adopted like the unified language to represent these heterogeneous sources in order to integrate them in a data warehouse. Many XML sources are available on the Web and are used by many applications, such as E-commerce. On the other hand, UML has been used in order to well understand and document business aspects.
In this paper, we propose a conceptual and logical methodology to design XML warehouse using XML and UML, which we call XUML Star. Our methodology starts by Construction of the Global XML Schema of data sources and converting them into a star UML schema using a set of mappings and then expresses them using a star XML schema. To validate our approach, we have implemented a Java based prototype.

1 Introduction

Nowadays, we attend an evolution of information systems and Internet which generates a large number of heterogeneous data sources (structured, semi structured, not structured), e.g., business-to-business (B2B) e-commerce. This situation has led the scientific community to examine the technology for storing and processing information. Since its introduction in 1998, XML has become one of the most influential standards in the industry of information technology. XML (eXtensible Markup Language) is emerging as the unified model to represent heterogeneous and distributed sources. XML offers great flexibility and sufficient power to describe these data sources. These are then stored in XML documents and validated according to a DTD (document type definition) or an XML schema.

The importance of data analysis has grown significantly in recent years as businesses in all sectors have discovered the competitive advantage that just-in-time information can give for the decision-making process. Bill Inmon defined data warehouses as 'subject oriented', integrated, time-varying, non-volatile collections of data that is used primarily in organizational decision making.

However, some application areas (medical, oil, aeronautics) cannot be handled satisfactorily using current multidimensional technology, as the data is too complex and heterogeneous. The eXtensible Markup Language (XML) (W3C, 2000) is now used in major of this application as a common representational and data transmission language, and is rapidly

being adopted as the standard syntax for the interchange of un-structured, semi-structured and structured data. This presents an interesting challenge to explore and study solution and models, for organizing and analyzing the XML documents for business intelligence in the form of XML warehouse repositories and XML marts. In this context, thanks to the flexibility and the power of the UML, all the semantics required for a multidimensional design for XML data can be considered, such as many-to-many relationships between facts and particular dimensions, multiple path hierarchies of dimensions.

Designing a data warehouse for XML sources is an important challenge for the databases community. This approach must use the traditional three level database design approach: (1) The conceptual design, which is to find a suitable scheme (dimension, facts), (2) logical design, star schema, snowflake schema, and so on. (3) Physical design, which is to propose optimization techniques tailored to the XML data (Index and Materialized View Selection for XML, XQuery Rewrite, Fragmentation of XML documents) In this paper, we propose a new methodology for building XML data warehouse based on XML and UML called XUML Star. Three processes are involved in our methodology. The first process converts XML Schemas associated with the XML data into UML class diagrams. We described our transformation rules and the techniques used for implementing them. The second process builds a multidimensional model based on the UML class diagrams; we represent a multidimensional model using a UML star diagram. Finally, the last process represents the UML star diagram using XML Schemas. The reason that we use XML Schemas to represent the multi-dimensional model is for the convenience of multidimensional analysis.

One of the main advantages of our approach is to cover as well as the conceptual level as the logical level by using UML package diagrams which are easy to understand by system designers and users. Secondly, XML Schemas are used in our approach in the opposite of almost previous work that rely on Document Type Definitions (DTDs). In addition, our approach is different from the existing ones that are running at the conceptual level (UML).

The rest of the paper is organized as follows. We first provide a categorization of the work in the area of multidimensional design for XML data by surveying some major academic efforts in the section 2. Then, section 3 presents our proposed methodology for building XML data warehouses. In section 4 we present a first version of our prototype, and finally in section 5 we give the conclusions and Future Work.

2 Related works

Several researches on multidimensional design for XML data were proposed. We briefly present here the different approaches; they can be classified into two categories: (1) storing XML documents, and (2) the sharing and transmission of XML data warehouse. In the first category, the work is to organize the storage of collections of XML documents (mostly textual) in order to optimize the search for information in those documents. In the second category, it is to propose XML models to share and transmit documents. Basically, the modelling process of XML data warehouse can be broken down into two major steps:

1. Conceptual and logical design: conceptual modelling is the determination of the facts, dimensions and hierarchies dimensions. This includes a set of steps that lead to the definition of a logical level (the star schema, the snowflake schema).

2. Physical design: deals mainly three steps: (i) the process of selecting materialized views, and (ii) the process of rewriting queries based on materialized views, and (iii) the selection process indexes. These stages share the same goal: maximizing the performance of queries in data warehouse.

Our work focuses on the storage of XML data for online analytical processing (OLAP). Some limited work has been done in this direction without any of them have been accepted as a complete and valid solution. The industrial environment and the research focus on the subject. In (Nassis, 2004), use Object Oriented (OO) concepts to develop a conceptual model for XML Document Warehouses. They propose a conceptual design formalism to build meaningful XML Document Warehouses (XDW). In this work the authors includes ; (1) conceptually design and build meaningful XML (warehouse) repository (xFACT) using OO concepts in integration with XML Schema constructs, (2) conceptually model and design virtual dimensions using XML conceptual views to satisfy warehouse end-user requirements and (3) use UML package diagrams to help logically group and build hierarchical conceptual views to enhance semantics and expressiveness of the XDW. Working in similar direction, (Trujillo, 2004) have also used UML in an object oriented (OO). They produce a DTD from which valid XML documents are generated to represent multidimensional models of conceptual level. Other works that use XML in data warehouse context (Park, 2005); they proposed a new framework for multidimensional analysis of XML documents, called XML-OLAP. They base XML-OLAP on XML warehouses where every fact data as well as dimension data are stored as XML documents, and build XML cubes from XML warehouses. They propose a new multidimensional expression language for XML cubes, called XML-MDX. The authors in (Baril. 2003) have developed a general architecture for storing XML documents, called DAWAX. The warehouse is defined as a set of materialized views XML(VMIX). The system is based on three main modules: a specification module, a implementation module and a management module. In (Jensen. 2001) the authors propose a scheme for specifying OLAP cubes on XML data. They integrated XML and relational data at the conceptual level based on UML. In their scheme, a UML model is built from XML data and relational data, and the corresponding UML snowflake diagram is then created from the UML model. In particular, they considered how to handle dimensions with hierarchies and ensuring correct aggregation. (Bordawekar. 05) proposed a logical model for XML analysis based on the abstract tree-structured XML representation. In particular, they proposed a categorization of XML data analysis system. This model is valid when the data warehouse is stored in an XML document.

in (Pokorny, 2001), the authors propose a new technique to organize the XML data warehouse using a star model, a different approach is followed, by considering that when the source XML data is gathered from different sources, then each source will provide its particular DTDs. Thus, dimensions are modelled as sequences of logically related DTDs, and the XML-star schema is defined by considering the facts as XML elements. In order to build the dimension hierarchies, this approach defines a sub DTD as the portion of a source DTD that characterizes the structure of a dimension member. The work of (Golfarelli, 2001) presents a semi-automatic approach for building the conceptual schema of a data mart starting from the XML sources. In (Hmmer, 2003), The authors proposed XCube : a family of XML templates used to exchange data cubes over a network. Three different use cases are considered: downloading cubes from a web server, querying cubes residing on a web server, and creating cubes on a web server. The main characteristic on XCube is modularity, which enables schema and data to be separately transmitted. An approach Called X-Warehousing

proposed by (Boussaid, 2006). Entirely based on XML; it designs warehouses with XML Schemas at a logical level, and then feeds them with valid XML documents at a physical level. Further, since it uses XML, this approach can also be considered a real solution for warehousing heterogeneous and complex data in order to prepare them for future OLAP analysis.

The following table summarizes the classification of the works dealing the solutions to the XML Data Warehouse design. by taking into account these following criteria :

1 The three levels of data modelling: they are conceptual, logical, and physical.
2 The type of approach : (***A***):decision approach (data analysis), (***B***):documentary approach (storage of XML documents), or ***(C):*** hybrid (combines the two approaches)

Works		**Levels of data modelling**				
		Conceptual	**Logical**	**Physical**	**Approach**	**Models**
Golfarelli, 2001		×	×		***A***	***Specific***
Pokorný, 2001			×		***A***	
Jensen, 2001		×	×		***A***	***UML***
Neimi, 2002			×		***C***	
Hümmer, 2003			×		***A***	
Vrdoljak, 2003		×	×		***A***	***Specific***
Pederson, 2003			×		***A***	
Park, 2005		×	×	×	***A***	***UML***
Trujillo, 2004		×			***C***	***UML***
Yu li and Aijun 2005		×	×		***C***	***UML***
Baril, 2003			×	×	***B***	
Nassis,2005		×	×		***B***	***UML***
Rusu, 2005			×	×	***B***	
Rajugan, 2005		×			***B***	***MDA, UML***
Boussaid, 06		×	×		***C***	***Specific***
Ravat, 07		×	×		***B***	***Specific***
Our approach		×	×		***C***	***UML***

Table (1): approaches of XML data warehouse design

As we can see, the majority of works start at the logical level, and using mostly DTD, in contrast to our work that we propose a conceptual and logical design, and adapted for storing and data analyzing using XML schema.

3 Methodology

As shown in Fig. 1, in our framework we have divided the design process into three stages as follows: (1) The Conceptual and Logical XML Warehouse. (2) Extraction and Transformation process for loading the transformed XML documents (according to the schema

XMLStar.xsd) into the warehouse. (3) Physical level: Submit the XML documents to populate the designed data warehouse.

In this paper we will focus on the first stage (the dashed box 1, in the Figure). We use XML Schemas and the UML model to extract multidimensional information from diverse XML data sources represented by Global XML schema. By using UML, our multidimensional model is easy to understand for designers and end users; our proposed methodology comprises a set of steps aiming:

- Construction of the Global XML Schema,
- Transformation of the XML Schema to UML diagrams, using our simple transformation rules,
- Converting from an UML model to a MD model (UML star)
- Transformation of the UML Star to XML Star, using the mapping techniques,

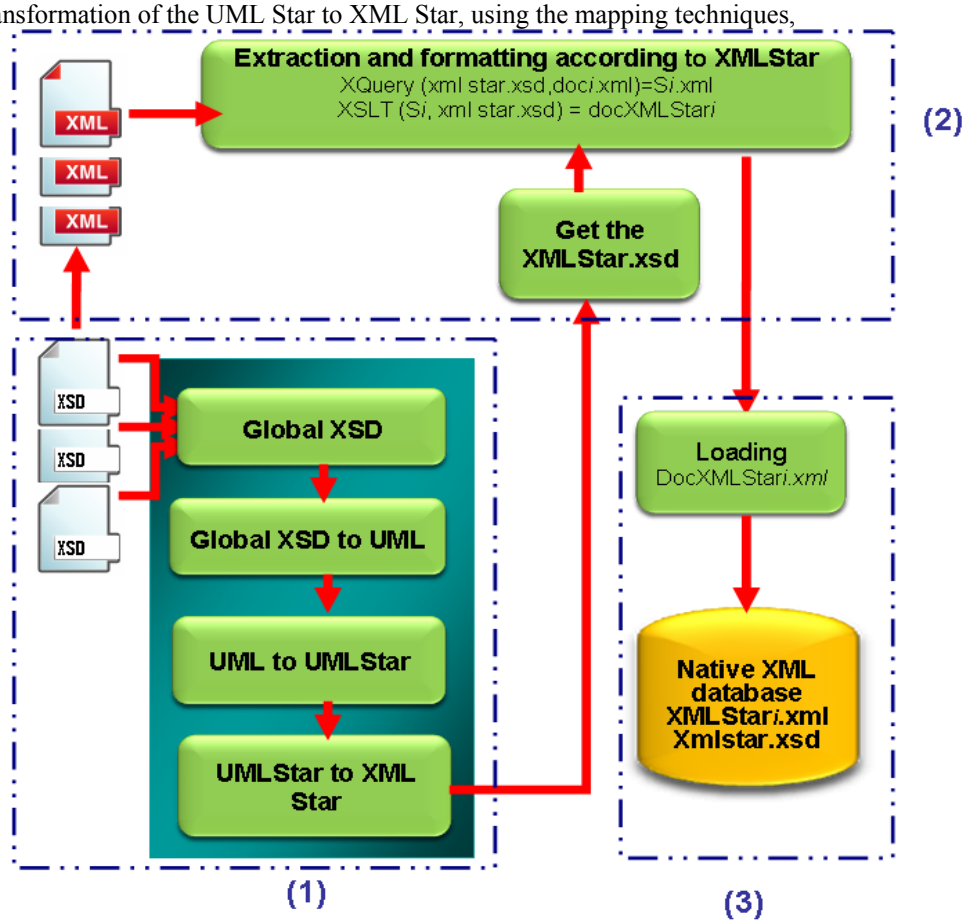


Figure 1: the process of Building XML Data Warehouses: XUMLStar

3.1 Construction of the Global XML Schema:

This step consist to obtain a global XML schema (GXSD), unifying the data sources represented by the corresponding XML Schema XSD_i. In order to construct a global XML

schema, our algorithm consists of an iterative sequence. The first of them consists to the generation of XML schema for data sources provided in input. After XML schema (XSD_i) have been generated, XSD_i and XSD_{i+1} are analysed to remove its inconsistencies and ambiguities, then concatenated for obtaining a global XML schema (union of XML elements of XSD_i and XSD_{i+1}), After all redundant and synonym elements (semantically equivalent) are removed and replaced by the new element < *tag* > receiving their content. Our algorithm consists of the following steps:

- **Transformation:** It allows to, select and transform data sources using XML Schema. The purpose of this step is the translation of data sources to incorporate into a common data model and homogenise their representations expressing them in the XML schema (i.e *XSDi*).
- **Cleaning:** this step eliminates the missing data, incorrect values and the duplicated data. It consists in cleaning every XML schma (*XSDi)* separately.
- **Merging:** In this step, we are seeking to generate from the *XSDi* of the previous phase, a global XML schema *GXSD* that represents all Data sources. The main objective is to have coherent non-redundant elements (removing Duplicates fragments XML). The Construction of such a scheme from a set of XML Schemas (*XSDi)*:

- *Root element:* Create complex XML element that plays the role of union operator and root element of *GXSD*. < *xs* : *element name* = "*::*"*type* = "*::*"= >
< *xs* : *complexT ype name* = "" >
- *Unified name:* Unified name of the two <*elements 1*> and <*elements 2*> can correspond with the name of one of them or can be one their synonyms.
- *Unified type:* correspond to the type of <*elements 1*> (or <*elements 2*>) if they have the same type, otherwise, the selected type is the minimum between the two types of elements or is chosen by the designer.
- *Unified Attributes* < *xs : minOccurs* > *and* < *xs : maxOccurs* >, this represent the cardinality, is defined as the less restrictive cardinality

3.2 XML2UML

The global XML schemas generated is fairly complex to understand the structure of XML data sources. It is difficult to get a quick and comprehensive understanding of an XML vocabulary based on an XML schema. Therefore, it is interesting to represent this vocabulary in UML. UML is the standard modeling language for Object-oriented modelling purposes and is widely adopted as the standard by most business organizations and software companies, which can be easily understood by designers or users. It is very convenient to use UML to represent the semantics of XML schemas favorable for multidimensional analysis.

Several approaches introduce extension of the ER model to design DTDs or XML schemas, other approaches based on UML class diagrams. However, some of these works have no rules for mapping basics of UML concepts such as dependence (or aggregation composition) and generalization. In this paper we extend some works and present our approach (simple rules) to transform XML Schemas to a conceptual model using UML:

- For each element and for each complex type declaration complex, creating a class with a primary key column. The name of this class is that the type of element or complex ;

- simple type attributes and the child elements of a complex elements will be transformed into the attributes of the corresponding UML class ;
- The pre-defined data types such as < *xs* : *int* >, < *xs* : *double* > and < *xs* : *string* > will be converted into their equivalent UML int, double and String ;
- For each attribute with multiple values and each simple element with several occurrences, create a separate class to contain values, the latter will be linked to the parent class through the primary key here ;
- For each complex child element, linking its corresponding UML class to UML class corresponding to the parent through the primary key of the parent class ;
- Attributes < *xs* : *minOccurs* > and < *xs* : *maxOccurs* > worn by the element are transformed into UML cardinalities . The cardinalities may include associations, aggregation, composition and attributes.

3.3 UML 2 UMLStar

The early work on multidimensional and data warehousing concepts date back to works done by W.H. Inmon. Later work by Ralph Kimball's popular Star Schema, The most well-known dimensional model, provided the base for other well known conceptual models such as fact constellations schema and the snowflake schema to be derived.

We propose in this section an algorithm for conversion of UML diagram generated from XML sources into one or more UML Star diagrams. Recently, (Yeol Song and all, 2007) present a method, which semi-automatically generates star schemas from an E/R by analyzing its semantics. Complementary, our approach is based on UML, which is the industry standard notation for software architecture, and is very good in many aspects then E/R, in data warehouse modelling. This section can be broken down into the following steps:

1. Select all the classes with many-to-many relationships in the UML class diagram containing numeric and additive non primary attributes and to designate them as fact class,
2. Select all the classes of the UML class diagram that are not candidates to be facts are candidates to be dimensions. And chose specially those related to a fact by one-to-many relationship,
3. Denormalize all of the remaining classes into fat classes with single-part keys that connect directly to the fact classes. These classes become the dimension classes,

The process of building a UML Star diagram from UML class diagrams that come from multiple XML data sources is semi automatic, requires the intervention of the designer who knows what information should be included in the resultant UML Star diagram.

3.4 UMLStar2XMLStar

In this section, we present the mapping from the conceptual model to the logical model (UML Star to XSDs Star). We purpose a set of rule describing in a detailed and strict way how to convert UML class diagram (such classes, attributes, associations, etc) into physical XML representations.

A some various approaches have been suggested towards mapping UML and XML, and receive increasing attention in the database community in the best of our knowledge. (Carlson,2001) discusses the transformation of UML model to XML schema. (Routledge, 2002) define a mapping between the UML class diagrams and XSDs using the traditional three-level database design approach (i.e. conceptual, logical and physical design levels). However, these techniques neither specify a standard methodology for the mapping nor do they address transformations for some of the primary elements of UML like associations, aggregation, composition, generalization.

3.4.1 Mapping UML class: A class UML is transformed into an XML element (< *xs* : *element* >), and a matching complex type declaration (< *xs* : *complexT ype* >) in XML schema.

3.4.2 Mapping data types UML: The pre-defined data types such as *string, integer, decimal, float , double, date* will be converted into their equivalent in XML schemas by the standard XSD notations, such as, < *xs* : *string* > , < *xs* : *integer* > , < *xs* : *decimal* >, < *xs* : *float* >, < *xs* : *double* >, < *xs* : *date* >. The types of data defined by the user will be marked in UML using UML stereotype << *XSDsimpleT ype* >>.

3.4.3 Mapping attributes UML: In general, an attribute UML is transformed into an XML element contained in the tags < *xs* : *sequence* > < =*xs* :*sequence* > and with a 'type' attribute to indicate the data type or mapped to XML attributes and can be controlled by using UML stereotypes. If the attributes in the UML class are not constrained to any order, < *xs* : *all* > element is used. If the attributes in the UML class are constrained by a specific order, < *xs* : *sequence* > element is used.

3.4.4 Mapping Associations: The two most effective techniques to mapped UML associations in an XML Schema are: (i) Key/Keyref references of elements. (ii) Hierarchical relationship. There are other techniques such as XLink and XPointer. We represent an XML entity equivalent in the both classes concerned in the association, using an XML element with reference attributes. Association definition is described using name="association" "ASS 1 *A B*" is used in class A definition and "ASS 2 *B A*" in class B definition. The direction of an association can be preserved by adding the number that indicates direction to read the association or navigation. If multiple constraints are present in the UML model, they will be represented by XML attributes *minOccurs* and *maxOccurs*. Otherwise, ¤ is used in the both classes linked by the association.

3.4.5 Mapping delegation (or dependency): The parent class and its child class are mapped to separate complex XML element with XML elements, using name="DEPEND" "*DEPEND A B*" respectively, for associating them (as shown in mapping associations). The notation XSD is similar to associations.

3.4.6 Mapping of aggregations: The parent class and its child class are mapped to a single XML schema with two complex XML element uniting attributes of all the UML classes concerned in the aggregations. We can also used, as an association, by XML element declaration in the class, with name attribute = "*AGGR A B*", in class A and ref attribute indicating the referencing class B

3.4.7 Mapping compositions: The existing researches treat composition as a general association. We can also represent the composition by an XML element declaration in the class where the composition originates, with name attribute = "*COMP A B*" in class A and ref attribute for the referencing class B.

3.4.8 Mapping Generalization: The most important characteristic of generalization is the inheritance of attributes of the superclass. For mapping the generalization, the complex type is defined as an extension of the complex type of the superclass element, if a class is a subclass. And other alternative, each and every one element and attributes of the superclasses are assigned to the subclasses.

4 XUML Star Description

This section describes the part of the DTD that defines our proposed approached (XUML Star):

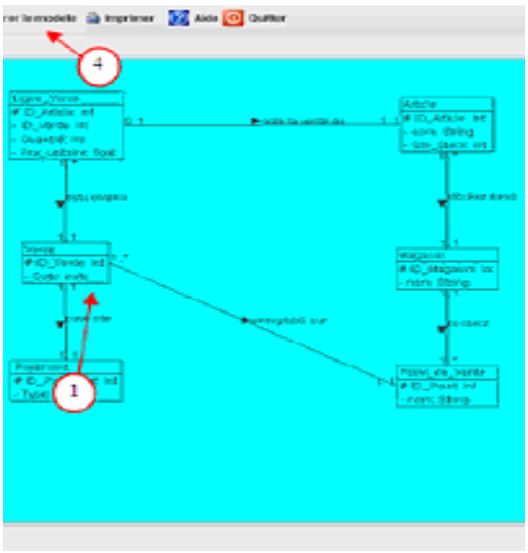
```
<?xml version="1.0" encoding="UTF-8"?>
<!—XML Warehouse element -->
<!ELEMENT XUMLStar (NameDW,Fact,Dimension+)*>
<!ELEMENT NameDW (#PCDATA)>
<!ELEMENT Fact  (NameF,Element+,IDFact,Source*)>
<!ELEMENT NameF (#PCDATA)>
<!ELEMENT IDFact  (ref)+>
<!ELEMENT ref  (IDDim, Dimension)>
<! ATTLIST source
                url CDATA #REQUIRED
>
<! ATTLIST IDDim
                IDDim IDRef CDATA #REQUIRED
>
<!ELEMENT Dimension  (NameD,Element+,IDDim,Source*)>
<!ELEMENT ID Dim CDATA #REQUIRED>
<!ELEMENT NameD (#PCDATA)>
<!ELEMENT Element  (#PCDATA)>
<! ELEMENT source EMPTY >
<! ATTLIST source
                url CDATA #REQUIRED
>
```

5 Implementation

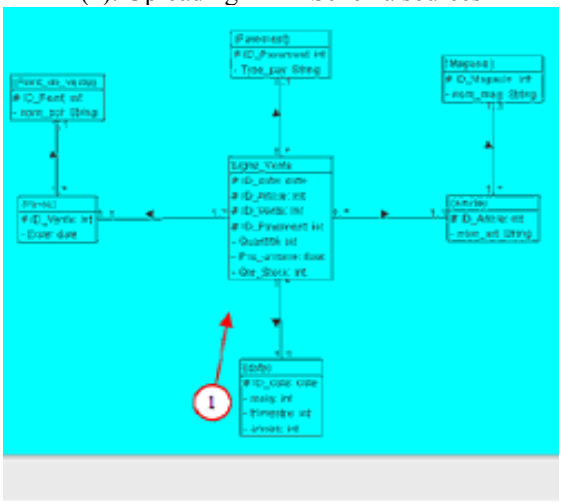
We have developed an application that implement a XUML Star and verify the mapping rules discussed in the previous section. We use the Java programming language. It has a powerful introspection and reflection mechanism. The application can be runs on any Java

enabled computing platforms, such as PCs, Unix, Linux, Mac OS X. The following figure illustrates the four major components and the first step of prototype:

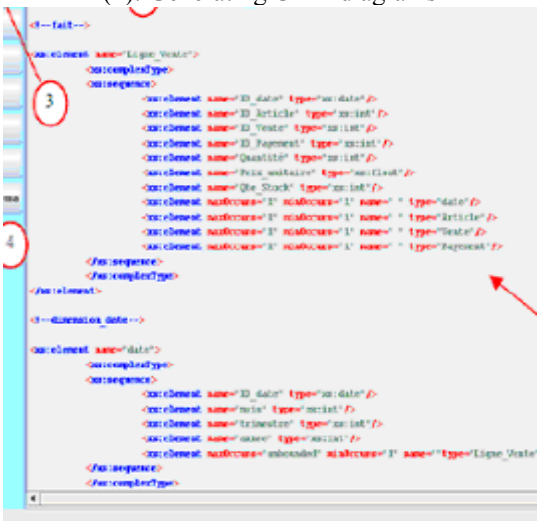
(1): Uploading XML Schema sources



(2): Generating UML diagrams



(3): Generating UML Star from UML diagrams



(4): Generating star XSD

6 Conclusions and Outlook

In this paper, we have proposed a conceptual and logical modelling of XML warehouse based on XML and UML, called XUML Star. We do in XML documents as data sources

useful for the purposes of analysis. We set matches between XML and UML and between UML and XML.

For future work, a lot of issues deserve investigation. We plan to extend this work in multiple ways. The first it would be interesting to extend UML classes to take into account the specific characteristics of XML, such as name spaces, entities parameter, the order and placement of elements and so on. A comparative study of the implementation of our solution in native XML RDBMS and in relational would position well our approach. In order to be complete (physical design), an adaptation of optimization technique proposed in the relational data warehouses (materialized views, indexes join and fragmentation) is recommended. Several other prospects seem to be emerging for our approach XUMLStar, namely the development of matching module for star diagrams, which is two diagrams correspondence classes (the end-users requirement diagram and the diagram generated from XML sources) to produce another star diagram class.

References

Baril, X., Bellahsène, Z. (2003). Designing and Managing an XML Warehouse. XML Data Management : Native XML ans XML-enabled Database Systems. Addison Wessley, pp. 455-473, 2003.

Bordawekar. R and Lang, C. (2005) Analytical Processing of XML Documents: Opportunities and Challenges. In Proceedings of SIGMD, pp. 27–32, 2005.

Bourret (2003). XML et les bases de données : Ronald Bourret Novembre 2003

Boussaïd. O, BenMessaoud, R, Choquet, R, Anthoard S, "X-Warehousing: an XML-Based Approach for Warehousing Complex Data", 10th East-European Conference on Advances in Databases and Information Systems (ADBIS 06),

Gofarelli, M., Rizzi, S., Vrdoljak, B. (2001). Data Warehouse Design from XML Sources, In Proc. The 4th ACM Intl Workshop on Data Warehousing and OLAP (DOLAP01), pp. 40–47, Atlanta, 2001.

Hümmer, W., A. Bauer, and G. Harde (2003). Xcube : Xml for data warehouses. In DOLAP '03 : Proceedings of the 6th ACM international workshop on Data warehousing and OLAP, New York, NY, USA, pp. 33–40. ACM Press.

Jensen, M. R. Mφller, T.H., Pedersen, T.B. (2001). Converting XML Data To UML Diagrams For Conceptual Data Integration, In Proc. The 1st Intl Workshop on Data Integration Over The Web, pp. 17–31, 2001

Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley & Sons.

Li, Y. and A. An (2005). Representing UML Snowflake Diagram from Integrating XML Data Using XML Schema. In Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC '2005), Tokyo, Japan, pp. 103–111. IEEE Computer Society.

Lujan-Mora, S., Trujillo, J., Vassiliadis, P. ((2004). Advantages of UML for multidimensional modeling. In Proceedings of International conference on Enterprise Information Systems (ICEIS'04), pp. 298-305, 2004.

Nassis V., Rajugan, R., Dillon, R., Rahayu, W. (2004). Conceptual Design of XML Document Warehouses, In Proc. Data Warehousing and Knowledge Discovery, 6th International Conference, DaWaK 2004, pp. 1–14, Zaragoza, Spain, 2004.

Park, B. K., H. Han, et I. Y. Song (2005). XML-OLAP : A Multidimensional Analysis-Framework for XMLWarehouses. In Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2005), Copenhagen, Denmark, pp. 32–42. Springer.

Pedersen, D. et T. B. Pedersen (2003). Achieving Adaptivity for OLAP-XML Federations.In *Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP(DOLAP 2003)*, New Orleans, Louisiana, USA, pp. 25–32. ACM Press.

Pokorný, J. (2001). Modelling Stars Using XML. In Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001), Atlanta, Georgia, USA, pp. 24–31. ACM Press.

Rajugan, R.; Chang, E.; Dillon, T.S. Conceptual design of an XML FACT repository for dispersed XML document warehouses & XML marts , Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on Volume , Issue , 21-23 Sept. 2005 Page(s): 141 – 147

Ravat, F. Teste, O. Tournier, R. Zurfluh, G: A Conceptual Model for Multidimensional Analysis of Documents. ER 2007: 550-565

Rusu, L. I., J. W. Rahayu, et D. Taniar (2005). A Methodology for Building XML Data Warehouses. International Journal of Data Warehousing and Mining, Idea Group Inc. 1(2), 67–92.

Trujillo, J., S. Lujàn-Mora, et I. Song (2004). Applying UML and XML for Designing and Interchanging Information for DataWarehouses and OLAP Applications. *Journal of Database Management 15*(1), 41–72.

Il-Yeol Song, Ritu khare, Bing Dai: SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram. DOLAP 2007: 9-16

Résumé. De nos jours, nous assistons à une évolution des systèmes d'information et de l'Internet qui ne cesse d'engendrer des flux de données de plus en plus hétérogènes (structurées, semi structurées, non structurées). XML a été largement adopté comme le langage unifié permettant de représenter ces sources hétérogènes afin de les intégrer dans un entrepôt de données. De nombreuses sources XML sont disponibles sur le Web pour les applications de commerce électronique « business to business », etc. D'autre part, UML a été utilisé afin de mieux comprendre et documenter ces aspects business. Nous proposons dans cet article, une méthode de conception d'un entrepôt de données XML, basée sur XML et UML, que nous appelons XUML Star. Le schéma XML global représentant les différentes sources de données est convertis en un diagramme UML en étoile puis exprimés par un schéma XML en étoile.

Supporting Virtual Group Decision Meeting

Abdelkader Adla[*,**], Mohamed Frendi[*]

*IRIT – Paul Sabatier University, 31062 Toulouse, France
Abdelkader.adla@irit.fr
**Computer Science Department, University of Oran, Algeria
Frendi.mohamed@univ-oran.dz

Abstract. Most meetings are perceived to be extremely unproductive in terms of efficiently utilizing the participants' time and effectively achieving the meeting objectives. Indeed, meetings consume a great deal of time and effort in organizations. These problems occur frequently because effective guidelines or procedures are not used. To overcome these problems, we propose in this paper a framework for distributed facilitation incorporating a model of the decision making processes. In this framework many group facilitation tasks are automated, at least partially to increase the ability of the facilitator to monitor and control the meeting process.

1 Introduction

Group Decision Support Systems (GDSS) are a widely used collaborative technology that has proven to increase user participation in and the quality of decision-making. They are intended to provide computational support to collaborative decision-making processes (Desanctis et Gallupe, 1987). In the virtual organization, GDSS seem extremely adequate to improve strategic decisions made at the upper levels of the organizational structures, through better information acquisition, perception of different perspectives and options, and consensus formation. This thread leads to an increasing presence of GDSS in organization, and facilitation activities must accompany such movement, augmenting the interest of the facilitator.

In this regard, the performance of groups interacting with GDSS has been the subject of numerous studies (De Vreede et al., 2002; Kwok et al., 2003; Yoong et Gallupe, 2001). Among them, a few studies have focused on the method used to interact with the GDSS, with emphasis on the use of facilitators (Wheeler et Valacich, 1996; Khalifa et al., 2002). Indeed, Research on facilitation in that field is still sparse, and relatively little attention has been given to support for group facilitation (Niederman et Vokema, 1999; Wong and Aiken, 2003). GDSS does not address areas of group functioning, such as meeting design or managing verbal communications. These and other facilitation activities must come from people. An integration of good computer tools with effective human facilitation can lead to a more effective meeting than either by itself. A significant question is how to effectively plan, coordinate, and direct – to "facilitate" – the work of group members who are using a GDSS.

With the recent advances in GDSS, many group facilitation tasks can be automated, at least partially to increase the bandwidth of group communication and the ability of the facilitator to monitor and control the meeting process. An effective system would reduce the need for developing technical competence, making any novice an effective facilitator in

aiding the group in the collaborative process. Hence an automated process to aid even the most inexperienced facilitator must include tools to monitor group and individual behaviours, indicators to know when to offer or integrate information, as well as know when to employ particular techniques to move the group towards congruence.

To this end, we consider the support to inexperienced facilitators by incorporating a model of the decision making process. The selected model provides a detailed view of decision making process. Having a model of the decision making process built into the system should enable intelligent decisional guidance. It enables the facilitator to appropriately choose and use the framework's tools and techniques in the group decision-making processes, to monitor group's behaviour, and to provide cues and customized explanations accordingly.

The remaining part of the paper is organized as follows. First, we present the distributed facilitation concept, followed by an overview of facilitation systems used in group decision making. Next, we present our framework for distributed facilitation. Finally, we present an example of scenario.

2 Distributed facilitation

2.1 Distributed meeting

Distributed meetings are referred to as virtual meetings because these are frequently attended by participants who are physically separate and rely on networked computers to interact with one another although teleconferencing facilities might be used simultaneously to provide a virtual view of one another. There are many tasks that can be accomplished in a distributed meeting: creating a strategic plan, solving a problem, making a decision, sharing information, resolving a dispute, negotiating a contract, and so on.

Facilitating distributed meeting is to manage the meetings to improve the quality of outputs (McQaid et al., 2000; Briggs et al., 2001). Distributed facilitation techniques are a necessity because many of the conventional techniques of group facilitation, essentially the face-to-face facilitation techniques, are no longer effective in a distributed environment and are deeply based on techniques that require line of sight between the facilitator and the meeting participants. Real world lessons about virtual meetings have been found that facilitating a virtual meeting is difficult because there is communication barrier over a distance, thus making the meeting environment difficult to manage for users and the facilitator (Nunamaker et al., 1997).

In distributed environments, facilitators must rely on computer-mediated communication to intervene in the group, which depending on media richness, requires additional effort and reduces the range of possible interventions. Furthermore, the quality of distributed meetings is difficult to control because the meeting participants have problems of bounding with other participants and may lead to low level of interest and energy during the meetings. As a result, distributed meetings are difficult to facilitate manually (Romano et al., 1999).

2.2 Definition of distributed facilitation

Group facilitation is a process in which a person who is acceptable to all members of the group intervenes to help improving the way it identifies and solves problems, and makes

decision (Schwarz, 1994). Facilitation, on the other hand, is a dynamic process that involves managing relationships between people, tasks, and technology, as well as structuring tasks and contributing to the effective accomplishment of the meeting's outcomes.

Ackermann (1994) found facilitation helped groups to contribute freely to the discussion, to concentrate on the task, to sustain interest and motivation to solve the problem, to review progress and to address complicated issues rather than ignore them.

A further task of facilitation is to engage the group in creativity and problem formulation techniques to help the group bring structure to the issues facing them. Facilitators attend to the process of decision making, while the decision makers concentrate on the issues themselves.

2.3 Human facilitation vs. Automated facilitation

Human facilitation has been identified as a group of activities that the facilitator carries out before, during, and after a meeting in order to help a group during the decision making process (Bostrom et al., 1993). Previous studies found that group performance is improved when individuals participate in the facilitated discussion and receive cognitive feedback (Reagan-Cirincione, 1994). From a virtual organization perspective, the facilitator is in a critical position monitoring efficiency, quality and commitment to solutions, and reporting results to the organization. In this sense, he is considered as the most crucial element of a GDSS (Nunamaker et al., 1997).

Automating facilitation tasks is not a new concept. In fact, previous studies have already shown that automating the technical, repeatable processes of facilitation is considered a positive experience for the participants as well as the facilitator. Automated facilitation is the enrichment of a GDSS with cues that guide decision makers towards successful structuring and execution of the decision making process (Liamayem et DeSanctis, 2000). Like human facilitation, automated facilitation consists of providing the group with decisional guidance in order to help them achieve their own outcomes. However, the tasks that have been automated generally represent the more routine, repeatable tasks that do not necessarily impact or indicate behaviours, and do not support means to develop the skills of inexperienced facilitators.

3 Facilitating group decision making

Distributed meetings are frequently attended by participants who are physically separate and rely on networked computers to interact with one another and with a facilitator. Facilitating distributed group enables to improve the quality of outputs (McQaid et al., 2000; Briggs et al., 2001).

There have been numerous studies in the literature about distributed facilitation during the past decade:

The ESP system (Expert System Planer) (Wong and Aiken, 2003), uses an expert system approach to help facilitators preparing an agenda and selecting GDSS tools. ESP addresses three main concerns: determining the appropriate participants for the meeting, scheduling a calendar for the meeting, and identifying which GDSS tools may be most adequate to tackle the problem. ESP does not recommend any decision process, which classifies this functionality as technology facilitation. One negative characteristic of ESP is that it produces

opaque recommendations, which do not allow facilitators to interpret the decisions made by the system.

COPE (Ackermann et al., 1994) is a system that supports content and strategy formulation by multiple groups along time. Furthermore, the system uses various computational techniques to cluster concepts into manageable parts and identify most central concepts, which supports development facilitation.

Antunes and Ho (1999) present a view of meeting facilitation that blends together the different classifications. The view is strictly focussed on the pre-meeting phase.

Unfortunately, there are not many examples of more advanced systems. The above systems do not support means to develop the skills of inexperienced facilitators. They are mostly beneficial for expert facilitators and none of them support the notion of decision process model. Several authors (Niederman et al., 1999) suggested an expert system approach capable to develop facilitation skills. These expert systems would include the recognition and interpretation of patterns of activity and possible facilitator's interventions.

4 A Framework for distributed facilitation

The goal of the framework is to develop skills necessary to design and conduct an effective and productive facilitated distributed meeting. The framework tools are integrated as embedded facilitation, illuminating the effect of the intelligent management tools reducing, but not eliminating intervention from the facilitator. To reach this goal, we explore how to model the group facilitation process and to manage the monitoring and control activities among human and software tools. Our approach to the development of facilitation skills considers the support to inexperienced facilitators by incorporating a model of the decision making processes. The first step of our design consists then in the selection of a model describing the group decision process. Although many rational models that we have seen related to GDSS could have been used, we adopted an approach for supporting the facilitation in GDSS stemmed from our analysis and observations of the previous models.

According to this model, the facilitator involvement in group meeting is conceptually divided into a three phase process: Pre-meeting, During meeting and Post meeting.

To support the selected decision process model, we propose the distributed software architecture depicted in (FIG. 1) (Adla et al., 2007). The architecture is, in essence, decentralized in terms of databases, model bases and knowledge engines, should make the best use of the available knowledge to offer optimally cooperation capabilities for the participants, and is designed to facilitate collaboration and communication among decision making group through mediated man-man cooperation allowing the group of decision makers and the facilitator to make collective decision. This kind of cooperation uses a machine as an intermediate communication medium.

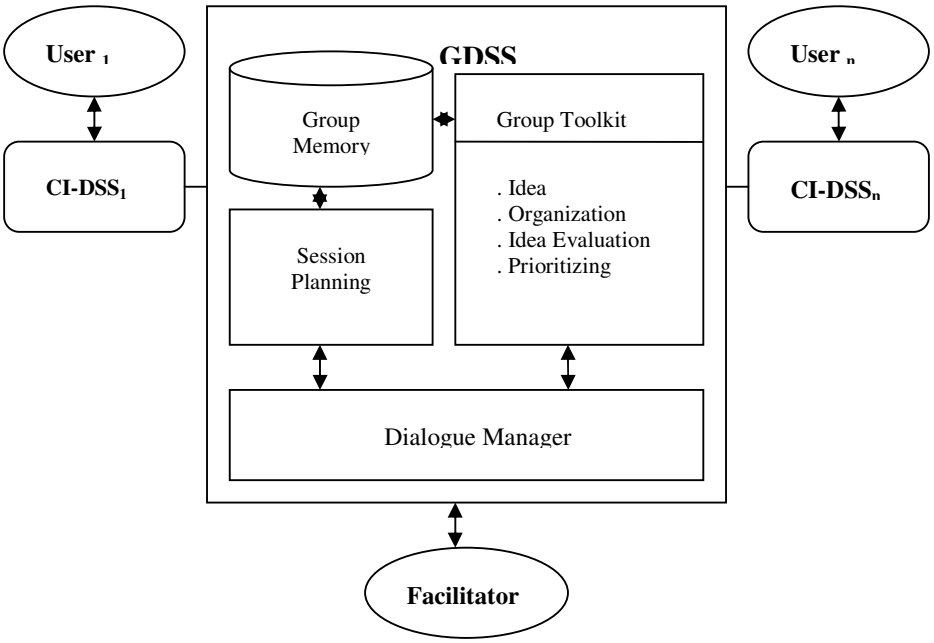


FIG. 1 : Distributed GDSS Architecture

The software architecture is composed of the following modules:

Dialogue Manager. The dialogue manager allow the facilitator and the participants to interact across a like type client-server network which may be web-based to allow distant decision makers to participate.

Group Memory. A group memory is used to implement the meeting repository storing all the meeting related information including meeting setup information, the trace of previous sessions, and intermediate results. It is essential to be able to capitalize knowledge of the decision-makers implicated in the distributed decision processes so that each can refer to it if necessary. Moreover, the decision-makers implicated in distributed and decision processes are supported by this tool by reusing existent resolutions for instance or simply parties of already established resolutions.

Session Planning. A session planning is made at facilitator disposal to set up and manage a meeting. The facilitator defines all the details related to planning decision-making processes. These include naming and describing agenda topics, scheduling, creation of participant lists, notification through e-mail, and definition of issues, expected outcomes. This function is the most important activity supported, since it specifies the sequence of tasks to be conducted during the actual meeting.

Group Toolkit. A set of tools to support group activities that can be classified into three major categories: (1) Idea generation: each decision maker tries to generate alternatives using his CI-DSS that integrate a local expert knowledge. An electronic brainstorming tool may also be used; (2) Idea organization: the facilitator uses tools to organize the ideas transmitted by participants (e.g. remove redundant alternatives); (3) Idea evaluation tools: a set of tools

are made at the group disposal to rate, rank, multi-criteria evaluate the alternatives before choosing a solution.

Each group tool has two versions: (a) participation version as private screen; it is used by a meeting participant engaging in a meeting activity; (b) Facilitation version as public screen; it is used by a meeting facilitator to set up parameters or data items associated with a meeting activity.

The Cooperative Intelligent DSS (CI-DSS). In the proposed system each networked decision maker is supported by a cooperative intelligent DSS (CI-DSS) (Adla et Zaraté, 2006]. The CI-DSS enables a decision-maker reaching decisions by combining personal judgements with information provided by the embedded tools. Within this system, a decision-maker undertakes environment assessment and strategic analysis, and provides data, judgements, intuition and personal vision as inputs to the system. An intelligent reasoning process is performed by the system to generate alternatives. Decision-makers review their overall viability and make suggestions.

The Group facilitation Support System. The selected decision process model provides a detailed view of decision making processes (FIG. 2). We should emphasise that having a model of the decision making process built into the system enables decisional guidance. It enables the facilitator to appropriately choose and use the system's functional capabilities in the group decision-making processes. An intelligent guidance system monitors group's behaviour and provides cues and customized explanations accordingly. The particular facilitation techniques the facilitator focuses on at various times depend on the particular stage of the meeting process.

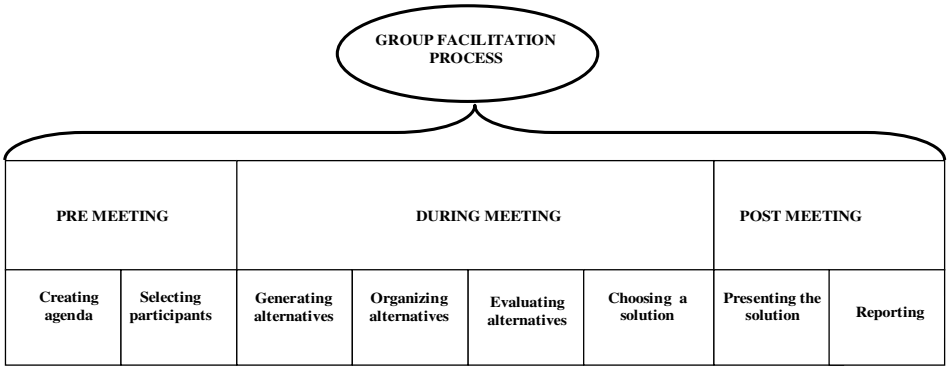


FIG. 2 – Group Facilitation Process

5 Example of application

Gas Liquefying Zone (GLZ) is a plant specialised in liquefying gas. It is one of several plants (a dozen on a national scale) which compose a parent oil company. The management system of the boiler combustion is one of the most critical systems for the good functioning of the plant. It has a high impact on the methods of cogitation and apprehension of various problems related to maintenance. To supervise the good functioning of the boilers, different sensors are set up to point out anomalies at different stages of the process. The exploiting staff is often confronted with situations that impose a quick reaction of decision-making. This requires consequent material resources and highly trained and experienced operators in oil process management.

Currently, in a contingency situation (i.e. breakdown of a boiler) it is the duty of the process administrator at a local site to identify and diagnose the breakdown. The handling of a boiler breakdown consists of three main steps: discerning defects while the boiler is functioning, diagnosing defects, and proposing one or several appropriate actions of repair. There are two types of breakdown that may occur:

1. Automatically signposted to the operator by means of a triggered-off alarm, the flag (the reference given to each alarm) is pointed out on the board (control room). It acquaints with a particular alarm.
2. Intercepted by the operator (case of defectiveness of the sensor where no alarm is triggered off but the boiler does not work), the operator explores a large research space of potential defects.

For the process administrator, there is two ways to solve the problem. In the former, he refers manually to several textbooks provided by the manufacturer to seek for the causes and determine the actions of repair. In this case, the use of the specific CI-DSS enables the operator to diagnose the problem and to carry out the actions of repair. In the latter, he performs a set of tests on the site. He may also recall from his knowledge a previously encountered and successfully resolved case with similar characteristics and signs. If no solution can be found locally, the operator informs the process administrator who makes contact with other process administrators and/or operators of the parent company and even calls on the technical services of the boilers manufacturer located abroad via traditionally communication media (phone call, fax, etc.). A lot of overhead might occur when the various actors try to exchange their knowledge and ideas which could undermine efficiency – when it is needed the most. The current set up of team work keeps the actors too distant from the actual site as often it is not possible to meet. Therefore, it is the process administrator who plays the role of a facilitator and attempts to interact with all the actors. Consequently, this situation is mainly characterized by highly decentralised data sets coming from various sources. There is constant need to access this decentralised information at any time, from anywhere, under tight time constraints. We will show how our distributed facilitation framework may provide answers to this challenge.

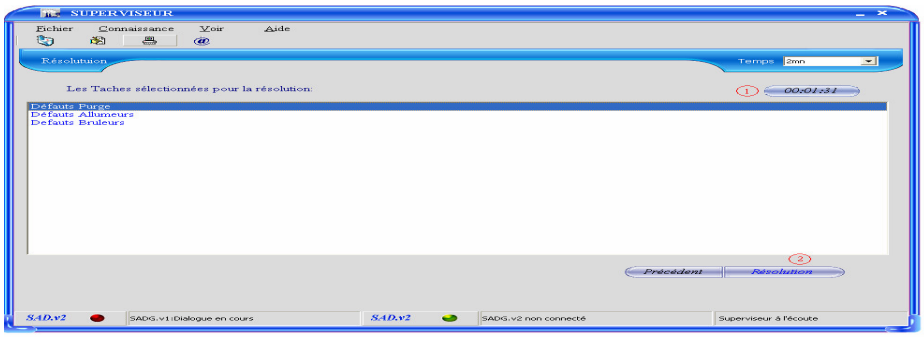
An on-line “meeting” is used to represent a group decision making for the specific problem at hand. The decision-makers (participants) are located in different places. A computer network is presumed that connects these different locations of participants. The decision-making process is controlled by a process administrator (facilitator). Various activities are needed to perform the meeting outcomes. The system provides the support for the facilitator to manage the group and achieve the meeting outcomes. The decision making process consists of the following three phases:

Pre-meeting. In this phase, the facilitator achieves three activities:

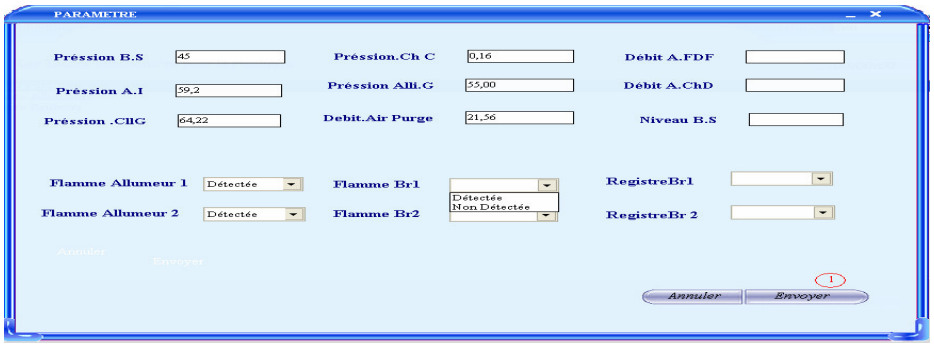
Agenda creation: The facilitator plans the decision making process and defines an agenda of the tasks and activities to be undertaken**.** It is the facilitator responsibility to specify into the session planning the agenda, ordering the items and establishing deadlines for each of them. Furthermore, the facilitator states the objectives of the meeting to be held within the deadline.

Participant selection: using the contract-net protocol, the facilitator selects the decision-makers (the process administrators and/or operators of the other plants of the parent company and technical services of the boiler manufacturer) to participate, and defines the ground rules for the process and time limits. A list of potential participants (a participant rooster) is made at the facilitator disposal. Participants have unique names within the environment. This name is a composite of the natural name and network address of a participant. The model requires the uniqueness of names of participants.

Problem submission and presentation: The facilitator introduces the participants, provides meeting goals/purpose, presents the problem, states the causes/effects, the impact and the extent of problem, and makes clear roles for the participants and clear rules for the process.



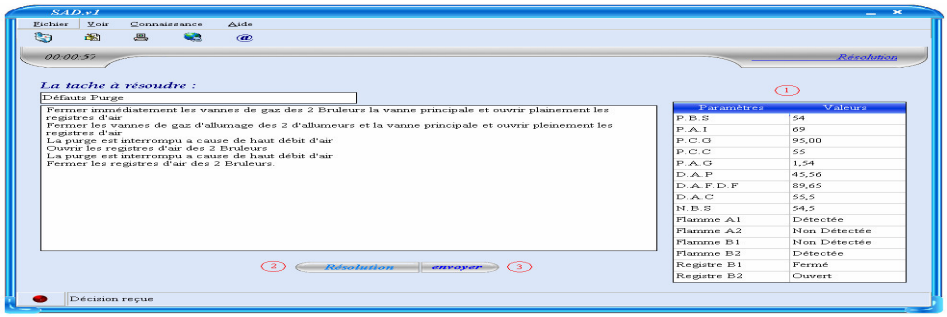
Problem submission



Default Signs and boiler parameters

During meeting. The problem resolution is achieved at two levels: the decision maker level (alternative generation) and the group level (alternative organization and evaluation, and solution choice):

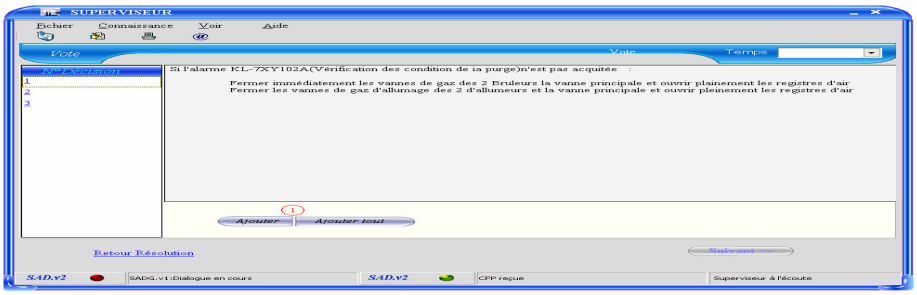
Generating alternatives: To create the solution alternatives to the problem at hand, each participant (decision-maker) uses his proper *specific Cooperative Intelligent Decision Support System (CI-DSS)*. Different methods are envisaged to achieve a task. The system chooses a method dynamically to achieve it. In order to do that, given the name of the task to be solve (wording of problem), the system constructs an action plan to be carried out (a sub-graph of tasks-methods hierarchy). The problem solving mechanism is based on a series of cycles until the entire problem is solved. Each cycle consists of the following steps: (1) identifying candidate methods; (2) Identify triggered methods; (3) Selecting a method; (4) assigning the method to an agent (system or user); (5) Executing the method; and (6) evaluating the task state.



Problem solving by a participant

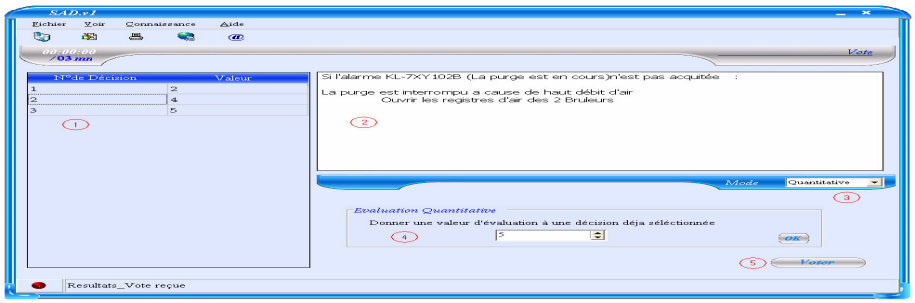
The issued alternatives are then put in the private space. Each participant can select some of his private alternatives to be exposed to the group. Participants have a delay for private creation of alternatives. Afterwards, the facilitator makes public the alternative proposals on the shared (public) space to the group.

Organizing alternatives: Using the system, the group generates in a short period of time a huge amount of meaningful contributions (alternatives) that need to be organized and synthesized. The similar or duplicated alternatives are eliminated or merged. Idea organization in a distributed environment is mainly the facilitator's responsibility.

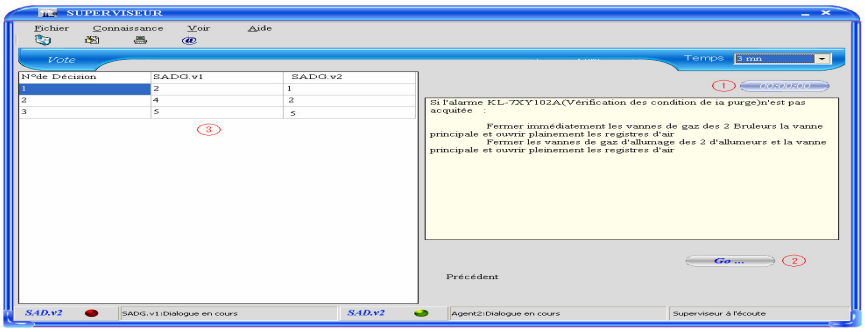


Organizing the alternatives received from the decision-makers

Evaluating alternatives: Participants submit their evaluations or votes. They also view group results that include group averages and standard deviations. A large standard deviation may indicate a lack of consensus on an alternative or issue. The facilitator brings issues with large standard deviations to participant's attention for further discussion. The participants recast their votes to see whether the team can come to a consensus. Four evaluation tools are developed: Rating, Ranking, selection, and multi-criteria evaluation tools.



Evaluating the alternatives by a participant according the voting tool



Group evaluation by the facilitator

Decision choice: At this stage one alternative is chosen according to the evaluation tool used. This decision constitutes the collective decision made by the group.

Post-meeting. At the time of the closure of a meeting, a number of questions, options and arguments have been generated. Participants are supposed to be aware of the contents of the meeting, but a remainder mechanism is necessary, particularly when the number of items is high. To help participants to access the elements during the meeting, summary and cross-reference reports are made available by the system.

6 Conclusion

We considered in this paper the support to inexperienced facilitators by incorporating a model of the decision making process. The selected model provides a detailed view of decision making process. Having a model of the decision making process built into the system should enable intelligent decisional guidance. It enables the facilitator to appropriately choose and use the system's functional capabilities in the group decision-making processes, to monitor group's behaviour, and to provide cues and customized explanations accordingly. Thus, the facilitator uses this framework to help select appropriate GDSS tools or techniques to be used in the meeting. The particular facilitation techniques the facilitator focuses on at various times depend on the particular stage of the meeting process.

One must conclude that facilitating group decision meeting is one critical role with impact on meeting outcomes, which raises the question of how do facilitators perform that task. We believe that by providing facilitation, we can alleviate increased cognitive facilitator and practitioner load. We can also simplify the facilitation process, so that even inexperienced team members can facilitate group processes thereby increasing the scalability of the use of GSS in virtual environments.

Références

Ackermann, F. and C. Eden (1994). Issues in computer and non-computer supported GDSSs. Decision Support Systems, 12, 381-390.

Adla, A. and P. Zaraté (2006): "A Cooperative Intelligent Decision Support System", IEEE International Conference on Service Systems and Service Management (ICSSSSM'06), October 25-27, Troyes (France).

Adla, A., J.L. Soubie and P. Zaraté. (2007): "A cooperative Intelligent Decision Support System for Boilers Combustion Management based on a Distributed Architecture". *Journal of Decision Systems (JDS),* 16/2, 241-263.

Antunes, P. and T. Ho (1999). Facilitation tool – A tool to assist facilitators managing group decision support systems. Proceedings of the 9th Annual Worshop on Information technologies and systems. Charlotte, NC, USA, 11-12, 1999. pp. 87-92

Bostrom, R.P., R. Anson and V.K. Clawson (1993). Group facilitation and group support systems. *In L. Jessup and J. Valacich (eds.), group support systems: new perspectives. New york: Macmillan*, 146-168.

Briggs, R.O., G.J. De Vreede, J.F. Nunamaker and D.H. Tobey (2001). ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS), Proceedings of the 34th HICSS.

De Vreede, G.J., J. Boonstra and F. Niederman, (2002). What is effective GSS facilitation? A qualitative inquiry into participants' perceptions, proceedings of HICSS-35, big Island, Hawaii, 7-10 January.

Desanctis, A.R., and R.B. Gallupe (1987). A Foundation for the study of group decision support systems. *Management Science*, 33/5, 589-609.

Khalifa, M., R. Davidson and R.C.W. Kwork (2002). The effects of process an content facilitation restrictiveness on GSS-mediated collaborative learning. *Group decision and negotiations*, 11/5, 345-361.

Kwok, R.C.W., Ma, J and Vogel, D.R. (2003). Effects of Group Support Systems and content facilitation on knowledge acquisition. *Journal of management Information systems*, 19/3, 185-230.

Liamayem, M. and G. DeSanctis (2000). Providing decisional guidance for multicriteria decision making in groups. *Information systems research*, 11/4, 386-401.

McQaid, M.J., Briggs, R.O., Gillman, D. Hauck, R. (2000). Tools for distributed facilitation. Proceedings of the 33rd HICSS, 4-7 January 2000.

Niederman, F. and R.J. Volkema (1999). The effects of facilitator characteristics on meeting preparation, set-up, and implementation. *Small group research*, 30/3, 330-360.

Nunamaker, J., R. Briggs, D. Mittleman, D. Vogel and P. Balthazard (1997). Lessons from a dozen years of group support systems research: a discussion of lab and field findings. *Journal of Management Information Systems*, 13/3, 163-207.

Reagan-Cirincione, P. (1994). Improving the accuracy of group facilitation social judgement analysis and information technology, organizational behaviour and human decision processes*,* 58/2, 246-270.

Romano N.C., J.F. Nunamker, R.O. Briggs and D.D. Mittleman (1999). Distributed GSS facilitation and participation: field action research. Proceedings of the 32th HICSS.

Schwarz, R. (1994). *The skilled facilitator*. Jossey-Bass Publishers.

Wheeler, B.C. and J.S. Valacich, (1996). Facilitation, GSS, and training as sources of process restrictiveness and guidance for structured group decision making: an empirical assessment. *Information systems research*, 7/4, 429-450.

Wong, Z. and M. Aiken (2003). Automated facilitation of electronic meetings. *Information & management* , 41/2, 125-134..

Yoong, P. and R.B. Gallupe (2001). Action learning and groupware technologies: a case study in GSS facilitation research. *Journal of information technology and people*, 14/4, 625-644.

Résumé

La plupart des réunions consomment beaucoup de temps et d'effort dans les organisations. Pour surmonter ces problèmes, nous proposons un cadre pour la facilitation distribuée intégrant un modèle de processus de prise de décision. Dans ce cadre, de nombreuses tâches d'animation de groupe sont automatisées, au moins en partie, afin d'accroître la capacité de l'animateur à surveiller et à contrôler le processus de réunion.

Réseau spatial flou de voronoï
Un support d'aide à la décision pour la planification urbaine

Aziz Mabrouk, Azedine Boulmakoul

Faculté des Sciences et Techniques de Mohammedia
FSTM – Département informatique - B.P. 146 Mohammedia Maroc
aziz.mabrouk@yahoo.fr
azedine.boulmakoul@yahoo.fr

Résumé. Les réseaux spatiaux constituent des supports essentiels de l'organisation spatiale du territoire, l'accessibilité joue un rôle clé dans la dynamique du transport et de l'urbanisme. Elle est dépendante de la qualité des réseaux de transport et de la structure des lieux; elle influence nos déplacements quotidiens ainsi que nos décisions de mobilité. Dans ce papier nous adoptons le diagramme de voronoï flou d'un réseau spatial (DVRS-Flou). Cette approche basée sur la logique floue et la théorie des graphes pour évaluer l'accessibilité géographique de voronoï flou à ses générateurs (station, service public, équipement,…). Cette évaluation ainsi que sa représentation cartographique occupent une position clé parmi les outils nécessaires qui procurent aux décideurs des véritables supports d'aide à la prise de décision dans le domaine des transports et de la planification urbaine. En effet, nous proposons dans ce cadre de travail deux algorithmes qui permettent de calculer le diagramme de voronoï flou d'un réseau spatial et qui consistent à assigner chaque générateur de Voronoï à ses plus proches nœuds du graphe modélisant un réseau spatial physique. Cette assignation des nœuds est basée sur la comparaison des poids des plus courts chemins parcourus et qui sont modélisés par des nombres flous représentant de façon fidèle la réalité, des durés, des vitesses ou de toutes valeurs incertaines qui permettent de décrire le déplacement.

1 Introduction

Aujourd'hui, nous remarquons un intérêt croissant envers les systèmes d'aide à la décision, tant en planification de transport que dans d'autres domaines. Cet intérêt provient non seulement de la motivation à développer de nouveaux outils SIG (Densham et Goodchild, 1989), mais également d'un besoin stratégique d'outils d'aide à la décision basés sur une infrastructure d'informations spatiales (Chen et McLaughlin, 1992), (Herrington, et al., 1991), (Armstrong, et al., 1991), (Worall, 1990).

L'accessibilité est souvent utilisée pour identifier les zones dont l'infrastructure de transport nécessite des améliorations ou pour évaluer l'impact de différentes politiques d'investissement dans ce domaine. Egalement, elle peut être utilisée pour évaluer la qualité d'un système de transport et de préciser les régions susceptibles de bénéficier d'un projet de transport donné.

Plusieurs algorithmes ont été proposés dans la littérature (Erwig, 2000), (Takehiro et al. 2005), (Margot Graf et Stephan Winter, 2003), (Mabrouk et Boulmakoul, 2008) et qui permettent de calculer le diagramme de voronoï de type réseau dont les poids des chemins parcourus sont représentés par des nombres réels. Or, cette estimation ne reflète pas parfaitement la réalité. En se basant sur la logique flou, nous étendons ces algorithmes afin d'évaluer la qualité des réseaux spatiaux avec des données incertaines et/ou imprécises. Le diagramme de voronoï réseau flou (DVR-Flou) permet alors de calculer les poids flous des plus courts chemins entre les nœuds du graphe modélisant un réseau spatial et leurs plus proches générateurs de voronoï (écoles, hôtels, hôpitaux, services publics, etc.…). Ceci permet d'évaluer de façon plus fidèle *l'accessibilité géographique de voronoï*. En effet, cet indicateur complexe basé sur l'information spatial floue permet de mieux appréhender le territoire afin de prendre les meilleures décisions.

Dans ce papier, une modélisation du réseau spatial réel avec un graphe flou sera présentée dans la deuxième section. Dans la troisième section, nous présentons les méthodes utilisées pour la génération des points d'accès qui permettent d'une part, d'accéder à tout service ou équipement (points d'origine) existant le long du réseau spatial et d'autre part, de générer le diagramme de voronoï flou du réseau spatial. Une définition de ce dernier sera donnée dans la quatrième partie où nous présentons également l'arithmétique des nombres flous. Dans la même section, nous proposons deux algorithmes permettant de calculer le diagramme de voronoï flou. Dans la cinquième section et avant de conclure, une nouvelle définition de l'accessibilité géographique basée sur la logique floue sera proposée en justifiant son apport pour la prise de la décision.

2 Graphe flou modélisant un réseau spatial

Un réseau spatial est un objet géographique implanté sur le territoire. Cette implantation spatiale est traduite par le géo-référencement des sommets et par la valuation des arêtes. Cette valuation représente une duré, une vitesse ou toute autre valeur réelle qui permet de décrire le déplacement entre deux emplacements géographiques. Cependant, cette valeur dans un réseau spatial réel est incertaine et doit être représentée de façon plus fidèle à la réalité. Ceci implique qu'un réseau spatial peut être modélisé par un graphe flou dont les sommets sont associés aux infrastructures nodales et les arêtes sont associées aux infrastructures linéaires de ce réseau spatial dont leurs poids sont des nombres flous.

D'après Zadeh (Zadeh, 1965), un sous ensemble flou est caractérisé par une fonction d'appartenance μ dans l'intervalle des nombres réels [0, 1], dénotant le degré d'appartenance.

Notons μ_A la fonction d'appartenance du poids d'une arête du graphe flou, dont la représentation trapézoïdale linéaire est donnée ci-après (figures 1 et 2) :

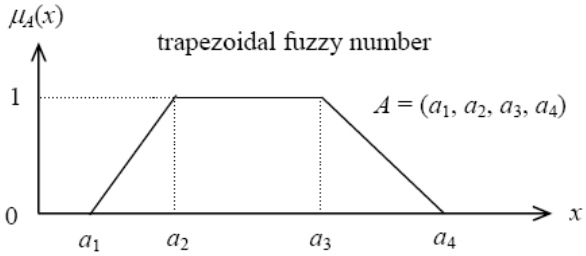


FIG.1 – *Nombre flou trapézoïdal.*

$\mu_A(x) = 0$ si $0 \leq x \leq a1$ ou $x > a4$
$\mu_A(x) = 1$ si $a2 < x < a3$
$\mu_A(x) = (x - a1) / (a2 - a1)$ si $a1 < x \leq a2$ (1)
$\mu_A(x) = (a4 - x) / (a4 - a3)$ si $a3 < x \leq a4$

Le poids flou d'une arête du graphe peut avoir une présentation triangulaire floue. Le poids flou peut être représenté alors, par un triple (*a1, a2, a3*) et sa fonction d'appartenance μ_A est donnée par :

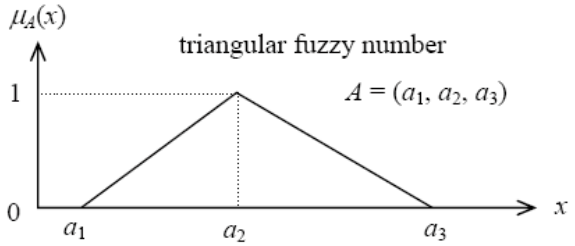


FIG.2 – *Nombre flou triangulaire.*

$\mu_A(x) = 0$ si $0 \leq x \leq a_1$ ou $x \geq a3$
$\mu_A(x) = 1$ si $x = a2$
$\mu_A(x) = (x - a1) / (a2 - a1)$ si $a1 < x \leq a2$ (2)
$\mu_A(x) = (x - a3) / (a2 - a3)$ si $a2 < x \leq a3$

3 Les nœuds d'origine et les points d'accès

Outre les nœuds du réseau spatial, il existe d'autre type de nœuds qui représentent des stations, des services ou des équipements d'infrastructure. On appelle ces nœuds les nœuds d'origine. Ces nœuds n'appartiennent pas au graphe modélisant ce réseau. Le diagramme de voronoï est basé sur la théorie des graphes, ce qui implique que le plus court chemin doit être calculé entre des nœuds qui appartiennent au graphe (figure 3).

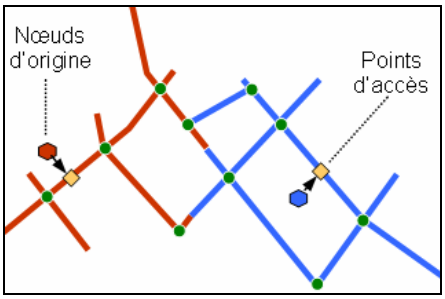


FIG. 3 – *DVR-Flou généré par des points d'accès localisés à partir des points d'origine.*

L'accès à ces noeuds d'origine doit être donc réalisé via des points spatialement référencés et qui appartiennent au graphe modélisant le réseau : ce sont les points d'accès. Ils constitueront les générateurs du diagramme de voronoï du réseau spatial.

Okabe a développé l'outil « SANET » (Okabe et al. 2000), (Okabe et al. 2002a) qui permet de générer les points d'accès à partir de deux couches de données géographiques intégrées dans ArcGIS (© ESRI) : une couche de lignes représentant les tronçons du réseau, et une couche de points représentant les points d'origine. Egalement, dans un travail précèdent (Mabrouk et Boulmakoul, 2008**)**, nous avons proposé une méthode qui permet d'une part, de chercher et localiser chaque point d'accès sur le plus proche segment du graphe à chaque point d'origine. D'autre part, de diviser ces segments avec des nouveaux nœuds qui sont insérés par la suite dans le graphe.

4 Diagramme de voronoï flou du réseau spatial

Définition 1: Le Diagramme de Voronoï Flou du réseau spatial (DVR-Flou) est définit par la division du réseau spatial en sous réseaux de Voronoï dont chacun contient les points les plus proches à chaque générateur de voronoï en parcourant le plus court chemin flou entre ces composantes.

4.1 DVR-Flou : diagramme basé sur la logique flou

Le diagramme de voronoï flou d'un réseau spatial se construit par l'assignation de chaque générateur de voronoï à ses plus proches nœuds du graphe modélisant un réseau spatial réel. Ceci implique :

1. la nécessité du calcul des poids flou total des plus courts chemins parcourus pour mettre en relation les générateurs de voronoï et le reste des sommets du graphe flou.
2. La comparaison de ces poids flous nous permettra par la suite de déterminer le générateur de voronoï le plus proche à chaque noeud du graphe.
3. L'assignation des arcs du graphe à leurs plus proches générateurs en se basant sur l'assignation des nœuds représentant leurs extrémités.

Des opérations arithmétiques et des fonctions de comparaison des nombres flous sont nécessaires pour accomplir cet objectif.

4.1.1 Addition des nombres flous

L'addition des deux nombres flous A(*a1, a2, a3*) et B(*b1,b2,b3*) peut être dérivée en utilisant le principe d'extension de Zadeh **(**Zadeh, 1965**)**, et se calcule comme suit :

$$\begin{aligned} A \oplus B &= (a1, a2, a3) \oplus (b1, b2, b3) \\ &= (a1+b1, a2+b2, a3+b3) \end{aligned} \qquad (3)$$

4.1.2 L'ordre sur les nombres flous

Dans la littérature, plusieurs auteurs (Boulmakoul, 2004)**,** (Chang et Lee, 1994), (Miloš Šeda, 2006) ont proposé des méthodes permettant de trier et de comparer des nombres flous. Dans le cadre du présent travail, nous utilisons la méthode proposée par (Ying-chao Shao, Zheng Pei, 2007) pour la simplicité de son implémentation. Ces auteurs ont utilisé la fonction GLRFN (General Left Right Fuzzy Number) définit ci-dessous :

$$f : F(R) \rightarrow (0, +\infty),$$
$$A\,(a1, a2, a3, a4) \rightarrow R_A$$

Soient *A (a1, a2, a3, a4)* et *B (b1, b2, b3, b4)* $\in$ *F(R)*

$$A\,(a1, a2, a3, a4) \leq B\,(b1, b2, b3, b4) \Leftrightarrow R_A \leq R_B \qquad (4)$$

Avec

$$\begin{aligned} R_{\bar{A}} &= \sqrt{2}|A(\frac{1}{3}[(1+2B+D)^{\frac{3}{2}} - D^{\frac{3}{2}}] \\ &+ (2E - 2D + 5BC - 11B^2) \\ &\cdot (\sqrt{1+2B+D} - \sqrt{D}) \\ &+ (C - 3B)\sqrt{1+2B+D} \\ &+ \frac{1}{2}(3CD - 5BD - 2BE - 5B^2C + 11B^3) \\ &\cdot (ln\frac{\sqrt{1+2B+D}+1+B}{\sqrt{1+2B+D}-1-B} - ln\frac{\sqrt{D}+B}{\sqrt{D}-A})|, \end{aligned}$$

$$\begin{aligned} A &= \frac{(a_2-a_1)(a_3-a_4)}{\sqrt{(a_2-a_1)^2+(a_3-a_4)^2}}, \\ B &= \frac{a_1(a_2-a_1)-a_4(a_3-a_4)}{(a_2-a_1)^2+(a_3-a_4)^2}, \\ C &= \frac{a_4(a_2-a_1)+a_1(a_3-a_4)}{(a_2-a_1)(a_3-a_4)}, \\ D &= \frac{a_1^2+a_4^2}{(a_2-a_1)^2+(a_4-a_3)^2}, \\ E &= \frac{a_1 a_4}{(a_2-a_1)(a_3-a_4)}. \end{aligned}$$

4.2 Voronoï spatial flou de type réseau : algorithmes proposés

Plusieurs algorithmes ont été proposés dans la littérature (Erwig, 2000), (Takehiro et al. 2005) permettent de calculer le diagramme de voronoï de type réseau. Dans un travail précédent (Mabrouk et Boulmakoul, 2008), nous avons appliqué ces algorithmes sur un réseau spatial pour évaluer l'accessibilité géographique. En se basant sur la logique de flou, nous étendons ces algorithmes afin d'évaluer d'une manière plus proche la réalité et la qualité des réseaux spatiaux. En effet, nous proposons deux algorithmes que nous allons décrire dans les paragraphes suivants :

Soit N(S,A) un graphe flou de sommets S et d'arcs A et G_n un ensemble de sommets G_n = {g1,…,gn} ⊆ S. G_n représente les générateur de voronoï dans le DVR-Flou. Chaque poids d'un arc, est représenté par un nombre flou. Considérons v et w deux sommets appartenant à S. Nous allons utiliser Pflou(v,w) pour représenter le poids du plus court chemin de v à w dans R. le DRV-Flou pour G_n divise le graphe N en sous réseaux de voronoï flous (figure 4) S(g1),….,S(gn) avec :

$$S(g_i) = \{\forall\ v \in V\ /\ P_{flou}(g_i,v) \leq P_{flou}(g_j,v),\ 1 \leq \forall j \leq n\} \quad (5)$$

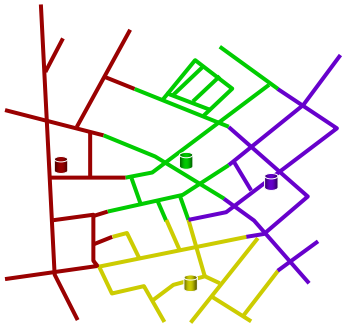


FIG. 4 – *Diagramme de Voronoï du réseau spatial flou.*

4.2.1 Algorithme utilisant les poids flous des plus courts chemins

En exploitant l'algorithme Dijkstra Flou proposé par (Boulmakoul, 2004) et (Miloš Šeda, 2006), notre algorithme calcule les poids flous des plus courts chemins parcourus entre chaque nœud du graphe et l'ensemble des générateurs de voronoï. En utilisant R la fonction de d'ordre portant sur les nombre flous décrite précédemment. Cet algorithme compare les poids flous trouvés et ensuite distingue le générateur de voronoï le plus proche au noeud sélectionné.

Calcul du des poids flous des plus courts chemins du sommet v à l'ensemble des générateur de voronoï

P_{flou}(v,Gn) : les poids flous des plus courts chemins flous du sommet v à l'ensemble des générateurs de Voronoï $g_i \in G_n$.
v : sommet du graphe flou choisi comme source
P_{flou}(v,g_i) : le poids flou du plus court chemin de v à un générateur $g_i \in$ Gn.
Adj[u] : l'ensemble des sommets voisins de u.
Q : la file d'attente associée par P_{flou}(v,u)

P_{flou}(v,v) :=(0,0,0); Inserer(v,Q);

Pour ∀ u∈ S-{v} **faire**
P_{flou}(v,u):= (∞,∞,∞); Inserer(u,Q);
Fin Pour

Tant que Q n'est pas vide **faire**
u := ExtraireMin(Q) ;
Pour ∀ w∈ Adj[u] **faire**
Si R(P_{flou}(v,u) ⊕ p(u,w))<**R**(P_{flou}(v,u)) **alors**
P_{flou}(v,w):= P_{flou}(v,u) ⊕ p(u,w); Mettre_à_jour(Q, w, P_{flou}(v,w));

Si w ∈ Gn **alors** P_{flou}(v,gi) :=P_{flou}(v,w) ; **Fin Si**
Fin Si
Fin Pour
Fin Tant que

Assignation des noeuds du graphe flou au générateur de voronoï le plus proche par la comparaison des poids flous.

Gt(v) : le plus proche générateur de voronoï flou temporaire à v;
Gp(v) : le plus proche générateur de voronoï flou permanant à v;
$Ppcc_{flou}$(v) : le poids flou fixe du plus court chemin de Gp(v) à v ;
Pt_{flou}(v) : le poids flou temporaire du plus court chemin de Gp(v) à v ;
P_{flou}i(v) : le poids flou du plus court chemin de v à un générateur g_i∈ G_n.

Pour chaque v ∈ **S - G_n faire**
Calculer P_{flou}(v,G_n) ; Gt(v):= g_1 ;

Pour chaque g_i∈ **Gn , 2≤∀j≤n faire**
Si ℜ(**P_{flou}(v,g_i)) ≤** ℜ(**P_{flou}(v,g_i-1)) alors**
Gt(v) := g_i; Pt_{flou}(v):= P_{flou}(v,g_i) ;
Sinon
Gt(v) := g_i-1; Pt_{flou}(v):= P_{flou}(v,g_i-1);
F**in Si**
Fin pour
Gp(v) := Gt(v) ; $Ppcc_{flou}$(v)= Pt_{flou}(v) ;
Fin pour

4.2.2 Algorithme Dijkstra parallèle flou

Cet algorithme, est basé sur l'algorithme de Dijkstra flou décrit précédemment. Il utilise une file d'attente où les opérations Inserer(), Extraire_Min() et Mettre_à_jour() sont disponibles. Il considère les générateurs de voronoï comme des sources multiples et cherche en parallèle les nœuds les plus proches à chaque générateur en se basant sur les poids flous des chemins parcourus.

Nous utilisons :
Gt(v) qui dénote le plus proche générateur de voronoï flou temporaire à v;
Gp(v) qui dénote le plus proche générateur de voronoï flou fixe à v ;
Ppccflou(v) qui dénote le poids flou du plus court chemin de Gt(v) à v.

Initialiser

Pour chaque v ∈ **S faire**
Si v ∈ G_n **alors** // v est un générateur
Gt(v):=v; $Ppcc_{flou}(v)$:=(0,0,0); Gp(v):= Γ; inserer(v,Q);
Sinon Gt(v) := Γ; $Ppcc_{flou}(v)$:= (∞,∞,∞) ; Gp(v) := Γ;
Fin Si

Marquer

v :=Extraire_ Min(Q); Gp(v) :=Gt(v) ; marquer v ;

Balayer

Soit Adj[v] l'ensemble des sommets voisins de **v** sélectionné.

Pour chaque w ∈ Adj[v] **n'est pas marqué faire**
Δ_{flou} := $Ppcc_{flou}(w) \oplus P_{flou}(v,w)$

Si $Ppcc_{flou}(w)$= (∞,∞,∞) **alors**
Gt(w):=Gt(v); $Ppcc_{flou}(w) = \Delta_{flou}$; inserer(w,Q);
Fin Si

Si $\mathcal{R}(Ppcc_{flou}(w)) \geq \mathcal{R}(\Delta_{flou})$ **alors**
Gt(w):=Gt(v); $Ppcc_{flou}(w) = \Delta_{flou}$; Mettre_à_jour(Q, w, Δ_{flou}) ;
Fin Si
Fin Pour

Contrôler

Tant que Q n'est vide **faire**
Marquer et **balayer**
Fin Tant que

4.3 Génération du diagramme de voronoï du réseau spatial flou

D'après (Margot Graf et Stephan Winter, 2003), chaque arc est marqué selon l'affectation de son noeud de début et de fin aux générateurs de voronoï. On distingue quatre cas différents qui peuvent se produire :

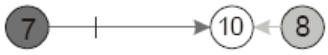
1. Le noeud de début et celui de la fin sont affectés au même générateur. Dans ce cas, l'arc est marqué par l'identifiant de ce générateur.

2. Le noeud de début et celui de la fin appartiennent aux différents générateurs, et l'arc déterminé par ces deux noeuds est unidirectionnel. Dans ce cas on ne peut pas

se déplacer de tout point sur l'arc vers le noeud de début. Alors n'importe quel point sur l'arc ne peut être atteint qu'à partir du noeud de début. Par conséquent, l'arc entier doit être assigné à l'arbre des plus courts chemins où appartient le noeud de début.

3. Le noeud de début et celui de la fin appartiennent aux différents générateurs. L'arc déterminé par ces deux nœuds est bidirectionnel et symétrique, et il ne peut être accédé qu'à partir de ses extrémités (noeud de début ou de fin). Dans ce cas l'arc entier est assigné à l'arbre des plus courts chemins où appartient le noeud (de début ou de fin) qui a un coût de déplacement minimal pour atteindre les générateurs les plus proches.

4. Le noeud de début et celui de la fin appartiennent à différents générateurs. L'arc est bidirectionnel et n'est pas nécessairement symétrique. Il est accessible par n'importe quel point qui lui appartient. Dans ce cas l'arc doit être divisé. Le point de division est le point auquel les coûts de déplacement au générateur du noeud de début sont égaux aux coûts de déplacement au générateur du noeud de fin. La première partie XA (noeud de début au point de division) sera assignée à l'arbre des plus courts chemins où appartient le nœud de début, et la deuxième partie, XE, du point de division au noeud de fin, sera assignée à l'arbre des plus courts chemins où appartient le noeud de fin. Au point de division on présentera un noeud qui n'est assigné à aucun arbre des plus courts chemins. Il représente la frontière entre deux cellules de voronoï de type réseau.



5 Accessibilité géographique de voronoï Flou : Un support spatial d'aide à la décision

5.1 Support spatial d'aide à la décision

L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans un processus de décision (Roy, B. 1985).

Un support spatial d'aide à la décision est défini comme étant un outil spécifique de résolution de problèmes basé sur une infrastructure d'informations spatiales. II est utilisé pour faciliter l'élaboration, le choix, la représentation et l'évaluation d'actions de nature spatiale (Chen et McLaughlin, 1992). Le but de tels supports est de fournir aux utilisateurs un outil qui leur permettra d'améliorer leur capacité à choisir ou à générer des solutions alternatives et de représenter spatialement ces situations, ainsi que leurs conséquences (Chen et Gold, 1992).

5.2 L'accessibilité géographique

L'accessibilité, comme elle est définit (Vincent Lenoir, 2007), est un indicateur complexe qui peut être créé par la combinaison, ou l'agrégation, des indicateurs simples qui traitent les trois éléments principaux du système morphologique urbain à savoir : le réseau, le bâti et le parcellaire (figure 5).

L'accessibilité géographique d'un lieu est définit (Magali Di Salvo, 2006) comme la capacité de ce lieu à être atteint à partir d'un ou de plusieurs autres lieux de localisation géographique différente, par un ou plusieurs individus susceptibles de se déplacer à l'aide de tout ou partie des moyens de transport existants. Elle traduit la pénibilité du déplacement, la difficulté de la mise en relation appréhendée le plus souvent par la mesure des contraintes spatio-temporelles.

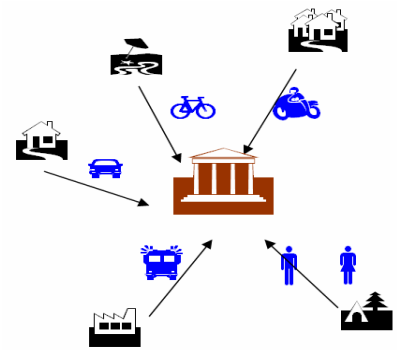


FIG. 5 – *L'accessibilité géographique d'un lieu.*

L'élément clé de l'accessibilité est la distance (Rodrigue, J-P *et al.* 1998). Elle illustre la friction de l'espace et l'endroit le plus accessible est celui ayant la friction minimale avec tous les autres endroits.

Conventionnellement, la distance est exprimée en kilomètres ou en unité temporelle, mais des variables telles que le coût ou l'énergie peuvent être utilisés.

5.3 Accessibilité géographique de voronoï flou

Définition 2 : l'accessibilité géographique de voronoï flou est l'accessibilité géographique à chaque « générateur de voronoï » par ses plus proches nœuds en parcourant le plus court chemin flou entre ces deux composantes (figure 6).

Le diagramme de voronoï réseau flou (DVR-Flou) permet de calculer les poids flous des plus courts chemins entre les nœuds du réseau spatial et leurs plus proches générateurs de voronoï (écoles, hôtels, hôpitaux, services publics, etc.…). Ceci permet d'évaluer de façon plus fidèle l'Accessibilité Géographique de Voronoï Flou par la formule suivante :

$$A_{flou}(G)i = \sum P_{flou}(i,j) \quad \text{avec:} \quad 1 \leq \forall j \leq n \qquad (5)$$

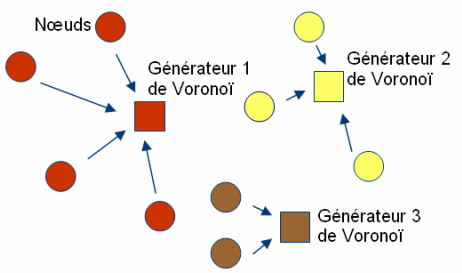


FIG. 6 – *Accessibilité géographique de Voronoï flou.*

A_{flou} (G)i: Accessibilité géographique de voronoï flou au générateur i ;
P_{flou}(i,j) : le poids flou du plus court chemin entre le générateur i et le nœud j associé à son plus proche générateur i ;
n : nombre de nœuds associés au générateur i.

6 Conclusion

Dans ce papier, nous avons proposé un processus de calcul du diagramme de voronoï du réseau spatial basé sur la modélisation par les graphes flous. En effet, nous avons déterminé les étapes pour construire ce diagramme qui consiste à assigner les arcs du graphe modélisant le réseau spatial à leurs plus proches générateurs de voronoï et ce en parcourant les plus courts chemins dont leurs poids sont des nombres flous. En se basant sur ces poids flous, nous avons évalué d'une manière objective l'accessibilité géographique flou de voronoï à tout service ou équipement. Cet indicateur complexe dérivé de l'accessibilité géographique « crisp » et basé sur l'information géographique spatiale, constitue un support spatial d'aide à la décision, tant en planification de transport que dans d'autres domaines. Le système développé est déployé sur des données urbaines de la ville de Tétouan (Nord du Maroc). Les mesures d'accessibilité seront validées par les experts urbains. Ce travail sera aussi considéré pour le calcul des zones de chalandises pour un projet de Géomarketing qui a été initié par notre équipe en 2008.

Références

Boulmakoul, A. "Generalized Path-Finding Algorithms on Semirings and the Fuzzy Shortest Path Problem," *Journal of Computational and Applied Mathematics*, vol. 162, pp. 263-272, 2004.

Boulmakoul, A., Mabrouk, A., *Diagramme de Voronoï du réseau spatial appliqués au problème de l'accessibilité- Conception et implémentation* -, Université Hassan II, FST de Mohammedia, Optique et traitement de l'information'08, avril 2008

Chang P. T. and E.S. Lee, "*Fuzzy Arithmetics and Comparison of Fuzzy Numbers*", in M. Delgado, J. Kacprzyk, J.-L. Verdegay and M.A. Vila (eds.), *Fuzzy Optimization*. Heidelberg: Physica-Verlag, pp. 69-82, 1994.

Erwig, M. *The Graph Voronoi Diagram with Applications. Networks*, 36 (3): 156-163, 2000.

Magali Di Salvo, *Calculs d'accessibilité : Impact des spécifications du réseau routier sur les calculs d'accessibilité Données sources méthodes*, centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques, Lyon, Janvier 2006.

Margot Graf et Stephan Winter , *Network Voronoï Diagrams*, Institut für Geoinformation, Technische Universität Wien, 2003.

Miloš Šeda, Fuzzy *Shortest Paths approximation for Solving the Fuzzy Steiner Tree Problem in Graphs*, Proceedings Of World Academy Of Science, Engineering And Technology Volume 14 August 2006 ISSN 1307-6884

Okabe, A., Boots, B., Sugihara, K., Chiu, S.N. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Chichester, United Kingdom, 2000.

Okabe, A., Okunuki, K., Funamoto, S. *SANET: A Toolbox for Spatial Analysis on a Network*, Center for Spatial Information Science, University of Tokyo, http://okabe.t.u-tokyo.ac.jp/okabelab/atsu/sanet/sanet-index.html, 2000.

Rodrigue, J-P *et al. Site Web Géographie des Transports*, Hofstra University: Department of Economics and Geography, 1998.

Roy, B. *Méthodologie multicritère d'aide à la décision*. Paris, Economica, 1985.

Takehiro Furuta, Atsuo Suzuki and Keisuke Inakawa, *The Kth Nearest Network Voronoi Diagram And Its Application To Districting Problem Of Ambulance Systems*, Nanzan University, No.0501, 2005

Vincent Lenoir, *Valorisation des SIG dans une démarche de formalisation du concept d'accessibilité piétonne*, Faculté de l'Environnement Naturel, Architectural et Construit (ENAC), Cycle Master - Semestre d'été 2007.

Yingchao Shao, Zheng Pei, *A Method for Ranking Fuzzy Numbers and Its Application to Game with Fuzzy Profit*, ISKE-2007 Proceedings, Advances in Intelligent Systems Research, October 2007

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

Summary

Spatial networks constitute an essential support for spatial organization of territory; the accessibility plays a key role in transportation dynamics. It is dependent on the transport network quality and the places structure; it influences our daily displacements and our mobility decisions. In this paper we adopt fuzzy spatial network voronoï diagram (Fuzzy-SNVD) an approach based on the fuzzy logic and on graph theory to evaluate fuzzy voronoï geographical accessibility to its generators (station, public facility, equipment…). This evaluation and its cartographic representation occupy a key position among necessary tools which get to the decision makers, true assistance support to decision-making in the transport planning. Indeed, we propose within this framework two algorithms which allow calculating the Fuzzy-SNVD and which consist in assigning each voronoï generator to its closer nodes of graph modelling a crisp spatial network. This assignment of nodes is based on the comparison of weights of the shortest traversed paths and which are considered fuzzy numbers representing in a way more faithful to reality, for duration, speeds or all values describing displacement.

Adaptation du scénario d'apprentissage

Lamia Fatiha Dali Youcef*, Mohamed Ismail Smahi**

*20, avenue EL Yebdri Mansour-Tlemcen-Algérie 13000
lamiadaliyoucef@univ-tlemcen.dz

**Ain Defla –Tlemcen- Algérie 13000
i_smahi@univ-tlemcen.dz

Résumé. L'enseignement à distance offre une nouvelle dimension dans le domaine de la formation pédagogique. Les dispositifs de formation en ligne ont un intérêt majeur puisqu'ils présentent un environnement d'apprentissage centré sur l'activité de l'apprenant, en lui fournissant des contenus et des parcours adaptés à ses besoins. Dans cet article, nous mettons l'accent sur les possibilités offertes en matière d'adaptation du scénario d'apprentissage. L'objectif est d'augmenter le niveau d'adaptation de l'unité d'apprentissage en offrant aux concepteurs une alternative pour opérer de modifications de la conception originale et instanciation du scénario d'apprentissage. En effet, nous proposons une adaptation au moment de l'instanciation du scénario d'apprentissage.

Mots clés. Scénario d'apprentissage, unité d'apprentissage, IMS-LD.

1 Introduction

En prenant en compte, d'une part, que le triplet conception, instanciation et déroulement représente les trois phases dans une unité d'apprentissage IMS (2003). De l'autre part, que les systèmes pédagogiques adaptatifs utilisent des méthodes et des techniques pour des fins de personnalisation relatives aux différents apprenants Brusilovsky, Peylo (2003). Nous mettons le point dans cet article sur, en premier lieu, les différentes possibilités de la prise en compte d'une action d'adaptation dans les différentes phases d'une unité d'apprentissage (Conception, Instanciation et Déroulement) via la spécification IMS LD IMS (2003). En second lieu, nous présentons notre propre modélisation pour le processus d'adaptation de l'unité d'apprentissage.

Les travaux en matière d'adaptation des scénarios d'apprentissage peuvent être scindés en deux catégories. La première concerne l'adaptation au moment de leur conception Berlanga, Garcia (2005a) Towle, Halm (2005). La seconde concerne l'adaptation au moment de leur déroulement Zarraonandia et al. (2006) Rosmalen, Boticario (2005). Dans ce papier, nous proposons une adaptation située entre les deux précédentes à savoir entre le moment de la conception des scénarios d'apprentissage et celui de leur déroulement. Il s'agit en effet d'adapter les scénarios d'apprentissage pendant le moment de leur instanciation.

Après cette introduction nous présentons, dans la section 2, les différentes méthodes et techniques d'adaptation utilisées dans les systèmes pédagogiques adaptatifs pour supporter les actions d'adaptation par rapport aux profils des apprenants. La section 3 introduit spécifi-

cation IMS-LD et son potentiel en terme d'adaptation à l'intérieur d'une unité d'apprentissage. La section 4, présente notre point de vue sur les différentes possibilités offertes pour réaliser une action d'adaptation dans les trois phases qui composent une unité d'apprentissage, et plus spécialement au moment de l'instanciation de la situation d'apprentissage, cela par rapport aux caractéristiques et préférences des différents apprenants. Nous terminons par des conclusions, et nous ouvrons des perspectives de recherches pour la poursuite de nos travaux.

2 Méthodes et techniques d'adaptation

Plusieurs méthodes et techniques d'adaptation ont été développées Brugos et al. (2006a) Brusilovsky, Peylo (2003) Brusilovsky (1996). Elles sont utilisées, en général, dans les systèmes pédagogiques adaptatifs et les systèmes pédagogiques intelligents. Leur objective est d'augmenter leur niveau de personnalisation et d'adaptation par rapport aux besoins des apprenants. Pour enlever l'ambiguïté qui existe entre ces méthodes et ces techniques, nous considérons que le processus d'adaptation est basé sur des méthodes et que ces dernières utilisent un ensemble de techniques (au minimum une) pour effectuer ce processus.

1. Méthodes d'adaptations : Il existe, dans la littérature, trois groupes de méthodes : (1) les méthodes dites traditionnelles Brusilovsky (1996), qui sont issues du domaine des systèmes hypermédias adaptatifs : l'*adaptation de la présentation*, l'*adaptation du contenu* et l'*adaptation de la navigation*. (2) les méthodes dérivées des précédentes issues des domaines des systèmes pédagogiques intelligents et des systèmes pédagogiques adaptatifs Brusilovsky, Peylo (2003) : l'*adaptation de l'appui (support) interactif de résolution des problèmes*, le *filtrage adaptatif d'information* et l'*adaptation par groupe*. (3) les méthodes définis par Daniel Brugos dans Brugos et al (2006a) : l'*adaptation de l'évaluation* et l'*adaptation en temps réel*.

2. Techniques d'adaptation : Dans Brusilovsky et al. (1998), cinq types de techniques ont été définies. Ces techniques peuvent être utilisées, d'une manière unique ou combinée, afin de permettre une action d'adaptation : *le guidage direct*, *l'ordonnancement des liens*, *le masquage des liens*, *l'annotation des liens* et *les cartes adaptatives*.

3 L'adaptation via la spécification IMS-LD

IMS Learning Design est une spécification basée sur les travaux de Rob Koper Koper (2001). Elle permet aux apprenants d'atteindre leurs objectifs par des activités ordonnées et dans des environnements spécifiés Kravcik, Gasevic (2006). Le modèle IMS LD décrit formellement un processus ou unité d'apprentissage (cour, module, séance, etc.). L'objectif de la spécification est de réaliser des unités d'apprentissage flexible (par rapport aux différents types de pédagogie), personnalisable (par rapport aux différents apprenants), interopérable (entre les différentes applications) et réutilisable (pour des contextes différents) IMS (2003).

La spécification IMS LD est structurée autour de trois niveaux : le niveau A, le niveau B et le niveau C. la figure 1 présente son modèle conceptuel :

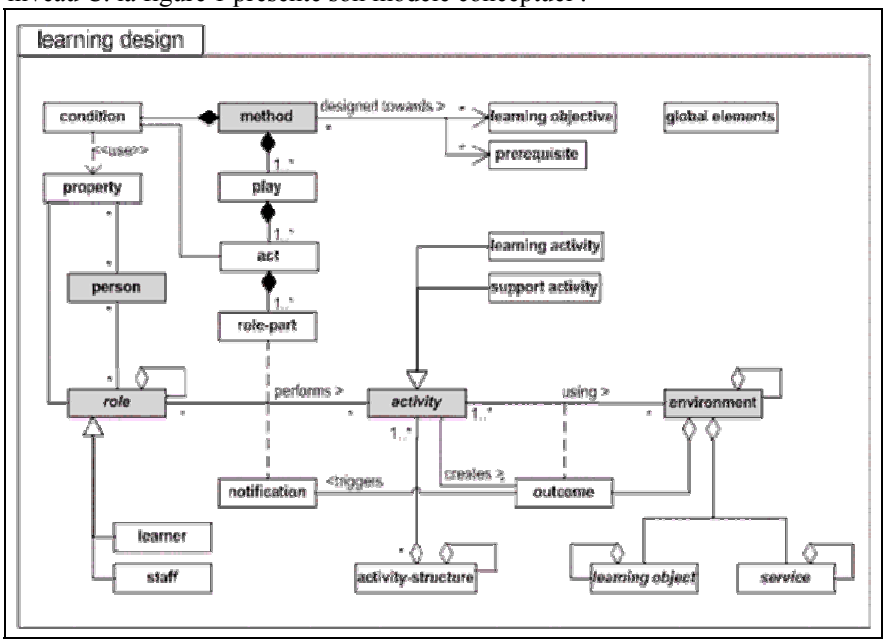


FIG. 1 – *modèle Conceptuel d'IMS LD IMS(2003).*

Le niveau A représente le noyau de la spécification. Il fait référence à la description des éléments qui configurent la spécification (Rôles, Activités et Environnement) et leur coordination via des éléments spécifiques (Methods, Play, Act, Role-part).

Le niveau B complète le niveau A par des propriétés, des conditions et des règles qui agissent l'ensemble. A travers l'évaluation des expressions (***si*** *Condition* ***alors*** *Action*), l'évolution du scénario d'apprentissage est bien déterminée. Ce qui permet de faire une personnalisation par rapport aux différents profils d'apprenants ou groupe d'apprenants.

Le niveau C ajoute et fournit des notifications au niveau A (déclenchement des activités par exemple envoi d'un mail).

Les niveaux B et C de la spécification permettent de construire et d'établir des scénarios d'apprentissage adaptés ou personnalisés Berlanga, Garcia (2005b et c) Towle, Halm (2005) Koper, Olivier (2004) aux différents apprenants, par l'utilisation des propriétés, des conditions, des éléments globaux et des agrégats.

4 L'adaptation dans les trois phases de construction de l'unité d'apprentissage

Une unité d'apprentissage Koper, Tattersall (2005) est une unité complète de travail pédagogique organisée selon une approche conceptuelle de l'apprentissage et qui assemble les

ressources liées, les liens Web et plusieurs services d'apprentissage dans un fichier XML « *content package* ». A l'intérieur de cette unité d'apprentissage le scénario d'apprentissage est modélisé via la spécification IMS LD. L'adaptation sur le scénario d'apprentissage consiste à adapter l'unité d'apprentissage, par contre une action d'adaptation de l'unité d'apprentissage revient à adapter soit le scénario d'apprentissage soit les ressources soit les deux en même temps. Dans le même contexte, l'adaptation d'un processus pédagogique est réalisée seulement si la modélisation, via la spécification IMS LD, des techniques et méthodes d'adaptation (cité ci-dessus) soit bien réalisée Brugos et al. (2006b) Brugos, Specht (2006) Specht, Brugos (2006) Berlanga, Garcia (2005a, b et c).

L'adaptation du scénario d'apprentissage peut se faire en deux phases différentes. Les travaux de Berlanga, Garcia (2005a) Towle, Halm (2005) partent du principe que l'adaptation du scénario d'apprentissage doit être réalisée pendant sa conception, dont la tache principale est donnée uniquement aux enseignants concepteurs qui doivent prendre en considération tous les cas possibles pour la mise en œuvre d'une situation d'apprentissage adaptative. Par contre, les travaux de Zarraonandia et al (2006) Rosmalen, Boticario (2005) partent du principe que le support d'adaptation doit être modélisé au moment même du déroulement de l'unité d'apprentissage, afin de supporter des actions de modification et de personnalisation. En effet, dans cette approche l'acteur principale est le tuteur qui doit réagir selon un ensemble d'événements déclenchés par l'apprenant en temps réel et tout au long de sa session d'apprentissage. Cependant, ces deux approches présentes plusieurs inconvénients :

1. Pour la première approche Towle, Halm (2005)

   - « *La difficulté engendrée par l'interaction entre plusieurs règles (impossible à un enseignant de prendre toutes les règles possible en même temps)*
   - *L'impossibilité de modification de l'unité d'apprentissage après la génération du manifeste*
   - *La redondance de stratégies d'adaptation (les mêmes stratégies peuvent être intégrées dans plusieurs manifestes.*
   - *Toute la connaissance est imbriquée à l'intérieur de l'unité d'apprentissage, d'où l'impossibilité de la réutilisée dans d'autre unité d'apprentissage* »

2. Pour la deuxième approche Rosmalen, Boticario (2005)

   - *« La difficulté de combiner entre l'adaptation effectuée au moment du déroulement avec l'adaptation effectuée au moment de conception du scénario d'apprentissage*
   - *La difficulté de contrôler l'apprenant durant toutes sa session de travail* »

4.1 L'adaptation au moment de l'instanciation

Par référence, d'une part, à la comparaison de Brusilovsky Brusilovsky, Peylo (2003) entre les systèmes intelligents et les systèmes adaptatifs, dont l'intersection représente le mieux le processus d'adaptation. Et de l'autre part, qu'un processus d'adaptation tel qu'il est définit par Brusilovsky est composé de deux actions, importantes et complémentaires, centré sur le système d'un côté : « *adaptabilité* », et sur l'apprenant de l'autre côté : « *adaptativité* ». Nous considérons que le manque d'un aspect intelligent pour effectuer l'adaptation dans un scénario d'apprentissage représente un autre inconvénient pour les deux approches. En effet, dans les deux alternative (approches) l'adaptation sera effectuée soit par l'enseignant concepteur au moment de la conception (dans ce cas, elle sera faite de la même manière pour l'ensembles des apprenants), soit par le tuteur au moment du déroulement de l'unité d'apprentissage en réagissant aux événements déclenchés par chaque apprenant a part et en temps réel (ce qui demande un suivi permanant de chaque apprenant par le tuteur).

Pour augmenter le niveau d'adaptativité dans un scénario d'apprentissage nous devons aider l'enseignant dans sa tâche de conception, d'un côté, et le tuteur pendant le déroulement de la situation d'apprentissage de l'autre côté, cela par la génération automatique, de règles d'adaptation. Part la suite, nous allons intégrer ces règles dans le manifeste de l'unité d'apprentissage sans que le scénario crée par le concepteur soit modifié.

Dans ce cas, nous devons poser les deux questions suivantes :

1. A quel moment (dans le cycle de vie du scénario d'apprentissage) peut-on introduire ces modifications ?
2. Comment augmenter le niveau d'adaptativité d'un scénario d'apprentissage, par l'introduction d'une manière automatique de règles d'adaptation ?

Selon les travaux de Zarraonandia et al. (2006) Berlanga, Garcia (2005b et c) Towle, Halm (2005) Rosmalen, Boticario (2005), nous remarquons que la conception et le déroulement représentent les deux phases uniques dans l'unité d'apprentissage, par contre la spécification IMS LD distingue trois phases différentes (**Erreur ! Source du renvoi introuvable.**), allant de la conception jusqu'au déroulement de l'unité d'apprentissage, en passant par l'instanciation de cette dernière. Donc, le moment qui nous semble opportun pour effectuer l'adaptation sera juste après que l'enseignant a fini sa conception et avant le lancement de la situation d'apprentissage pour l'apprenant.

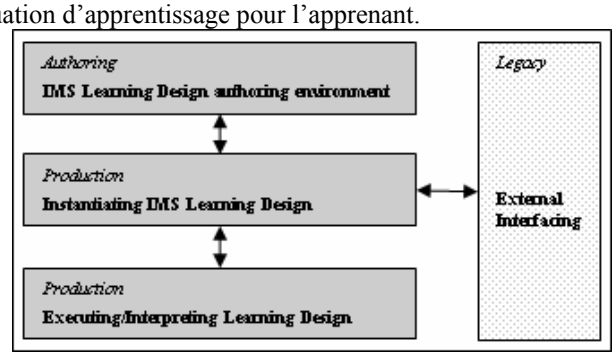


FIG. 2 – *Les trois phases d'une unité d'apprentissage (IMS, 2003)*

De ce constat, nous nous intéressons aux possibilités d'adaptation qui sont offertes au moment d'instanciation de l'unité d'apprentissage.

L'idée initiale est inspirée du travail préparatif pour les scènes théâtrales, où le scénario, initial, de la pièce doit être joué plusieurs fois avant sa présentation finale, et pendant ces répétitions plusieurs modifications et améliorations peuvent être introduites. Par le même principe, sur un scénario d'apprentissage, nous devons opérer à plusieurs modifications, pour des fins d'adaptation, avant le déroulement final de l'unité d'apprentissage. D'où, à cette phase précise, nous effectuerons les modifications à l'intérieur de l'unité d'apprentissage et non pas à l'extérieur. La (Fig. 3 – L'adaptation au moment de l'instanciation) présente le processus que nous devons déclencher au moment de l'instanciation de l'unité d'apprentissage.

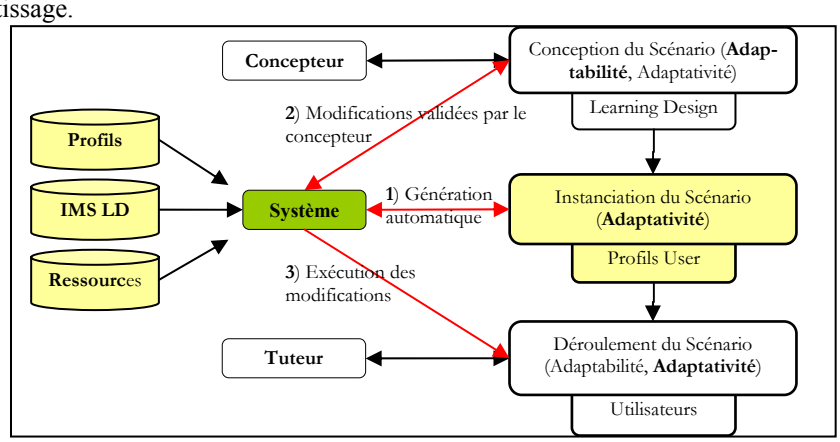


FIG. 3 – *L'adaptation au moment de l'instanciation*

Notre processus est composé de trois étapes:

1. ***Génération des règles*** : cette action consiste à déterminer les meilleures règles qui peuvent être appliquées sur le scénario d'apprentissage en se basant sur : la spécification IMS LD, une base de profils et les ressources pédagogiques disponibles.

2. ***Validation des règles*** : une fois la génération des règles est terminée, la deuxième phase consiste à la validation de ses règles par l'enseignant concepteur.

3. ***Exécution des règles*** : cette phase consiste à la modification de la situation d'apprentissage par l'application de ses règles au moment du déroulement de la situation d'apprentissage.

4.2 La modélisation du processus d'adaptation

4.2.1 Mise en œuvre

L'adaptation d'un scénario d'apprentissage au sein d'une unité d'apprentissage revient à la description de plusieurs modifications sur plusieurs éléments dans le processus d'apprentissage. La définition de ces modifications doit être modélisée par des règles XSLT

qui vont être générées au moment de l'instanciation et appliquées au moment du déroulement de l'unité d'apprentissage. Ces règles vont être intégrées avec l'unité d'apprentissage sans effectuer aucun changement sur l'organisation de la situation d'apprentissage initiale (scénario d'apprentissage) déjà crée par l'enseignant dans la phase de conception (Fig. 4 – Intégration des modifications sur l'unité d'apprentissage
). Nous devons appliquer ces règles pour des:

4. Adaptations du parcours entre les activités ;
5. Adaptations opérées à l'intérieure des activités ;
6. Adaptations des ressources utilisées dans le scénario.

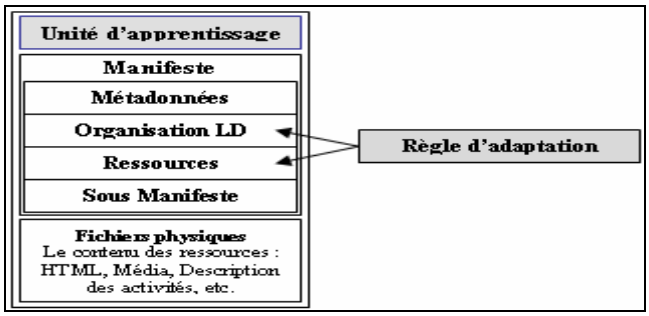


FIG. 4 – *Intégration des modifications sur l'unité d'apprentissage*

La spécification IMS LD permet d'intégrer les ressources pédagogiques, qui accompagnent la situation d'apprentissage, à l'intérieur du package généré. Par contre dans notre approche si nous voulons faire une adaptation sur une ressource pédagogique (par exemple ajouter d'autres ressources), nous allons générer une règle XSLT qui permet de spécifier l'URI de la ressource concernée. Dans ce cas, nous considérons que toutes les ressources soient centralisées dans une base de ressources.

4.2.2 Exemples d'adaptation

Pour plus de compréhension, nous illustrons, dans cette partie, à travers des exemples quelques types d'adaptation opérés sur un scénario d'apprentissage initial, ce là par l'application de règles XSLT générées automatiquement. L'unité d'apprentissage que nous avons utilisé a été conçue et exécutée via les outils *Reload Learning Design*[1].

7. Exemple1 : **Adaptation sur la séquence des activités** : les recherches ont montré Shute (1993) que certains étudiants exhibent un comportement exploratoire, ce qui leurs donnent plus de compréhension si des exemples de clarification précèdent la définition d'un concept, alors que certains apprenants préfèrent avoir la définition du concept avant les exemples. Cette stratégie d'adaptation peut être implémentée comme suit. La première phase consiste à la détermination si des définitions de concepts avec des exemples existent dans la situation d'apprentissage actuelle. La seconde phase, consiste à détecter si dans la base des profils il se trouve que certains étudiants ont des préférences différentes sur la manière que les concepts et les exemples soient présentés. Après, le processus d'adaptation doit

[1] http://www.reload.ac.uk/

générer une feuille de style (Figure 5) qui va être appliquée sur le scénario du départ afin d'adapter la situation d'apprentissage (Figure 6).

```
....
<xsl:element name="learning-activity">
     <xsl:attribute name="Identifier">Cons_Exemple</xsl:attribute>
     <xsl:attribute name="isvisible">true</xsl:attribute>
</xsl:element>
<xsl:element name="learning-activity">
     <xsl:attribute name="Identifier">Visio_Conf</xsl:attribute>
     <xsl:attribute name="isvisible">true</xsl:attribute>
</xsl:element>
....
```

FIG. 5 – *La génération de la feuille XSLT*

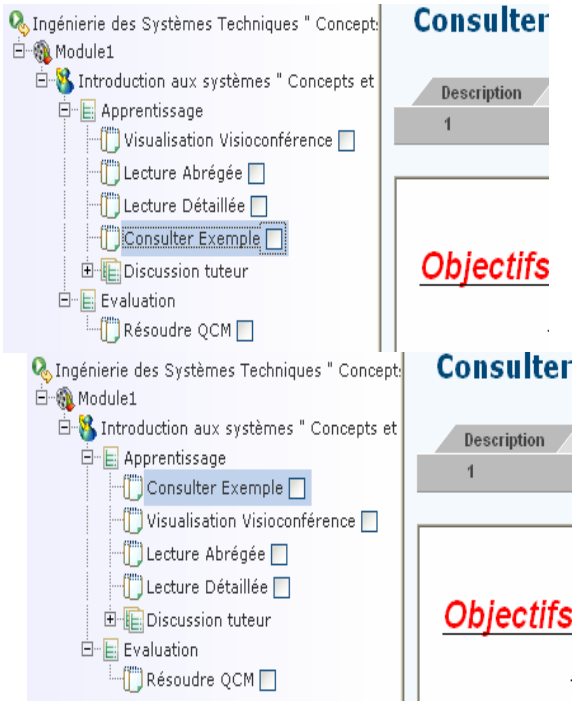


FIG. 6 – *La structure du cours avant et après l'adaptation*

8. Exemple 2 : **Adaptation des ressources** : Une situation d'apprentissage est composée de trois dimensions Faerber (2004); une problématique définie par l'enseignant, un traitement de cette problématique par l'apprenant et un environnement technologique qui contient un ensemble de ressources numériques. Ainsi, pour la dernière dimension, l'enseignant doit sélectionner la meilleure ressource pédagogique disponible, afin de la intégrée dans l'unité d'apprentissage. Cette action d'adaptation est concrétisée par l'application d'une feuille XSLT (Figure 7) sur la situation d'apprentissage initiale, qui a pour but d'ajouter ou de supprimer des ressources pédagogiques pour certains apprenants (Figure 8).

```
.....
<xsl:element name="learning-object">
    <xsl:attribute name="Identifier">Id_Res_Def_En</xsl:attribute>
    <xsl:attribute name="isvisible">true</xsl:attribute>
    <xsl:element name="title">Définitions (EN)</xsl:element>
    <xsl:element name="item">
        <xsl:attribute name="Identfier">Id_Res</xsl:attribute>
        <xsl:attribute name="identifierref">Defini_En.pdf</xsl:attribute>
        <xsl:attribute name="isvisible">true</xsl:attribute>
    </xsl:element>
</xsl:element>
.....
```

FIG. 7 – *Les règles XSLT générées*

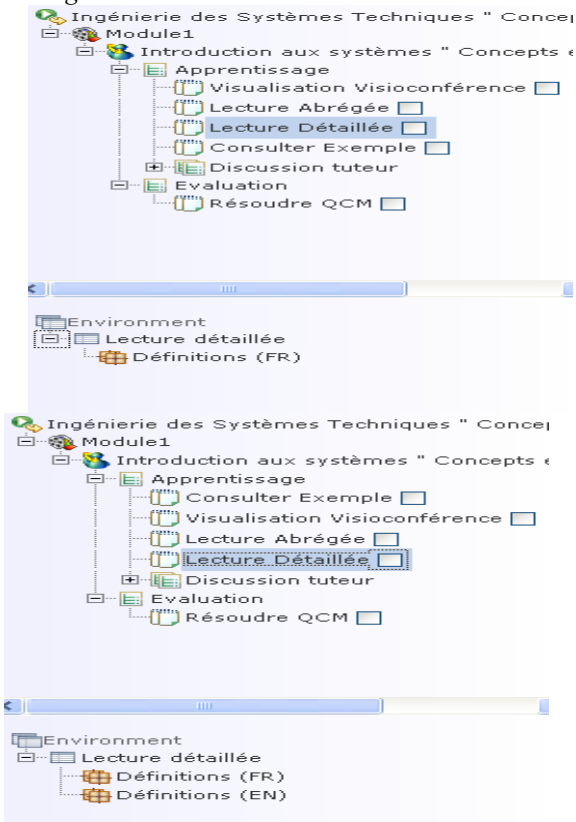


FIG.8 – *Modification de l'environnement pour la même activité*

Conclusion

Notre thème de recherche est l'adaptation des unités d'apprentissage, et plus spécifiquement l'adaptation du déroulement de la situation d'apprentissage (scénario d'apprentissage) et ces ressources pédagogiques associées. Notre principal objectif est de développer un mécanisme pour supporter une action d'adaptation sur les scénarios d'apprentissages. Le but est d'augmenter le niveau d'adaptativité d'une unité d'apprentissage

en utilisant un procédé simple pour présenter les modifications opérées sur la conception originale.

L'avantage de notre approche réside dans le choix du moment adéquat pour effectuer des actions d'adaptation, c'est-à-dire au moment de l'instanciation de l'unité d'apprentissage, en générant automatiquement des règles, qui vont être validées par l'enseignant concepteur, en premier lieu, et exécutées au moment du déroulement, en second lieu. Ces règles vont être intégrées dans l'unité d'apprentissage sans effectuer aucun changement sur la conception originale.

Les prochains points que nous développerons, doivent prendre en compte l'aspect clé dans les systèmes adaptatifs, qui est l'apprenant. La modélisation de l'apprenant doit être réalisée par la spécification IMS LIP (Learner Information Package) LIP (2001) d'un côté et par des observations sur ce dernier afin d'effectuer des modifications plus complexe. En outre, et comme finalité, un outil logiciel doit être développé pour supporter ces actions d'adaptation opérées sur la conception originale de la situation d'apprentissage.

Références

Berlanga A. J., Garcia F. J. (2005a), Modelling Adaptive Navigation Support Techniques Using the IMS Learning Design Specification, *ACM International Conference Proceeding*, pp.148-150.

Berlanga A. J., Garcia F. J. (2005b), Authoring tools for adaptive learning designs in computer-based education, *ACM International Conference Proceeding*, Vol.124, pp.190-201.

Berlanga A. J., Garcia F. J. (2005c), Using IMS LD for Characterizing Techniques and Rules in Adaptive Educational Hypermedia Systems, *UNFOLD/Prolearn Proceeding*, pp.61-80.

Brugos D., Specht M. (2006), Adaptive e-learning methods and IMS Learning Design. An integrated approach, *Proceedings of ICALT (IEEE)*, pp.1192-1193.

Brugos D., Tattersall C., Koper R. (2006a), Representing adaptive eLearning strategies in IMS Learning Design, *TENCompetence Conference*.

Brugos D., Tattersall C., Koper R. (2006b), How to represent adaptation in eLearning with IMS Learning Design, *Educational Thechnology Expertice Centre*.

Brusilovsky P. (1996), Methods and Techniques of Adaptive Hypermedia, User Modeling and User-Adapted Interaction, Vol.6, pp. 87-127.

Brusilovsky P., Peylo C. (2003), Adaptive and Intelligent Web-based Educational Systems, *International Journal of Artificial Intelligent in Education*, pp.156-169.

Brusilovsky P., Eklund J., Schwarz E. (1998), Web-based education for all: A tool for developing adaptive courseware, *The 7th World Wide Web Conference*, pp.291-300.

Faerber R. (2004), Caractérisation des situations d'apprentissage en groupe, *revue STICEF*, Vol. 11, pp.297-331.

IMS (2003) Global Learning Consortium. IMS Learning Design Specification.

Koper R., Tattersall C. (2005), *Learning Design*, The Netherlands, Springer.

Koper R. (2001), Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML, Open University of the Netherlands.

Koper R., Olivier B. (2004), Representing the Learning Design of Units of Learning, *Educational Technology & Society*, Vol. 7, p.97-111.
Kravcik M., Gasevic D. (2006), Knwoledge Representation for Adaptive Learning Design, *Proceedings of Adaptive Hypermedia*, pp.274-284.
LIP (2001) Global Learning Consortium: IMS Learner Information Package.
Rosmalen P., Boticario J. (2005), Using Learning Design to Support Design and Runtime Adaptation, *Learning Design*, Springer, pp.291-301.
Shute V. J. (1993), A Comparison of Learning Environments: All that Glitters, *Computers as Cognitive Tools*, Hillsdale, Lajoie SP, Derry SJ (eds), Lawrence Erlbaum.
Specht M., Brugos D. (2006), Implementing Adaptive Educational Methods with IMS Learning Design, *Proceedings of Adaptive Hypermedia*, pp.241-251.
Towle B., Halm M. (2005), Design Adaptive Learning Environments with Learning Design, *Learning Design*, The Netherlands, Springer, pp. 215-226.
Zarraonandia T., Fernandez C., Dodero J. M. (2006), A late Modelling Approach for the Definition of Computer-Supported Learning Process, *Pro. Adaptive Hypermedia*, pp.241-251.

Summary

The e-learning has a new dimension in the teaching training area. Indeed, the devices of formation on-line have a major interest since they present an environment of learning centred on the activity of learner, by providing him an adapted contents and courses. In this article, we stress the possibilities offered to carry out modifications on a teaching scenario for ends of personalization. The objective is to increase the level of adaptation of the unit of learning by offering to the designers an alternative to operate with modifications on their original design of this situation of learning, that at the instantiation time of the unit of learning.

Keywords

Scenario of learning, unit of learning, IMS-LD.

Extraction des règles à partir des données : Graphes d'inductions et automates d'arbres

Souad TALEB ZOUGGAR, Baghdad ATMANI

Département Informatique, Faculté des Sciences, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie
souad.taleb@gmail.com, atmani.baghdad@univ-oran.dz

Résumé. Dans le domaine de l'apprentissage automatique plusieurs méthodes dédiées à la tâche de classification ont été mises au point. Nous nous intéressons particulièrement aux techniques symboliques et plus précisément aux méthodes à base de graphes d'induction.
Dans cet article nous proposons une approche structurée pour la génération de graphes d'induction en utilisant un formalisme ayant fait ses preuves dans plusieurs domaines de l'informatique : les automates d'arbres.
Notre contribution dans ce domaine concerne le post-élagage des modèles de classification à base de graphes d'induction. Nous nous sommes fixés comme objectifs l'expérimentation des algorithmes de minimalisation d'automates d'arbres pour optimiser les modèles de classification générés.
Mots-clés : Classification, Apprentissage automatique, Graphe d'induction, Automates d'arbres, Automates d'arbres stochastiques.

1 Introduction

La classification par apprentissage automatique est au cœur du processus global d'extraction de connaissances à partir de données. Plusieurs méthodes ont été misent au point pour résoudre ce problème (Quinlan, 1986, Rabaseda et al., 1995, 1996b); parmi lesquelles, on retrouve les méthodes statistiques, à base de réseaux de neurones (Atmani et Beldjilali, 2007a), à base d'arbres de décision (Atmani et Beldjilali, 2007b), à base de graphes d'induction (Zighed et al., 1992), etc.

Les méthodes à base de graphes d'induction présentent un grand intérêt pour les chercheurs car ce sont des méthodes intelligibles qui représentent graphiquement un ensemble de règles, dites règles de classification, qui sont facilement interprétables.

Notre contribution s'inscrit dans le domaine de l'apprentissage automatique à partir de données et utilise comme base de travail les méthodes de classification supervisées à base de graphe d'induction. L'objectif est la conception et l'expérimentation de techniques de génération et d'optimisation des graphes d'induction par une méthode formelle de modélisation. Pour cela, nous allons expérimenter la théorie des automates et en particulier celle des automates d'arbres ; qui sont des modèles de calcul à états finis généralisant les automates classiques à la reconnaissance de termes.

La finalité de ce travail est de simplifier le modèle de classification traduit dans le formalisme d'automates d'arbres en utilisant les algorithmes de simplification d'automates d'arbres existants.

Cet article est structuré comme suit. La section 2 est consacrée aux automates d'arbres, et en particulier à la présentation d'une synthèse des différents travaux de recherche à base de

cet outil formel dans le domaine de l'apprentissage automatique. Dans la section 3 nous illustrons à travers un exemple notre contribution à base d'automates d'arbres. Enfin, nous concluons en précisant que le modèle proposé est valable pour les méthodes arborescentes de classification existantes (Taleb Zouggar et Atmani, 2008) et nous exposons aussi nos besoins en terme de conception en proposons d'utiliser les automates stochastiques permettant de prendre en compte plus de concepts concernant les graphes d'induction. Enfin, nous proposons une perspective visant à optimiser l'automate d'arbres obtenu et par conséquent optimiser la base de règles générée pour la classification.

2 Etat de l'art sur les automates d'arbres

Les automates d'arbres (Comon et al., 2005) ont été employés à l'origine pour la vérification de circuits et ont été ensuite souvent utilisés dans le cadre de l'interprétation abstraite à partir de contraintes, dans le contexte de problèmes de réécriture, de preuves automatiques de théorèmes ou encore de vérification de programmes. Ils ont également été utilisés pour l'inférence de programmes logiques.

Leur utilisation s'est récemment étendue dans des applications diverses :

Dans le domaine de traitement de données structurées en arbres, telles que les données au format XML, les travaux de (Dal Zilio et Lugiez, 2002a) consistaient à la vérification de la pertinence d'un modèle réalisé à partir d'automates d'arbres pour la compilation de langage existants pour l'interrogation de documents XML. Les mêmes auteurs (Dal Zilio et Lugiez, 2002b) ont abouti vers une nouvelle classe d'automates : les sheaves automata et une logique dérivée de la logique des ambiants dédiés à l'interrogation de ces documents.

Dans le même contexte Dal Zilio et Acciai (2004) utilisent une extension de la classe d'automates d'arbres proposée dans (Dal Zilio et Lugiez 2003, Dal Zilio et al., 2004) et proposent des opérateurs de filtrage pour l'interrogation de tels documents. Dans le domaine de la validation de documents XML, Bouchou et al. (2003a) proposent d'utiliser les automates d'arbres pour la vérification des contraintes de clés des documents XML suite à une mise à jour et dans un autre travail des mêmes auteurs (Bouchou et al., 2003b) les automates sont utilisés pour tester de façon incrémentale, en vérifiant seulement la partie du document concernée par les mises à jour, la validité d'un document.

Les protocoles cryptographiques sont des spécifications de suites de messages entre agents, utilisant des moyens de chiffrement (entre autres), et cherchant à assurer des besoins en confidentialité, authentification, ou d'autres propriétés. Leur vérification par les outils classiques s'est avérée coûteuse ou non satisfaisantes. Monniaux (1999) a proposé une nouvelle approche de vérification basée sur l'interprétation abstraite et l'utilisation de langages réguliers d'arbres, Genet et Klay (2000) proposent une nouvelle technique basée sur des systèmes de réécriture et sur des automates d'arbres.

Toujours dans le domaine de la vérification, Genet et al. (2003) réalisent la preuve d'une manière automatisée sur un modèle de réécriture de termes du protocole utilisant TIMBUK (Genet et al., 2001). TIMBUK est un outil de vérification qui utilise l'interprétation abstraite sur des domaines d'automates d'arbres.

Dans le domaine de l'apprentissage automatique, le travail de Carrasco et al. (2001) consiste à généraliser un ancien algorithme pour l'identification de langages réguliers à partir d'exemples stochastiques au cas de langages d'arbre. Il sera décrit aussi une méthode de cal-

culer efficacement l'entropie relative entre la grammaire cible et celle inférée, utile pour l'évaluation de l'inférence.

Habrard et al. (2002) étudient le problème de l'apprentissage d'une distribution statistique de données dans une base de données relationnelle en proposant une méthode capable de prendre en compte la structure de la base de données et ne nécessite pas de transformations de données qui peuvent engendrer une perte d'informations essentielles. Le travail de Habrard (2004) dans le cadre de l'inférence grammaticale s'intéresse principalement à deux problématiques: le traitement de données bruitées ou non pertinentes, et l'extraction de connaissances à partir de données arborescentes.

Le travail de Tommasi (2006) aborde la question sous l'angle de la réalisation automatique de programmes d'annotation d'arbres, permettant de dériver des procédures de transformation ou d'exécution de requêtes.

Dans un contexte applicatif, plusieurs systèmes à base d'automates d'arbres ont été mis au point, à titre d'exemple, le système GIFT « Grammatical Inference for Terms » (Bernard et De La Higuera, 1999, 2001) est un système qui apprend un automate d'arbres à partir d'un ensemble de termes, qui par la suite sera traduit en un programme logique, l'algorithme est appliqué à des données structurées, et un système de typage est aussi inféré pour éviter les situations incorrectes. CPV « Cryptographic Protocol Verification » (Goubault, 2000) est un logiciel de vérification automatique de propriétés de confidentialité de protocoles cryptographiques fondé sur des techniques d'automates d'arbres.

Le système Squirrel (Carme et al., 2006) utilise l'algorithme d'apprentissage de langages d'arbres d'arité non bornée qui dérive de RPNI (Oncina et Garcia, 1992). Le travail a été formulé pour les séquences mais qui repose sur des principes que l'on retrouve à la fois dans les automates de mots et dans les automates d'arbres.

Enfin, pour terminer ce bref historique sur les automates d'arbres nous soulignons qu'il existe des outils destinés à la manipulation d'automates d'arbres comme par exemple TIMBUK Genet et al. (2001), etc.

3 Conception d'un modèle d'automates d'arbres pour la classification

3.1 Définition d'un automate d'arbres

Un automate d'arbres fini A (voir Habrard, 2004) est un quadruplet A= (Q,Σ_S,δ,Q_f) où:

Σ_S	Signature définie par le quadruplet (τ,V, arité, σ) :
τ :	Ensemble fini dont les éléments sont des types,
V :	Alphabet dont les éléments sont des symboles fonctionnels,
arite: $V \rightarrow N$	arite (f) appelée l'arité de f,
σ: $V \rightarrow \tau$	$\sigma(f)$ est appelée le type de f,
Q	Ensemble fini d'états,
$\delta : V \times Q^* \rightarrow Q$	Fonction de transition,
$Q_f \subseteq Q$	Ensemble d'états finaux ou d'acceptation.

– Pour les automates ascendants, les règles de transition sont de la forme : $\delta(f,q_1,...,q_s)=q$ pour un symbole $f \in V$ dont le nombre d'arguments (arité=s>0). Ces règles peuvent aussi s'écrire : $f(q_1,q_2,\ldots,q_s) \rightarrow q$.

Un arbre t est accepté par l'automate ascendant A ssi $\exists$ q $\in Q_f$; δ'(t)→q, δ' correspond à plusieurs applications de δ.

– Pour les automates descendants, le quadruplet est A=(Q,Σ_S,δ,Q_I) où $Q_I \subseteq Q$ est l'ensemble des états initiaux ; l'ensemble des états finaux est remplacé par l'ensemble des états initiaux, une règle de transition est de la forme : q → f(q_1,q_2,…,q_s).

Un arbre t est accepté par l'automate descendant A ssi $\exists$ q $\in Q_I$; q → δ'(t).

– L'ensemble des arbres reconnus par un automate A définit le langage reconnu par A, noté L (A).
– Une règles-ε est une règle reliant deux états, elle est de la forme : $q_i \rightarrow q_j$; $q_i, q_j \in Q$.
– Une transition f (q_1,q_2,…,q_s)→q est représentée graphiquement par le schéma ci-dessous :

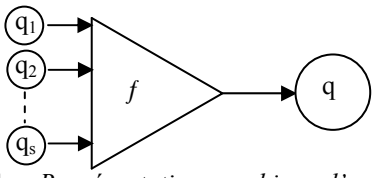


FIG. 1 – *Représentation graphique d'une transition*

3.2 Contribution à travers un exemple

L'exemple est présenté dans (Rabaseda, 1996a), il contient 22 exemples de bonnes ou mauvaises conditions de circulation en voiture, en ville.

Chaque exemple est décrit par quatre attributs (descripteurs) : le temps, les heures de circulation, les jours fériés, la taille de la ville.

Selon les valeurs des descripteurs, un exemple peut être classé dans la variable conditions de circulation (classe).

Temps	Heures	Jour-Férié	Taille-ville	Conditions
Beau	Creuse	Non	Grande	Bonnes
Beau	Creuse	Non	Petite	Bonnes
Neigeux	Creuse	Non	Grande	Mauvaises
Pluvieux	Sortie	Non	Grande	Mauvaises
Pluvieux	Soir	Oui	Grande	Mauvaises
Pluvieux	Soir	Oui	Petite	Bonnes
Neigeux	Soir	Oui	Petite	Mauvaises
Beau	Sortie	Non	Grande	Mauvaises
Beau	Soir	Non	Grande	Bonnes
Pluvieux	Sortie	Oui	Grande	Mauvaises
Beau	Sortie	Oui	Petite	Mauvaises

Suite des individus de l'échantillon d'apprentissage				
Neigeux	Sortie	Non	Petite	Mauvaises
Neigeux	Creuse	Oui	Grande	Mauvaises
Pluvieux	Sortie	Non	Petite	Bonnes
Pluvieux	Soir	Non	Petite	Bonnes
Beau	Sortie	Non	Petite	Bonnes
Pluvieux	Sortie	Oui	Petite	Bonnes
Pluvieux	Creuse	Non	Petite	Bonnes
Pluvieux	Creuse	Non	Grande	Mauvaises
Beau	Creuse	Oui	Grande	Bonnes
Neigeux	Soir	Oui	Grande	Mauvaises
Beau	Soir	Oui	Petite	Mauvaises

TAB.1 – *Echantillon d'apprentissage des conditions de circulations*.

3.2.1 Apprentissage supervisé et graphes d'induction

Apprentissage supervisé. Le but de l'apprentissage supervisé est de construire un modèle de prédiction, appelé aussi classifieur, qui nous permettra d'identifier un attribut à prédire Y, appelé variable endogène, classe, variable à expliquer, variable à prédire, à partir d'un certain nombre d'attributs explicatifs appelés variables exogènes, variables explicatives, variables prédictives, notées X (voir Denis et Gilleron, 1997).

- Le modèle de prédiction ou fonction de classement φ est construit sur un sous ensemble de la population, appelé échantillon d'apprentissage noté Ω_a.
- Un individu appartenant à l'échantillon est noté ω.
- L'attribut à prédire Y associe à chaque individu de Ω_a une classe appartenant à C {ensemble des classes $C = \{c_1, ..., c_m\}$.

$$Y : \Omega_a \rightarrow C$$
$$\omega \rightarrow Y(\omega)$$

- Chaque variable exogène (explicative) X_j est définie par:

$$X_j : \Omega_a \rightarrow E_j$$
$$\omega \rightarrow X_j(\omega)$$

E_j : ensemble des modalités (valeurs) de X_j.

Parmi les méthodes de l'apprentissage supervisé, on s'intéresse plus particulièrement aux méthodes à base de graphes d'induction qui sont un cas général de celles à base d'arbres de décision : ID3 (Quinlan, 1986), C4.5 (Quinlan, 1990, 1993), CART (Breiman et al., 1984), et enfin SIPINA (Zighed et al., 1992), (Atmani et Bldjilali, 2007a, 2007b) qui est une méthode à base de graphe d'induction.

Méthodes à base de graphes d'induction. Les modèles de classification générés par ces méthodes sont des graphes ; des arbres contenant des cycles, un graphe est constitué de plusieurs niveaux appelés partitions S_i, une partition est composée de nœuds ou sommets s_j qui sont reliés par des arcs.
Il existe trois types de sommets, un sommet initial appelé aussi racine de l'arbre, des sommets intermédiaires et des feuilles.
La génération du graphe consiste à la génération de partitions, une partition représente un ensemble de feuilles à un instant t, le passage de S_i à S_{i+1} se fait en optimisant un critère de variation d'incertitude $\tau(S_{i+1}) = \tau(S_i) - \tau(S_{i+1})$.
Dans la figure 2, ci-dessous, est présenté le graphe, correspondant à l'exemple du tableau 1, généré par SIPINA. A partir d'un arbre ou d'un graphe, il est possible d'extraire des règles (Rabaseda et al., 1995, 1996b) de la forme :

Si <Conditions > Alors < Classe= valeur >.

Où Condition est une expression logique sous forme normale conjonctive pour les arbres de décision et sous forme conjonctive disjonctive pour les graphes d'induction, et Conclusion la classe majoritaire dans le sommet atteint par la condition.
A partir du graphe de la figure 2, on peut extraire 4 règles correspondant aux feuilles :
R1) SI (Temps= 'Pluvieux' ET Taille-ville='Grande') OU Temps='Neigeux' ALORS Conditions='Mauvaises'.
R2) SI ((Temps='Pluvieux' ET Taille-ville='Petite') OU Temps='Beau') ET (Heures='Creuses' OU Heures='Soir') ALORS Conditions='Bonnes'.

R3) SI ((Temps='Pluvieux' ET Taille-ville='Petite') OU Temps='Beau') ET Heures='Sortie' ET Jour-Férié='Non' ALORS Conditions='Mauvaises'.
R4) SI ((Temps='Pluvieux' ET Taille-ville='Petite') OU Temps='Beau') ET Heures='Sortie' ET Jour-Férié='Oui' ALORS Conditions='Bonnes'.

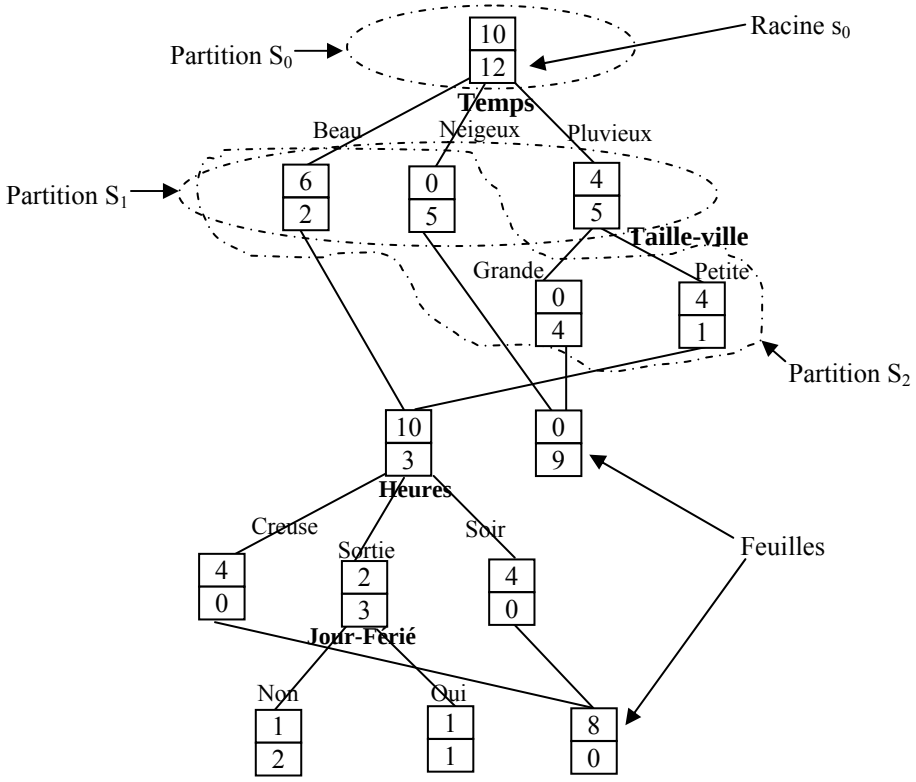


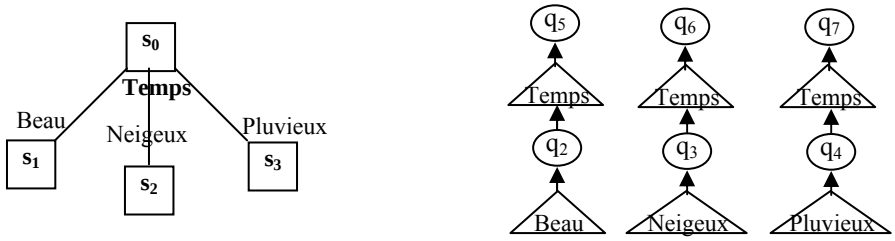
FIG. 2 – *Graphe d'induction de l'échantillon du tableau 1 généré par SIPINA.*

3.2.2 Transformation d'un graphe d'induction dans le formalisme d'automates d'arbres

L'automate d'arbres calculé A est défini par le quadruplet (Q,V, Δ,Q_f) tel que : V est un élément de Σ_S et Δ est un ensemble qui contient les règles de transition δ, son calcul est réalisé au fur et à mesure de la génération du graphe. L'automate est de type ascendant ; ses règles sont de la forme $f(q_1,q_2,\ldots,q_s) \rightarrow q$ tel que q est un état final, ceci se justifie par le fait qu'à un instant t les derniers nœuds générés sont considérés comme des feuilles de l'arbre donc le parcours de l'arbre par l'automate va se faire des feuilles considérées à un instant t vers la racine.

Pour la construction d'un graphe d'induction, il existe deux opérations de base que sont éclatement et fusion (Zighed et al., 1992). Pour la transformation du graphe nous avons deux types d'éclatements à considérer et à traduire dans le formalisme d'automate d'arbres :
Interprétation de l'éclatement.

- Eclatement du sommet initial s_0



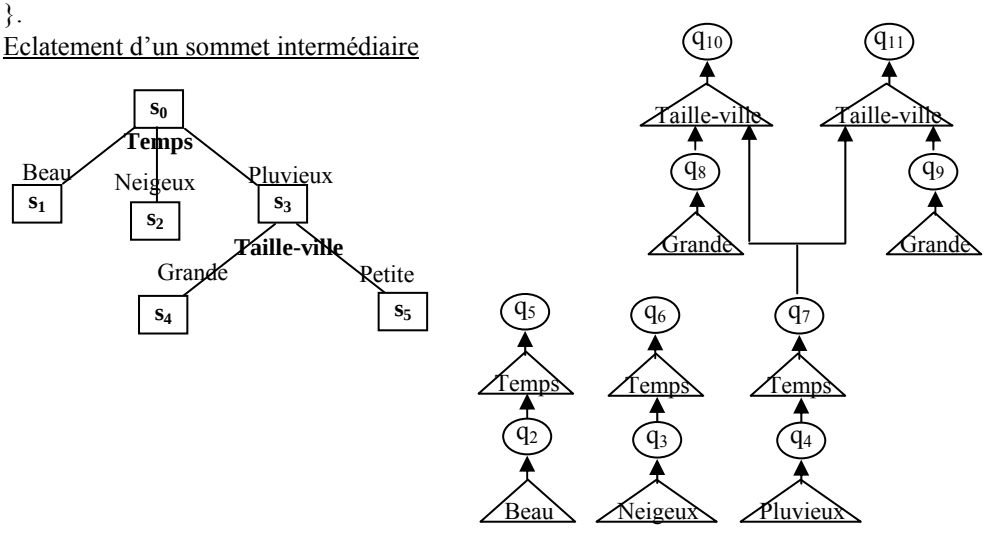
(1) **(2)**

FIG. 3 – (1) *Graphe du tableau 1 généré à un instant t_1.* (2) *Automate d'arbres correspondant au graphe* FIG.3-1.

L'automate d'arbre A = (Q, V, Δ, Q_f) correspondant à l'arbre généré à un instant t_1 est composé de :

- Q_f= {q_0, q_1}, Q_f contient initialement les états q_0 correspondant à la modalité 'Mauvaises' de la classe conditions et q_1 correspondant à la modalité 'Bonnes',
- Q=Q_f ∪ {q_2, q_3, q_4, q_5, q_6, q_7},
- V=V∪ {Ciel (), Ensoleillé, Couvert, Pluie}, Ciel() un symbole d'arité 1 et Pluie d'arité 0
- Δ={Beau→q_2,Neigeux→q_3,Pluvieux→q_4,Temps(q_2) →q_5,Temps(q_3) →q_6,Temps(q_4) →q_7 }.

- Eclatement d'un sommet intermédiaire



(1) **(2)**

FIG. 4 – (1) *Graphe du tableau 1 généré à un instant t_2.* (2) *Automate d'arbres correspondant au graphe* FIG.4-1.

L'automate d'arbre A = (Q, V, Δ, Q_f) correspondant à l'arbre généré à un instant t_2 est composé de :

- Q=Q∪ {q_8, q_9, q_{10}, q_{11}},
- V=V∪ {Taille-ville (,), Grande, Petite}, Taille-ville (,) désigne que Taille-ville est un symbole fonctionnel d'artié 2,
- Δ= Δ ∪ {Grande→q_8, Petite→q_9, Taille-ville(q_8,q_7)→q_{10}, Taille-ville(q_9,q_7)→q_{11}}.

Interprétation de la fusion. L'automate d'arbre A = (Q, V, Δ, Q*f*) correspondant au graphe généré à un instant t3 est composé de :

– Q=Q ∪ {q_{12}},
– V=V ∪ {Ou (,)}, Ou (,) est un symbole fonctionnel d'artié 2 qui est utilisé chaque fois qu'on rencontre un regroupement, Δ= Δ ∪ {Ou (q_{10},q_6)→q_{12}}.

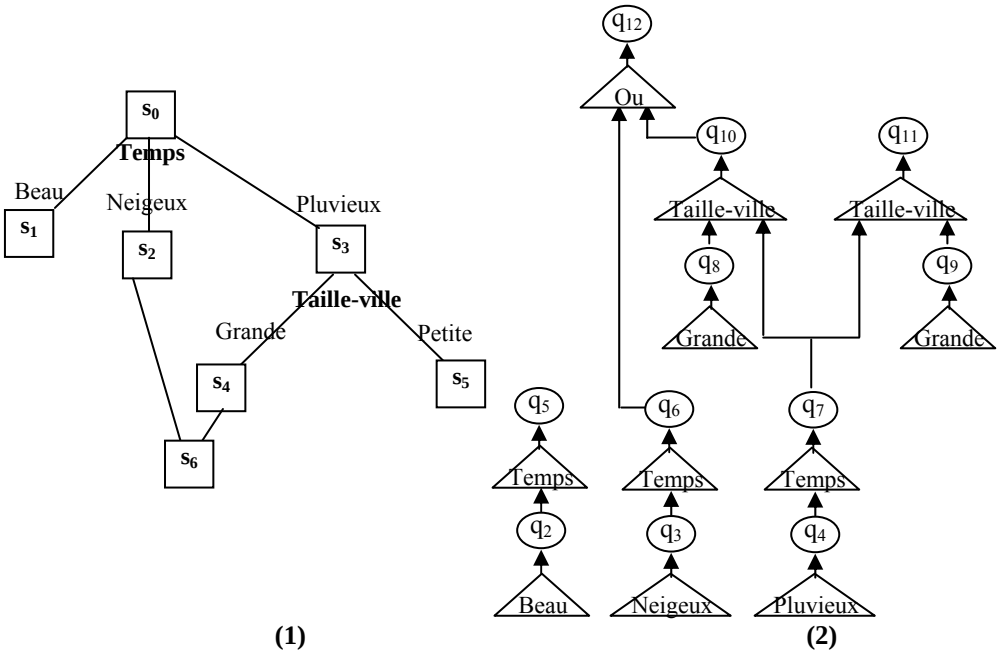


FIG. 5 – (1) *Graphe du tableau 1 généré à un instant t_3.* (2) *Automate d'arbres correspondant au graphe* FIG. 5-1.

L'automate d'arbre correspondant à l'arbre complet présenté dans la figure 2 est illustré par la figure 6 présentée dans la page suivante de cet article.

Extraction des règles de l'automate par substitution. En fin d'apprentissage, les composants Q, V, Δ, Q_f de l'automate A seront composés de :

– Q_f= {q_0, q_1},
– Q= { $q_{i(i=0,24)}$},
– V={Temps(),Taille-ville(,),Heures(,),Jour-Férié(,),Ou(,)},
– Δ={Beau→q_2, Neigeux→q_3, Pluvieux→q_4, Temps(q_2)→q_5, Temps(q_3)→q_6, Temps(q_4)→q_7, Grande→q_8, Petite→q_9, Taille-ville(q_8,q_7)→q_{10}, Taille-ville(q_9,q_7)→q_{11}, Ou(q_6,q_{10})→q_{12}, Ou(q_5,q_{11})→q_{13}, Soir→q_{14}, Sortie→q_{15}, Creuse→q_{16}, Heure(q_{13},q_{14})→q_{17}, Heure(q_{13},q_{15})→q_{18}, Heure(q_{13},q_{16})→q_{19}, Ou(q_{17},q_{19})→q_{20}, Non→q_{21}, Oui→q_{22}, Jour-Férié(q_{21},q_{18})→q_{23}, Jour-Férié(q_{22},q_{18})→q_{24}, q_{20}→q_1, q_{12}→q_0, q_{23}→q_0, q_{24}→q_1}.

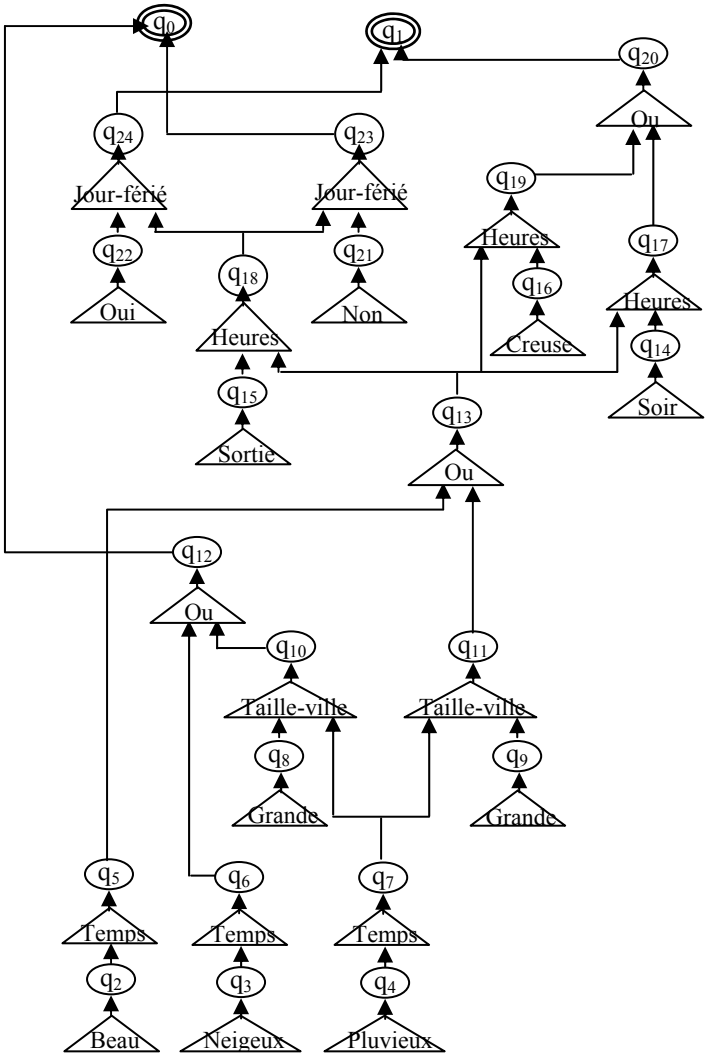


FIG. 6 – *Automate d'arbre correspondant au graphe complet.*

Les règles du graphe d'induction correspondent au règles-ε de l'ensemble Δ, ces règles sont au nombre de 4 :{ $q_{20}\rightarrow q_1$, $q_{12}\rightarrow q_0$, $q_{23}\rightarrow q_0$, $q_{24}\rightarrow q_1$}.

Les règles-ε de l'ensemble Δ se présentent sous la forme « Prémisse → état final ». On substitue les états de chaque règle par les valeurs qui leur correspondent dans Δ et on réitère ce processus jusqu'à ce que la règle ne contienne que des symboles fonctionnels. Le problème consiste à mettre la prémisse sous forme de disjonction de conjonction, pour cela nous donnons trois règles de réécriture des prémisses selon leurs formes.

Les règles par substitution sont calculées comme suit:

- $Ou(q_6,q_{10})\rightarrow q_0$ qui correspond à successivement à :
  $Ou(Temps(q_3),Taille\text{-}ville(q_8,q_7))\rightarrow q_0$,
  $Ou(Temps(Neigeux),Taille\text{-}ville(Grande,Temps(q_4)))\rightarrow q_0$,

Ou(Temps(Neigeux),Taille-ville(Grande,Temps(Pluvieux))) $\rightarrow q_0$

Ce qui correspond à : SI (Temps= 'Pluvieux' ET Taille-ville='Grande') OU Temps='Neigeux' ALORS Conditions='Mauvaises' **<=> R1)**.

- Ou (q_{17},q_{19})$\rightarrow q_1$ correspondant à la règle : SI (Heure='Soir' ET (Temps='Beau' Ou (Taille-ville='Petite ET Temps='Pluvieux'))) Ou (Heure='Creuse' ET (Temps='Beau' Ou (Taille-ville='Petite ET Temps='Pluvieux'))) ALORS Conditions='Bonnes' **<=> R2)**.
- Jour-Férié(q_{21},q_{18})$\rightarrow q_0$ correspondant à la règle de décision: SI Jour-Férié='Non' ET Heure='Sortie' ET (Temps='Beau' Ou (Taille-ville='Petite' ET Temps='Pluvieux')) ALORS Conditions='Mauvaises' **<=> R3)**.
- Jour-Férié(q_{22},q_{18})$\rightarrow q_1$ correspondant à règle: SI Jour-Férié='Oui' ET Heure='Sortie' ET (Temps='Beau' Ou (Taille-ville='Petite ET Temps='Pluvieux')) ALORS Conditions='Bonnes' **<=> R4)**.

3.3 Simplification du modèle d'automate d'arbre généré

Théorème

L un langage reconnu par un automate d'arbres avec règles-ε alors L est reconnu par un automate d'arbres sans règles-ε (voir Comon et al., 2005). Donc l'automate de la figure 6 peut être optimisé en supprimant les règles-ε q→q' et renommant les état q par les q' et ceci conduit à supprimer tout les états q (états reliés à des états finaux par des règles-ε).

Les éléments Q, Δ de l'automate A vont être réduits en taille en appliquant le théorème cité précédemment :

– $\Delta = \Delta - \{ q_{20} \rightarrow q_1, q_{12} \rightarrow q_0, q_{23} \rightarrow q_0, q_{24} \rightarrow q_1 \}$, les états q_{20}, q_{24} se transforment q_1 et q_{12}, q_{23} vont se transformer en q_0.

Δ={Beau$\rightarrow q_2$, Neigeux$\rightarrow q_3$, Pluvieux$\rightarrow q_4$, Temps(q_2)$\rightarrow q_5$, Temps(q_3)$\rightarrow q_6$, Temps(q_4)$\rightarrow q_7$, Grande$\rightarrow q_8$, Petite$\rightarrow q_9$, Taille-Ville(q_8,q_7)$\rightarrow q_{10}$, Taille-Ville(q_9,q_7)$\rightarrow q_{11}$, Ou(q_6,q_{10})$\rightarrow q_0$, Ou(q_5,q_{11})$\rightarrow q_{13}$, Soir$\rightarrow q_{14}$, Sortie$\rightarrow q_{15}$, Creuse$\rightarrow q_{16}$, Heure(q_{13},q_{14})$\rightarrow q_{17}$, Heure(q_{13},q_{15})$\rightarrow q_{18}$, Heure(q_{13},q_{16})$\rightarrow q_{19}$, Ou(q_{17},q_{19})$\rightarrow q_1$, Non$\rightarrow q_{21}$, Oui$\rightarrow q_{22}$, Jour-Férié(q_{21},q_{18})$\rightarrow q_0$, Jour-Férié(q_{22},q_{18})$\rightarrow q_1$}, $|\Delta| = 23$ alors qu'on avait $|\Delta| = 27$,

– $Q = Q - \{q_{20}, q_{12}, q_{23}, q_{24}\}$, $|Q| = 20$ alors qu'on avait $|Q| = 24$.

4 Conclusion

L'objectif de cet article est de présenter la problématique de notre travail et qui concerne la conception et l'expérimentation d'une nouvelle technique de génération et d'optimisation de graphes d'induction par une méthode formelle de modélisation : les automates d'arbres. Ce cadre formel d'automates d'arbres présente aujourd'hui une approche puissante pour la simplification et l'optimisation (Comon et al., 2005). Nous nous sommes fixés comme objectifs l'expérimentation des algorithmes de minimalisation d'automates d'arbres pour simplifier les modèles de classification générés par des techniques à base d'arbres de décisions et de graphes d'induction : ID3, C4.5, CART, SIPINA, etc. Notre première perspective concerne l'intégration de la simplification des bases de règles dans une interface d'acquisition automatique des connaissances à partir des données. Cette simplification a pour objective la détection des incohérences dans les bases de règles et l'élimination des variables exogènes non pertinentes. Comme deuxième perspective nous proposons d'utiliser la classe

d'automates d'arbres stochastiques pour pouvoir représenter la notion d'effectif dans un sommet.

References

Atmani, B., B. Beldjilali (2007a). Neuro-IG : A Hybrid System for Selection and Elimination of Predictor Variables and non Relevant Individuals. *Informatica, Journal International*, Vol. 18, N°2 163-186.

Atmani, B., B. Beldjilali (2007b). Knowledge Discovery in Database: Induction Graph and Cellular Automaton. *Computing and Informatics Journal*, Vol.26, N°2 171-197.

Bernard, M., C. De La Higuera (1999). *GIFT "Grammatical Inference For Terms"*. Late Breaking paper at the 9th International Conference on Inductive Logic Programming.

Bernard, M., C. De La Higuera (2001). Apprentissage de programmes logiques par inférence grammaticale. *Revue d'Intelligence Artificielle*, 14(3/4), 375–396.

Bouchou, B., M. Halfeld Ferrari Alves, and M. Musicante (2003a). Tree automata to verify XML key constraints. *6th International Workshop on the Web and Data Bases WebDB*, 37-43.

Bouchou, B., M. Halfeld Ferrari Alves (2003b). Updates and Incremental Validation of XML Documents. *9th Int. Conference on Database Programming Languages*, 216-232.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification And Regression Trees*. New York: Chapman and Hall.

Carme, J., R. Gilleron, A. Lemay and J. Niehren (2006). *Interactive learning of node selecting tree transducers*. Machine Learning.

Carrasco, R., J. Oncina and J. Calera-Rubio (2001). Stochastic Inference of Regular Tree Languages. *Machine Learning*, 44(1/2), 185–197.

Comon, H., M. Dauchet, R. Gilleron, F. Jacquemard, D. Leguiez, S. Tison and M. Tommasi (2005). *Tree Automata Techniques and applications*.

Dal Zilio, S., D. Lugiez (2002a). *Fondements de l'interrogation de données semi structurées*. Laboratoire d'Informatique Fondamentale de Marseille, CNRS, UMR 6166, ATIP Jeunes chercheurs.

Dal Zilio, S., D. Lugiez (2002b). *XML Schema, Tree Logic and Sheaves Automata*. Rapport technique, INRIA.

Dal Zilio, S., D. Lugiez (2003). *XML Schema, Tree Logic and Sheaves Automata*. In Proc. of RTA - Rewriting Techniques and Applications, LNCS 2706.

Dal Zilio, S., L. Acciai (2004). Pattern matching et documents XML « Un nouvel opérateur de filtrage pour les documents XML ».

Dal Zilio, S., D. Lugiez, C. Meyssonnier (2004). Logic you Can Count On. *In Proc. of the 31st ACM SIGPLAN-SIGACT symposium on Principle of Programming Languages*, 135-146.

Denis, F., R. Gilleron (1997). *Apprentissage à partir d'exemples*. Université Charles de Gaulles, Lilles.

Genet, T., F. Klay (2000). *Rewriting for Cryptographic Protocol*. Verification in Proceedings 17th International Conference on Automated Deduction, série Lecture Notes in Artificial Intelligence, volume 1831, Berlin: Springer-Vergla.

Genet, T., V. Viet Tiem Tong (2001). Reachability Analysis of Term Rewriting Systems with Timbuk. *In Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, série Lecture Notes in Artificial Intelligence volume 2250. Berlin: Springer-Verlag*, 691–702

Genet, T., Y. M. Tang-Talpin and V. Viet Triem Tong (2003). *Verification of copy-protection cryptographic protocol using approximations of term rewriting systems*. In Proc. Of WITS'03, Workshop on Issues in the Theory of Security.

Goubault-Larrecq, J. (2000). A method for automatic cryptographic protocol verification (extended abstract). *In Proceedings of the International Workshop on Formal Methods in Parallel Programming, Techniques and Applications, volume 1800. Springer Verlag Lecture Notes in Computer Science*, 977–984.

Habrard, A., M. Bernard and F. Jacquenet (2002). Apprentissage d'automates d'arbres stochastiques généralisés à partir de bases de données relationnelles. *In Conférence d'Apprentissage, Orléans, France. Presses Universitaires de Grenoble*, 180–191.

Habrard, A. (2004). *Modèles et techniques en inférence grammaticale probabiliste : de la gestion du bruit à l'extraction de connaissances*. Thèse de doctorat, Université Jean Monnet de Saint-Etienne.

Monniaux, D. (1999). *Abstracting Cryptographic Protocols with Tree Automata*. Static Analysis Symposium.

Oncina, J., P. Garcia (1992). Inferring regular languages in polynomial update time. *In Pattern Recognition and Image Analysis,* 49-61.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning* 1, 81–106.

Quinlan, J.R. (1990). *Probabilistic decision trees*. In Machine Learning: An Artificial Intelligence Approach, Volume 3. San Mateo: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann.

Rabaseda, S., R. Rakotomalala and M. Sebban (1995). Génération automatique de connaissances par induction. *Actes des 3èmes rencontres de la société francophone de classification*, 45-46.

Rabaseda, S. (1996a). *Contribution à l'extraction automatique de connaissances : Application à l'analyse clinique de la marche*. Thèse de Doctorat de l'Université Lyon 1.

Rabaseda, S., R. Rakotomalala and D. A. Zighed (1996b). Rules extracted automatically by induction. *Proceeding of the 6th conference on information processing and management of uncertainty*, 551-556.

Taleb Zouggar, S., B. Atmani (2008). *Vers la conception d'un modèle de classification par automates d'arbres*. International Conference on Web and Information Trechnologies. 29-30 juin 2008, Sidi Bel Abbes, Algérie.

Tommasi, M. (2006). *Structures arborescentes et apprentissage automatique*. Habilitation à diriger des recherches, Université de Lilles 3, France.

Zighed, D. A., J. P. Auray and G. Duru (1992). *SIPINA : Méthode et Logiciel*. Lacassagne.

Summary

In the automatic learning field several methods dedicated to the classification task were focusing. We are particularly interested in the symbolic techniques and more exactly in the methods based on induction graphs. In this paper we propose a structured approach for induction graphs generation by using a formalism having showed its ability in several domains of the computing: trees automata. Our contribution in this domain concerns the post-pruning of the classification models based on induction graphs. We settled as objective the experiment of the minimalisation algorithms of trees automata to optimize the classification models generation.

Une approche parallèle distribuée pour la génération des motifs fermés fréquents basée sur une infrastructure CORBA

Abdelfettah Idri, Azedine Boulmakoul

Département Informatique, Laboratoire Informatique de Mohammedia
Faculté des Sciences et Techniques de Mohammedia, Maroc
abdelfattah_id@yahoo.com
azedine.boulmakoul@yahoo.fr

Résumé. La technique d'extraction des règles d'association est un problème connu dans le domaine de la fouille de données (data mining) et celui de la découverte de la connaissance (knowledge discovery). Dans ce processus, l'étape nécessitant le plus de calcul et par conséquent de temps est celle de la génération des motifs fermés fréquents. Du moment que la fouille de données est fortement liée à l'analyse formelle des concepts, on se base dans ce papier sur le treillis de Galois comme élément central pour la génération des motifs fermés fréquents. Lorsque appliqué dans le domaine de la fouille de données comme dans notre cas, le processus de génération de Treillis de Galois (Concept Lattice) s'accompagne souvent d'une complexité exponentielle aussi bien d'espace que du temps. Il existe plusieurs algorithmes standard pour la construction du Treillis de Galois d'une relation binaire. Dans ce papier, notre proposition s'adresse à l'aspect parallèle distribué de l'algorithme de construction du treillis de Galois afin d'améliorer les performances de l'algorithme séquentiel. L'architecture du système et l'algorithme de génération du treillis de Galois seront exposés. L'extraction des règles d'association est basée sur l'exploitation du treillis de Galois généré en adoptant la notion du générateur minimal des règles d'association inspiré du modèle mirage de Zaki.

1 Introduction

La prospection de données est l'activité d'extraire toute information (connaissance) utile depuis des entrepôts de données volumineux dans l'objectif de transformer ces informations en des faits aidant à la prise de décision. L'analyse des concepts formels (FCA) représente en fait un socle pour la prospection de données du moment que ces deux domaines sont étroitement liés. Dans ce papier, on s'intéresse au Treillis de Galois qui est à la base de la génération des motifs fermés fréquents. Aussi, on s'intéresse à la génération des règles d'association basée sur le Treillis de Galois. Notre approche s'inscrit dans l'optique d'améliorer les performances de l'algorithme séquentiel de génération de Treillis de Galois en préconisant une approche parallèle distribuée.

La construction de treillis de Galois a fait l'objet de plusieurs recherches, spécialement dans les domaines d'analyse de concepts formels d'une part Ganter et Wille (1999), Bordat (1986), Chein (1969) et la fouille de données d'autre part Zaki et Ogihara (1998), Pasquier et al. (1999). Depuis leur apparition, l'analyse des concepts formels et la fouille de données

trouvent leur voie d'application dans plusieurs domaines, surtout ceux manipulant des bases de données volumineuses.

Généralement dans le domaine d'analyse de concepts formels, les données sont formulées sous forme de contexte. Un contexte est constitué d'un triplet (O, M, I) où O représente l'ensemble des objets, M l'ensemble des attributs et I une relation binaire entre O et M. Sur la base de ce contexte, un ensemble de concepts peut être construit. Lorsque ce dernier satisfait une relation d'ordre partiel, on parle alors de treillis de Galois ou de treillis de concepts Barbut et Montjardet (1970).

Par ailleurs, il est important de considérer la relation entre les treillis de Galois et la prospection de données. En fait il existe une correspondance bijective entre les treillis de Galois et les motifs fermés du moment que l'intention du concept coïncide avec la notion de motif fermé Zaki et Ogihara (1998). D'où le besoin énorme d'algorithmes de construction de treillis de Galois, puisque la résolution du problème dans l'analyse formelle des concepts peut directement servir dans la prospection de données.

Dans ce papier, on présente un algorithme parallèle pour la construction de treillis de Galois en se basant sur les mêmes techniques des algorithmes séquentiels tels que celui de Bordat (1986) et Choi (2006). Nous présentons également l'architecture du système supportant cet algorithme ainsi que son implémentation. Ce document est organisé comme suit. Le paragraphe 2 rappelle la théorie et la terminologie des treillis de Galois. Ensuite nous verrons dans le paragraphe 3 l'architecture du système. Le paragraphe 4 aborde l'algorithme parallèle. Le paragraphe 5 aborde l'implémentation de l'algorithme et expose les résultats de cette approche. Dans le paragraphe 6 on traite les règles d'association. On conclut dans le paragraphe 7 avec nos suggestions et recommandations.

2 Analyse formelle des concepts, théorie et terminologie

L'analyse formelle des concepts (ou FCA) est un domaine de recherche vaste et elle est dérivée de la théorie des treillis basée sur la notion de concepts. FCA s'intéresse à la construction des treillis de concepts fournissant ainsi un outil efficace pour la fouille de données et la génération des règles d'associations. Dans la suite du paragraphe on aborde les notions de base de FCA.

2.1 Définitions

Définition 1 Contexte : Dans FCA, on nomme un contexte le triplet (O, M, I), où $O = \{g_1, g_2, ..., g_n\}$ désigne un ensemble de n éléments appelés objets ; $M = \{1,2,...,m\}$ désigne un ensemble de m éléments appelés attributs et $I \subseteq O \times M$ la relation binaire entre les objets et les attributs.

Le contexte est représenté souvent sous forme d'un tableau dont les objets sont en ligne et les attributs sont en colonne comme le montre le **Erreur ! Source du renvoi introuvable.**. On appelle ensemble d'objets un sous-ensemble $X \subseteq O$. De même, on appelle un ensemble d'attributs un sous-ensemble $J \subseteq M$. Par convention, on écrit un ensemble d'objet $\{b, d, e\}$ sous la forme bde, et un ensemble d'attribut $\{3,4,6\}$ sous la forme 346.

Définition 2 Listes adjacentes : L'ensemble des objets communs d'un élément $i \in M$ est défini par $nbr(i) = \{g \in O : (g,i) \in I\}$ et appelé liste adjacente de i. L'ensemble des attributs communs d'un élément $g \in O$ est défini par $nbr(g) = \{i \in M : (g,i) \in I\}$ et appelé liste adjacente de g. Dans le tableau 1, on peut lire $nbr(a) = \{1,6\}$ et $nbr(1) = \{a,b,c\}$.

I	1	2	3	4	5	6	7
a	x					x	
b	x		x	x	x	x	
c	x			x		x	
d		x	x		x		
e		x					x

TAB. 1 – *Exemple de contexte* (O,M,I) *avec* $O = \{a,b,c,d,e\}$ *et* $M = \{1,2,3,4,5,6,7\}$ *. Le tableau représente la relation binaire I.*

Définition 3 Les fonctions attr et obj : La fonction $attr : 2^O \rightarrow 2^M$ fait correspondre à un ensemble d'objets donnés leurs attributs communs : $attr(X) = \bigcap_{g \in X} nbr(g)$ avec $X \subseteq O$. De la même façon, la fonction $obj : 2^M \rightarrow 2^O$ fait correspondre à un ensemble d'attributs donnés leurs objets communs : $obj(J) = \bigcap_{j \in J} nbr(j)$ avec $J \subseteq M$.

Définition 4 La fermeture des ensembles : Un ensemble d'objets $X \subseteq O$ est fermé si $X = obj(attr(X))$. Un ensemble d'attributs $J \subseteq M$ est fermé si $J = attr(obj(J))$.

Dans **Erreur ! Source du renvoi introuvable.**, l'ensemble $X = abc$ est fermé puisque $obj(attr(X)) = abc$. $attr(abc) = 16$ et $obj(16) = abc$.

Définition 5 Concepts : Un concept est un couple de la forme $C = (X, J)$ avec $X \subseteq O$ et $J \subseteq M$ dans lequel $X = obj(J)$ et $J = attr(X)$.

L'ensemble X est nommé l'extension (extent) de C et noté $X = ext(C)$. L'ensemble J est nommé l'intention (intent) de C et noté $J = \text{int}(C)$. Par définition, X et J sont tous les deux fermés. L'ensemble de tous les concepts du contexte (O,M,I) est noté par $B(O,M,I)$ ou B. La relation d'ordre définie sur B de la manière suivante :

$$(A_1, B_1) \prec (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (B_2 \subseteq B_1)$$

Où (A_1, B_1) et (A_2, B_2) sont deux concepts de B, est une relation d'ordre partielle sur B.

Définition 6 treillis de Galois : $L =< B, \prec >$ est un treillis de concepts (Galois) complet.

2.2 L'algorithme séquentiel

Construire un treillis de Galois revient à générer tous les concepts en identifiant les successeurs de chacun d'entre eux. L'idée principale d'un algorithme séquentiel est de commencer par le concept parent $(O, attr(O))$ et de générer ensuite tous ses successeurs. D'une manière récursive, on génère chacun de ces successeurs selon l'algorithme de parcours en largeur (BFS : Breadth First Search). La figure ci-dessous montre l'architecture de l'algorithme séquentiel. Il s'agit de trois étapes principales :

- La préparation de données
- La génération du treillis de Galois en utilisant un trie global et un trie local
  - La génération des concepts enfants
  - Le test de la fermeture d'un concept candidat
  - Le test d'existence d'un concept
- La visualisation du treillis

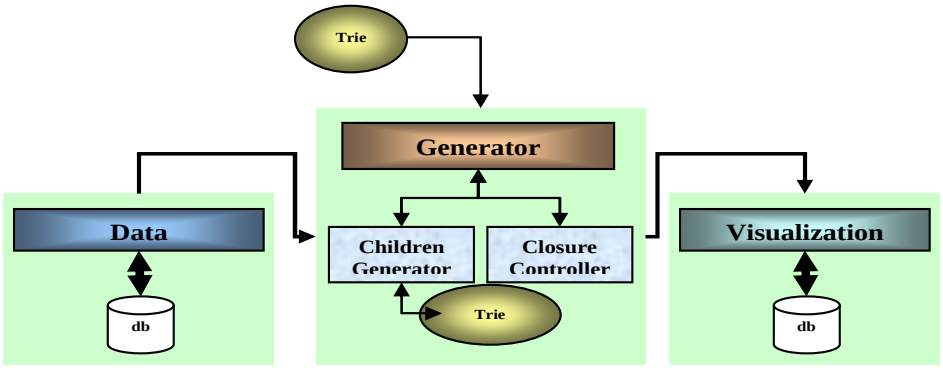
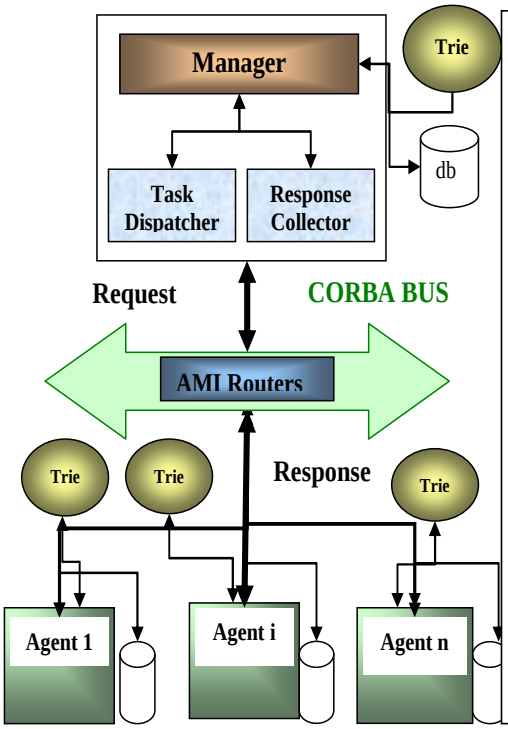


FIG. 1 – *Architecture de l'algorithme séquentiel.*

3 Architecture du système cible

En général, le nombre de concepts issu d'un contexte donné est exponentiel par rapport à la taille des données initiales. Par conséquent, la génération des concepts (treillis de Galois) peut devenir très coûteuse en terme de complexité temporelle et spatiale. De ce fait, on s'est penché sur l'étude de possibilités pour améliorer les performances du processus de construction du concept Galois en s'intéressant à l'aspect distribution et parallélisme d'exécution de l'algorithme.

3.1 Architecture



Algorithme parallèle Manager

```
1: Initialise concept queue Q= {C};  C = (O, attr(O))
2: Initialise Context, Agenda, Trie
5: Initialise and Synchronise Agents
6: while Q is not empty or Agenda is not empty
                    or waiting for reponses do
7:     while Q is not empty do
8:        C = dequeue(Q)
9:        Enqueue C to Agenda
10:    end while
11:    for each available Agenti do
12:      if Agenda is not empty then
13:         C = dequeue(Agenda)
14:         Send_request(generateChildren(C)) to Agenti
15:         Mark Agenti as busy
16:      end if
17:    end for
18:    for each buzy agenti do
19:      if response of agenti is available then
20:            get response
21:            for each child in response do
22:               if child not exists in Trie then
23:                  insert child into Trie
24:                  Identify child as successor of C
25:                  Enqueue child to Q
26:               else
27:                  Identify child as successor of C
28:               end if
29:            end for
30:            mark agenti as free
31:      end if
32:    end for
33: end while
```

FIG. 2 – *Architecture du système et Algorithme.*

L'architecture proposée dans le schéma ci-dessus est constituée de trois composantes principales dans l'objectif d'améliorer la performance de l'algorithme séquentiel (voir **Erreur ! Source du renvoi introuvable.**) comme illustré dans le paragraphe 3.2 :

- Le Manager : celui-ci utilise d'une part un Trie global pour gérer les concepts constituant le treillis de Galois. D'autre part, le Manager repose sur deux modules pour assurer la communication avec les agents, notamment le Dispatcher et le Collecteur.
  - Dispatcher : distribue les tâches aux Agents. Une tâche comprend en fait la génération des concepts enfants d'un concept donné.
  - Collecteur : collecte les résultats et les transmet au Manager. Un résultat est constitué d'une liste de concepts.
- Les Agent : l'Agent est responsable de la génération des concepts enfants en utilisant un Trie local.
- La communication : elle est assurée par le biais d'une infrastructure CORBA basée sur un routeur AMI (*Asynchronous Method Invocation*).

3.2 Motivation

La démarche globale adoptée pour la conception de cette architecture est décrite dans ce qui suit. La première phase a été consacrée à identifier les actions indépendantes de l'algorithme qui peuvent participer à la réduction du temps d'exécution et l'optimisation de l'espace. Dans la deuxième phase, on doit vérifier si ces actions sont dissociables sans alourdir la communication entre elles. Finalement, il reste à étudier les possibilités d'implémentation de l'architecture. En analysant des algorithmes existants Bordat (1986), Choi (2006) et Ganter et Reuter (1991), on a pu distinguer les actions suivantes :

- La génération des enfants d'un concept.
- Le contrôle de fermeture d'un ensemble.
- Le contrôle d'existence d'un concept.

Le choix a été fait sur le modèle Manager/Agent puisqu'il garantit la scalabilité et la distribution des services et ceci coïncide bien avec notre objectif.

La génération des enfants d'un concept est un processus complexe et utilise un algorithme spécial ainsi qu'un arbre local. Cette tâche peut être déléguée aux Agents puisqu'elle peut s'exécuter d'une manière totalement indépendante. La multiplication du nombre d'agents implique directement la réduction du temps d'exécution et permet d'éviter les piques de mémoires pendant la génération des enfants. Par ailleurs ceci exige une implémentation efficace pour le transport des concepts enfants entre le Manager et les Agents.

De même, le contrôle de fermeture d'une intention ou une extension peut être aisément délégué aux Agents.

Par contre, L'existence d'un concept est réalisée à l'aide d'un arbre de codification (Trie). La clé se compose des éléments de l'intention du concept. Cette tâche ne peut pas être totalement déléguée aux Agents puisque l'arbre contient au fur et à mesure tous les concepts générés par tous les Agents et donc il doit être partagé par eux pour pouvoir tester l'existence d'un concept donné. C'est donc le Manager qui prend en charge la gestion de l'arbre.

Le Manager utilise un dispatcher pour distribuer les tâches aux Agents et un collecteur pour collecter les résultats envoyés par ces derniers. On a choisi CORBA pour la communication entre tous les acteurs de cette architecture. L'utilisation de CORBA nous permet d'une part de cacher la complexité des structures de données utilisées dans l'algorithme. D'autre part, CORBA offre des mécanismes de programmation évolués tel que la gestion des événements distribués, le support de la communication asynchrone (AMI) et la programmation orienté objet.

Les services offerts par le Manager et les Agents sont listés ci-dessous.

Manager :

- Gestion de l'arbre (insertion d'un concept, contrôle de l'existence d'un concept)
- Gestion des tâches (distribution, collection, synchronisation)

Agent :

- Génération des concepts enfants
- Contrôle de fermeture de l'intention ou l'extension d'un concept

4 L'algorithme parallèle de construction de treillis de Galois

La **Erreur ! Source du renvoi introuvable.** expose l'algorithme du système. On explique dans ce qui suit les principales étapes de l'approche adoptée.

Selon notre schéma proposé, la construction du treillis de Galois est réalisée dans deux phases principales réparties sur le Manager et les Agents.

Première phase :

Tout d'abord, l'Agent s'occupe de la génération des concepts enfants candidats d'un concept donné. Ensuite, l'Agent applique simultanément la fermeture au résultat obtenu de façon à n'envoyer au Manager qu'un ensemble de concepts déjà traité.

Deuxième phase :

Le Manager envoie progressivement les concepts disponibles dans l'Agenda (une file de concepts) aux Agents sélectionnés par le Dispatcher. En retour, le collecteur reçoit les réponses des Agents sous forme d'ensembles de concepts représentant les concepts enfants des concepts envoyés. Le Manager procède alors à la mise à jour de l'arbre des concepts : soit par insertion du concept enfant et connexion avec son concept parent ; soit par connexion seulement de ces deux concepts en cas d'existence préalable du concept enfant dans l'arbre. Ce processus est répété jusqu'au traitement de tous les concepts dans la file des concepts.

5 Implémentation et résultats

Dans ce paragraphe, nous traitons les aspects d'implémentation de l'algorithme dans la première section. Notre implémentation est applicable aussi bien dans le domaine de l'analyse formelle des concepts que dans le domaine de la fouille de données. On expose nos résultats dans la deuxième section.

Environnement de travail

Pour l'environnement de travail et de test on a utilisé la configuration suivante :

- Plateforme : Windows XP, C++ de Visual Studio
- Communication : CORBA de Orbacus 4.3
- Performance machine : centrino avec un processeur de 2,26 GH et une mémoire de 1 GB
- Comme outil de visualisation du treillis nous avons utilisé Galicia 3.2

5.1 Implémentation

La première section aborde les données d'entrée et de sortie. La deuxième section traite les structures de données principales utilisées dans l'algorithme. Dans la troisième section on discutera la communication entre le Manager et les Agents ainsi que l'outil CORBA.

La figure ci-dessous expose le diagramme de collaboration (UML) reflétant les relations entre les différentes classes. On a choisi celui de la classe *cchildren_impl* (l'implémentation de l'interface CORBA côté agent) du fait qu'il est le plus significatif et il couvre la majorité des classes qu'on désire aborder. La signification de ces dernières est la suivante :

- *cchildren_impl et POA_lattice::cchildren* : implémentent les fonctionnalités de l'Agent intégrant l'interface CORBA
- *attributeSet* : implémente l'ensemble des attributs (l'intention d'un concept)
- *objectSet* : implémente l'ensemble des objets (l'extension d'un concept)
- *binaryRelation* : implémente la relation binaire entre l'ensemble des attributs et celui des objets (voir les liens vers les deux ensembles)
- *Concept* : implémente le concept et se compose entre autres de l'ensemble des attributs et des objets.
- *Context* : implémente le contexte du treillis de Galois. Il est évident qu'il comporte l'ensemble des attributs, l'ensemble des objets et la relation binaire comme indiqué dans le schéma.
- *Triespr* : implémente le Trie local servant à la génération des concepts enfants. Dans le schéma, cette classe pointe sur *triespr_node* pour exprimer qu'il s'agit d'une structure de donnée récursive (arbre).
- *Lattice* : implémente les fonctionnalités de génération des concepts enfants.

Dans ce qui suit on fera référence aux classes de ce diagramme pour exprimer le lien avec les concepts correspondants.

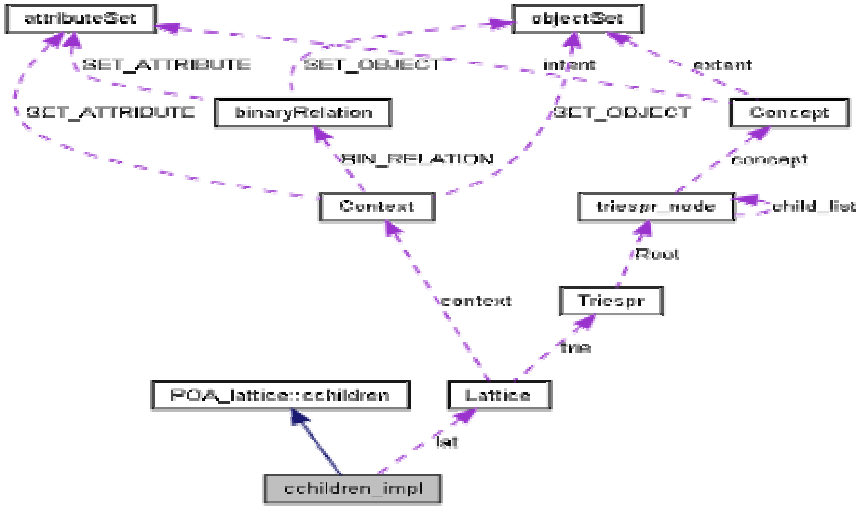


FIG. 3 – *Digramme de collaboration de cchildren_impl*

Les entrées et sorties

On a adopté deux formats pour les données de notre algorithme : le format SLF de Galicia et le format transactionnel matriciel. De même pour les sorties, on génère deux formats : le format GSH-XML de Galicia qui est un fichier XLM et un format interne spécifique pour des fins d'analyse et de recherche. La classe *Context* se charge de la lecture du contexte alors que la classe *Lattice* supporte la génération du format GSH-XML.

Structures de données

En général, pour les structures de données standards telles que les ensembles, les files, les listes et les vecteurs, on utilise la librairie standard de C++ : STL. Dans cette section on aborde spécialement les structures de données les plus spécifiques et relevantes.

Le contexte

Celui-ci est constitué d'un ensemble d'objets, un ensemble d'attributs et une relation binaire, conformément à sa définition originale. La relation binaire comporte sa table de données sous forme d'un ensemble de structure comportant un objet, un attribut et une booléenne indiquant leur relation. Dans le domaine de la fouille de données, les données sont sous forme d'une matrice où la première colonne indique les transactions (objets) et les autres colonnes représentent les items (attributs). Par conséquent, la relation binaire est définie implicitement. Ce sont les classes *Context, binaryRelation, attributeSet et ObjectSet* du diagramme de collaboration qui implémentent ces structures de données.

L'arbre des concepts (Trie)

Quant à l'arbre des concepts, on a adopté une codification lexicographique pour mémoriser les concepts candidats. Un concept candidat est identifié par une clé se composant des éléments de son intention (ou extension). La classe *Triespr* (agent) ou *Trie* (Manager) implémente le Trie (dans ce cas *Triespr* relativement à l'agent).

Le concept

Un concept est constitué d'un ensemble d'attributs (extension), un ensemble d'objets (intention), une identité unique, une liste des identités des parents et une liste des identités des enfants. Voir la classe *Concept* dans le diagramme.

CORBA et communication Manager/Agent

Parmi les alternatives présentées par CORBA, on a choisi la technique ***AMI Poller*** pour son avantage qu'elle n'impacte pas l'agent en plus qu'elle offre le mode asynchrone.

Par ailleurs, on peut se contenter d'une communication CORBA simple tout en adoptant le multithreading pour le Manager. Une explication plus élaborée sur cet aspect est exposée dans Idri et al. (2008).

5.2 Résultats et expérimentations

5.2.1 Exemple

Visualisation

Pour nos expérimentations et tests, on a intégré notre algorithme dans Galicia (voir adresse site web de Galicia).On a adopté le format de sortie GSH-XML pour s'interfacer avec Galicia. L'utilisateur peut soit lancer L'algorithme directement depuis Galicia, soit le lancer séparément et utiliser le résultat ainsi généré sous le format GSH-XML dans Galicia pour le visualiser.

Afin d'illustrer le fonctionnement de notre architecture, on présente ci-dessous le résultat de l'exemple de données transactionnelles listé dans la figure 4, à l'aide de Galicia. Les données de cet exemple représentent en fait un sous-ensemble du fichier de test « mushroom » utilisé dans la fouille de données. Ces données sont sous la forme **SLF** et **transactionnelle** ((a) respectivement (b)). Le treillis de Galois relatif au contexte de la figure 4-b est présenté par le graphe de la figure 4-c. Les nœuds représentent les concepts. Le concept racine se

trouve tout en haut de l'arborescence. Les concepts enfants sont liés directement au concept parent du niveau hiérarchique supérieur. On retrouve pour chaque concept (nœud) les références vers son intention et son extension.

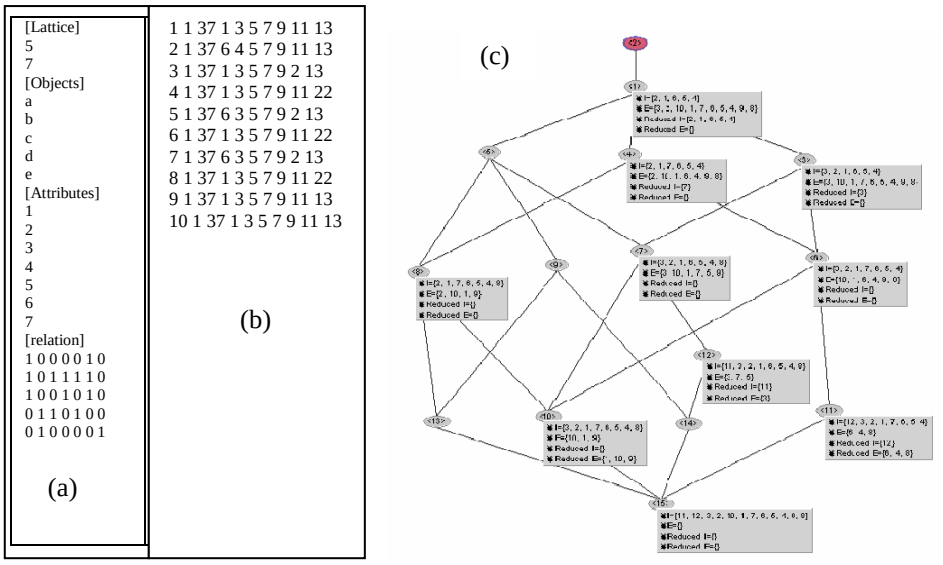


FIG. 4 – *Exemple : (a) SLF; (b) Transactionnel;(c) Treillis de Galois sous Galicia.*

5.2.2 Expérimentations

Pour nos expérimentations, nous avons réalisé les tests avec la configuration suivante sur la base du même environnement mentionné ci-dessous : une seule machine monoprocesseur et donc les agents partagent en fait le même processeur physique:

- Données : fichier « mushroom » (8124 objets et 119 attributs)
- Nombre d'agents : 3 agents sur la même machine
- Nombre de routeur logique : 1 seul routeur AMI

La figure ci-dessous montre la performance de l'algorithme vis-à-vis du fichier « mushroom ». Selon nos constatations et nos mesures ces résultats peuvent être améliorés avec un facteur de trois ; déjà en exécutant chaque agent sur une machine physique séparée. Il est évident qu'en multipliant le nombre d'agent on fait accroître la performance jusqu'au point à ne pas pénaliser la communication entre tous ces acteurs (Manager et Agents). D'autres techniques peuvent être appliquées pour améliorer la performance d'avantage. Celles-ci seront mentionnées dans la partie conclusion et perspective. Il faut noter que l'objectif principal de ce travail est dans un premier temps la conception et l'implémentation de l'architecture parallèle distribuée supportant la génération du treillis de Galois et par conséquent les règles d'association. La performance de l'algorithme viendra dans une seconde place surtout qu'il est à considérer, pour se comparer à des algorithmes déjà existants,

d'utiliser les techniques améliorant la performance telle que *diffset* de Zaki et ceci nécessite bien évidement leur adaptation à notre architecture. Ainsi, les temps réalisés dans Zaki et Hsiao ou Bouchahda et al., sont, pour cette version, plus performants que ceux de nos expérimentations du fait que le contexte actuel ne cible pas la performance en plus de l'environnement d'exécution qui est dans ce cas virtuel (nœuds des Agents virtuels).

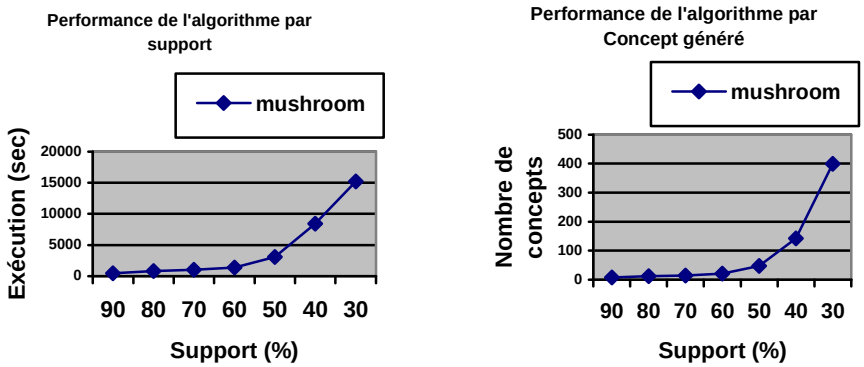


FIG. 5 – *Temps d'exécution et nombre de concepts générés par rapport au support.*

6 Règles d'association

Notre algorithme génère le Treillis de Galois et par conséquent, tout les motifs fermés. Le support étant donné (cardinalité de l'extension d'un concept), la génération des motifs fermés fréquents devient évidente. Seulement les règles d'association restent la connaissance la plus précieuse à explorer dans la base de données. Pour se faire, on est amené à exploiter le même treillis déjà généré.

On a examiné les techniques d'extraction des règles d'association basées sur les Treillis de Galois. Comme modèle, on s'est inspiré du framework Mirage de Zaki et Phoophakdee pour les générer vue que c'est algorithme repose sur le treillis de Galois aboutir au résultat. La figure ci-dessous expose l'architecture adoptée pour générer les règles d'association.

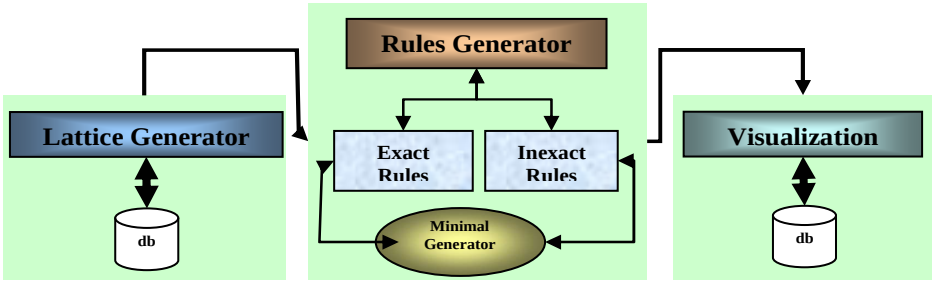


FIG. 6 – *Génération des règles d'association.*

La génération du Treillis de Galois est en fait la tâche la plus coûteuse, celle-ci est traitée dans les paragraphes précédents en adoptant l'approche parallèle distribuée. On exploite le

treillis déjà obtenu pour générer les règles d'association. L'élément clé dans ce processus est celui du générateur minimal d'un motif fermé donné qui est en fait un de ses sous-ensembles mais qui ne doit être contenu dans n'aucun de ses enfants directs dans le Treillis. La génération des règles exactes et inexactes entre deux motifs fermés est basée directement sur ces générateurs minimaux. On travaille actuellement sur la partie visualisation des règles d'association. L'implémentation du générateur est globalement achevée.

7 Conclusions et perspectives

La distribution de l'algorithme de construction de treillis de Galois nous a permis une bonne maîtrise du processus de génération du Treillis et ceci en dissociant ses tâches principales : la génération des concepts enfants (Agents) et la gestion de l'arbre des concepts (Manager). La première est gourmande en capacité du processeur et la deuxième en mémoire. Ceci nous a permis de tester et d'optimiser chacun de ces processus séparément et d'atteindre des résultats encourageants qui n'étaient pas possible avec l'algorithme séquentiel. Par ailleurs, le parallélisme nous a permis d'améliorer la performance et la scalabilité de l'algorithme en multiplexant les agents selon le besoin. Cependant comme cette approche construit le treillis en mémoire (Trie global) et effectue des calculs intenses (inclusions et intersections) pour tester la fermeture, deux points sont candidats d'optimisation si on veut la préconiser pour le datamining notamment : mémoire au niveau du manager et contrôle de fermeture au niveau des agents. On propose ci-dessous des alternatives pour surmonter ces difficultés.

Perspectives

- **Algorithmes hybrides** : En généralisant cette distribution sur plusieurs algorithmes on peut combiner des Agents et des Managers de différents algorithmes et choisir par conséquent les plus performants d'entre eux. Ceci générera des algorithmes hybrides mais sûrement plus robustes que les originaux.
- **Optimisation de la mémoire** : Pour optimiser l'utilisation de la mémoire il est possible de distribuer l'arbre hébergeant le treillis (Trie) du moment qu'on adopte une codification lexicographique et que les clés (extension on intension) sont ordonnées.
- **Optimisation du contrôle de fermeture** : les opérations d'inclusions et d'intersection pénalisent le processeur, de ce fait on peut faire recours à la technique **diffset** de Zaki pour simplifier le calcul.

Références

Barbut M. et Montjardet B. (1970), Ordre et Classification : Algèbre et Combinatoire. Hachette.

Ben Yahia S. et Nguifo E.M. (2004), Approches d'extraction de règles d'association basées sur la correspondance de Galois, RSTI-ISI, pages 23-55

Berry A. et Sigayret A. (2004), Discrete Applied Mathematics, volume 144, Issue 1-2, Discrete Mathematics & Data mining (DM & DM), pages 27-42.

Bordat J. P. (1986), Calcul pratique du treillis de Galois d'une correspondance, Math. Sci. Hum. 96 31-47.

Bouchahda A., Ben Yahia S. et Slimani Y., Une approche pour l'extraction des itemsets (fermés) fréquents, Université de Tunis El Manar

Chein M. (1969), Algorithme de recherche de sous-matrice première d'une matrice, Bull. Math. R. S. Roumanie 13.

Choi V. (2006), Faster Algorithms for Constructing a Concept (Galois) Lattice, Department of Computer Science, Virginia Tech, USA.

Ganter B. et Reuter K. (1991), Finding all closed sets : a general approach. Order, 8:283-290.

Ganter B. et Wille R. (1999), Formal Concept Analysis : Mathematical Foundations. Springer Verlag.

Idri A.F., Boulmakoul A. et Marghoubi R (2008), Une approche parallèle pour la construction des Treillis de Galois, SITA 2008.

Lakhal L. et Stumme G., Efficient Mining of Association Rules Based on Formal Concept Analysis

Levy G. et Baklouti F., A distributed version of the Ganter algorithm for general Galois Lattices

Njiwoua P. et Nguifo E. M., A Parallel Algorithm to build Concept Lattice, In proceedings of 4 Groningen Intl. Information Tech. Conf. for Students, pp. 103-107, 1997.

Pasquier N., Bastide Y., Taouil R. et Lakhal L (1999). Efficient mining of association rules using closed itemset lattices. Information systems. 24(1), p25-46.

Pasquier N., Bastide Y., Taouil R. et Lakhal L (1999), Closed set based discovery of small covers for association rules. In Actes des 15èmes journées Bases de Données Avancées (BDA'99), pages 361- 381.

Stumme G. (1999), Conceptual knowledge discovery with frequent concept lattices. FB4-Preprint 2043, TU Darmstadt.

www.orbacus.com

www.iro.umontreal.ca/~galicia/publication.html

Zaki M. et Hsiao Ching-Jui, CHARM : An Efficient Algorithm for Closed Itemset Mining

Zaki M. J. et Ogihara M. (1998), Theoretical foundations of association rules. Proc. 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, p1-7.

Zaki M. et Phoophakdee B. MIRAGE: A Framework for Mining, Exploring and Visualizing Minimal Association Rules Rensselaer Polytechnic Institute.

Une approche parallèle distribuée pour la génération des motifs fermés fréquents

Summary

The usefulness of the concept Lattice is proven in Data mining since the two fields are extremely dependent of each other in terms of closed item sets. The generation process of concept lattices have often an exponential time and space complexity, especially when dealing with very large databases in the domain of Data Mining. A couple of standard algorithms exist for building the concept lattice of a binary relation. In this paper, a parallel approach to improve the performance of the sequential algorithm is proposed. The algorithm and the system architecture are exposed as well as the aspects of its implementation. In this paper we also apply FCA (Concept Lattice) to generation of the association rules. This process is based on concept of minimal generators as used by Zaki.

Algorithme de construction & calcul d'un benchmark pour contrôle de k critères

Majda Fikri* , Mohammed El Khomssi** , Sahar Saoud***

Laboratoire Modélisation & Calcul Scientifique, Département de mathématiques,
FST-Fès, B.P 2202 V.N- Fès-Maroc
*majdafikri@yahoo.fr
**khomsixmath@yahoo.fr , ***math_rahas@hotmail.com

Résumé. Dans ce travail,l'originalité se manifeste à deux niveaux:
− Définir une écriture mathématique d'un benchmark B_p mesurant une propriété "P" engendrée par k critères.
− Coupler entre deux différentes notions, la causalité et l'indépendance entre les critères, afin de modéliser les contraintes de construction de ce benchmark, qui représente un outil de comparaison avec les leaders existants dans le domaine des entreprises.
Ainsi l'intérêt et la particularité de ce travail s'expriment par des hypothèses naturelles et compatibles avec le monde des entreprises, que nous avons posées dans le théorème donné par la suite, pour obtenir une solution unique d'un problème classique, sauf qu'ici nous avons intégré le savoir des experts et l'historique des résultats déjà connus par l'entreprise, ce qui renforce la valeur et l'objectivité du benchmark B_p.

1 Généralités sur le benchmarking / benchmark

1.1 Définition du benchmarking

Le benchmarking est considéré comme l'un des trois déterminants critiques d'une démarche d'amélioration de la performance; qui remonte à plusieurs siècles (Codling.S, 1996); les deux autres éléments étant la présence d'un leader convaincu et la qualité des processus de gestion et de production implantés dans l'entreprise (Matheson, 2000). En pratique, cette activité permet de prendre conscience des points faibles de l'organisme en étude lorsqu'il est comparé aux autres, et permet de donner de l'information sur ce qu'on doit faire afin d'y remédier. Vers la fin des années 70 le benchmarking a été développé par la compagnie américaine Xerox, dans le but de se comparer aux " Leaders" qui se sont positionés sur le marché, et de s'inspirer de leurs idées, de leurs pratiques et de leurs fonctionnement, afin de prendre une décision concernant un investissement lourd destiné à moderniser la gestion des stocks. Les militaires utilisent également l'analyse comparative depuis quelques décennies; pour prendre des décisions stratégiques et tactiques, ils comparent la puissance de leurs ennemis à la leur.

En général, en identifiant les points à améliorer à base de la performance des organismes comparables avec celui en étude, le Benchmarking permet de déterminer quels résultats souffrent d'un écart par rapport à un groupe témoin d'organismes. Par la suite, ces résultats peuvent être améliorés par des modifications appropriées dans les pratiques de gestion et de production. D'où l'intérêt de définir une propriété engendrée par un ensemble de critères qui donnent un sens à l'action de se comparer.

1.2 Différents types de benchmarking

Tout benchmarking se base sur une procédure qui nécessite un ou des indicateurs quantitatifs , nommés benchmarks, chiffrés de performance, tirés de l'observation des résultats de l'entreprise qui a réussi le mieux dans le domaine étudié, puis les prendre comme valeurs de référence. Il existe différents types de benchmarks (SCHOETTL 2005):
- Benchmarking interne : Comparaison d'un processus, d'un produit ou d'un service similaire à l'intérieur de l'organisation.
- Benchmarking concurrentiel / compétitif : Comparaison aux meilleurs concurrents sur le marché.
- Benchmarking fonctionnel : Comparaison à des entreprises de la même branche non-concurrente.
- Benchmarking générique : Comparaison aux meilleures entreprises du monde.

2 Modélisation mathématique d'un Benchmark

2.1 Modélisation et contraintes

Le benchmarking est une méthode ou processus de mesure d'une propriété "P" (exp, performance, rentabilité, coût...) dans une entreprise ou organisme donné, qui se base sur le calcul d'un benchmark (indice) principalement relatif à la propriété étudiée. Nous proposons alors de modéliser mathématiquement l'écriture générale d'un Benchmark quelque soit la propriété étudiée ou l'entreprise considérée, sachant que cette propriété est engendrée par un nombre de critères.
Pour le calcul de ce Benchmark noté B_P nous nous basons sur un tableau contenant un certain nombre de critères indépendants choisis par les experts[1]; qu'ils trouvent importants et essentiels, et qu'ils ordonnent suivant leurs priorités et causalité historique. Chaque critère sera évalué par une note dans une échelle de notation considérée (Fikri et al.2007).

	$N_1 = 2$	$N_2 = 4$	$N_3 = 6$	$N_4 = 8$	$N_5 = 10$
Critère 1		✓			
Critère 2				✓	

TAB.1: Exemple d'évaluation d'un Benchmark basé sur deux critères.

Dans l'esprit de la modélisation mathématique que nous avons développée par la suite, et qui cible à exprimer quantitativement une donnée qualitative qui n'est d'autre que

[1]les experts du domaine d'étude relativement à "P"

la propriété P, nous proposons les trois définitions suivantes.
Définition1: On appelle Benchmark B_P mesurant une propriété "P" donnée, la somme des notes associées aux k critères pondérées par leurs potentiels

$$B_P = \sum_{i=1}^{i=k} \alpha_i N_i$$

Où K est le nombre de critères considérés pour définir B_P.
N_i est la note associée au critère i, et α_i est la pondération associée au critère i
Définition2: On appelle un Benchmark partiel de B_P la quantité B_l définie par:

$$B_l = \sum_{i=1}^{i=l} \alpha_i N_i \quad où \quad l \leq k-1$$

2.1.1 Contraintes des pondérations

Associer une pondération α_i à un critère i, c'est exprimer d'une manière quantitative l'intérêt du critère i dans la définition de B_P. En effet, chacune des α_i traduit le poids du critère i pour la propriété "P" donc, d'une part, on a nécessairement

$$\alpha_i \geq 0 \; pour \; tout \; i \in \{1, 2...k\} \tag{1}$$

D'autre part, ces pondérations répondent à des contraintes exprimant les conditions vérifiées par les critères associés:
a. Causalité: Supposons; par exemple; que l'étude de la propriété "P" pour les AGR [2] se base sur certains critères parmi lesquels on trouve:
- h: Durabilité.
- f: Amélioration de la commercialisation.

Il est claire que l'historique, ou la succession logique, des événements nous impose de ne pas discuter la possibilité de durabilité d'une AGR avant de passer par l'évaluation de l'amélioration de la commercialisation de ses produits, ainsi le critère f doit apparaître avant h. Cet ordre de causalité entre les critères est exprimé au niveau des pondérations par la contrainte suivante. Soient i et j indices de critères ordonnés:

$$\alpha_i \geq \alpha_j \quad pour \; tout \quad i < j \quad dans \{1, 2, ...k\} \tag{2}$$

b. Indépendance: Nous considérons que deux critères i et j sont dépendants si et seulement si l'un est dominé par l'autre, ou les deux expriment la même donnée par rapport à la propriété "P" en étude. Par exemple,
- i: L'autofinancement.
- j: La dépendance en capital.

Dans ce cas nécessairement l'une des pondérations associées est nulle car son critère exprime une donnée déjà faite par un autre. Par conséquent l'indépendance des critères est exprimée par la contrainte suivante:

$$\alpha_i \neq 0 \quad pour \; tout \quad i \in \{1, 2, ...k\} \tag{3}$$

[2] AGR: Activité génératrice de revenu, dite aussi projet générateur de revenu

Ainsi lorsque nous considérons k critères indépendants, alors aucun d'eux n'est engendré par une sous famille des autres.
Les k critères considérés décrivent entièrement la propriété "P" en question, sinon nous ajoutons les critères nécessaires pour compléter cette description, ce que nous exprimons par:

$$\sum_{i=1}^{k} \alpha_i = 1 \quad avec\ k\ \ le\ seuil\ de\ saturation\ des\ critères \tag{4}$$

Définition3: On appelle seuil de saturation, le nombre $k \in IN^*$ nécessaire et suffisant pour décrire entièrement la propriété "P". ▪

Lemme1: Soit $\alpha_0 = \frac{1}{k}$. La première et la dernière pondérations vérifient:

$$\alpha_1 > \alpha_0 \quad et \quad \alpha_k < \alpha_0 \tag{5}$$

Preuve: Pour tout i dans $\{1, 2, ...k\}$, on a : $\alpha_1 > \alpha_i > \alpha_k$
en écrivant cette inégalité k fois pour toutes les pondérations α_i, puis faisant la somme terme à terme des inégalités, et ensuite écrivant la contrainte (4), nous obtenons: $k\alpha_1 > 1 > k\alpha_k$, d'où le résultat du lemme ▪

2.1.2 Fonction Objective

En programmation mathématique, il s'agit souvent d'optimiser la fonction objective ou fonction cot, dite encore fonction économique (Minoux, 1983), sous certaines contraintes dont l'ensemble constitue un système dit problème de satisfaction de contraintes CSP, que ses solutions représentent "l'ensemble des solutions réalisables" dans laquelle nous optimisons la fonction objective. Bien que notre modélisation ci-dessus entre dans le cadre des CSP, le but sera de le résoudre afin de construire la fonction objective et non pas de l'optimiser, ce qui donne un aspect particulier à ce problème, et le libère du contexte classique. Le benchmark B_P est donné par:

$$B_P = \sum_{i=1}^{k} \alpha_i N_i$$

telles que les α_i vérifient (1),(2),(3) et (4), ainsi nous modélisons notre problème comme suit:

$$\begin{cases} Trouver \quad B_P = \sum_{i=1}^{k} \alpha_i N_i \\ sous\ contraintes \\ \sum_{i=1}^{k} \alpha_i = 1 \\ \alpha_i > 0 \quad pour\ tout\ i\ \in \{1, 2...k\} \\ \alpha_i \geq \alpha_j \quad pour\ tout\ i \leq j\ dans\ \{1, 2...k\} \end{cases}$$

Les N_i sont des valeurs positives dépendent du cas [3] pour lequel nous mesurons la propriété "P", et les α_i des coefficients fixes pour un Benchmark donné B_P, alors

[3] un cas étudié peut être une entreprise, organisation, expérience...

définir B_P revient à déterminer toutes les pondérations α_i; ainsi le problème devient :

$$(Pb) : \begin{cases} Calculer\ \alpha_i\ avec\ i \in \{1, 2, ...k\} \\ Pour\ B_P = \sum_{i=1}^{k} \alpha_i N_i \\ sous\ contraintes \\ \sum_{i=1}^{k} \alpha_i = 1 \\ \alpha_i > 0\ \ pour\ tout\ i\ \in \{1, 2...k\} \\ \alpha_i \geq \alpha_j\ \ pour\ tout\ i \leq j\ dans\ \{1, 2...k\} \end{cases}$$

Lemme2 : La suite $(B_l)_{l=1}^{k-1}$ de benchmarks partiels est positive croissante, et on a

$$\lim_{l \longrightarrow k} B_l = B_P$$

Preuve: Soit $l > l'$ dans $\{1, 2...k-1\}$, de la définition de B_l nous obtenons
$B_l - B'_l = \sum_{i=l'+1}^{l} \alpha_i N_i \ > 0$ ($\alpha_i > 0$ *et* $N_i > 0$ pour tout i), d'où $(B_l)_l$ est croissante.
De plus on a $B_1 = \alpha_1 N_1 \geq 0$ donc $0 \leq B_1 \leq B_l$ *pour tout* $1 \leq l$

$$et \qquad \lim_{l \longrightarrow k} B_l = \lim_{l \longrightarrow k} \sum_{i=1}^{l} \alpha_i N_i = \sum_{i=1}^{k} \alpha_i N_i = B_P \qquad .$$

2.2 Ecriture algébrique du problème CSP*

La résolution de (Pb) peut être vue classique, en effet il suffit de voir l'exemple suivant:
Considérons le problème (Pb) pour $k = 3$, donc les vecteurs
$(\alpha_1 = 0,5; \alpha_2 = 0,3; \alpha_3 = 0,2)$ et $(\alpha_1 = 0,6; \alpha_2 = 0,3; \alpha_3 = 0,1)$ sont des solutions de (Pb). donc nous pouvons constatater que:
- il n y a pas nécessairement unicité de la solution.
- il n'existe aucun critère ou raison d'admettre ou de rejeter une solution.
- il n' y a aucun lien entre une telle solution et les données de l'entreprise considérée.
Par conséquent ce genre de solution est automatiquement rejeté par les décideurs.
Pour éviter cette carence, nous allons choisir une écriture algébrique qui nous facilitera le calcul des pondérations α_i, en tenant compte, d'une part, du savoir faire des experts, et d'autre part, de l'historique des mesures de la propriété "P". En effet, les expressions:

$$B_j = \sum_{i=1}^{j} \alpha_i N_i \ \ pour\ tout\ j \in \{1, 2, ..k-1\}; \quad et \quad B_P = \sum_{i=1}^{k} \alpha_i N_i$$

peuvent se regrouper dans la forme matricielle suivante:

$$N\alpha = B \quad \textbf{(S)}$$

$$Où \ \ \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ .. \\ .. \\ \alpha_k \end{pmatrix} \quad ; \quad B = \begin{pmatrix} B_1 \\ B_2 \\ .. \\ B_{k-1} \\ B_P \end{pmatrix} \ \ et \ \ N = \begin{pmatrix} N_1 & 0 & 0 & & 0 \\ N_1 & N_2 & 0 & & 0 \\ & & & & \\ N_1 & N_2 & & N_{k-1} & 0 \\ N_1 & N_2 & & N_{k-1} & N_k \end{pmatrix}$$

[0]* Dans la littérature de la programmation mathématique la notation CSP signifie: Constraint Satisfaction Problem

3 Technique de calcul de la pondération

3.1 Cadre général pour l'étude du système (S)

Pour travailler dans un cadre plus général nous considérons le système (S_g) suivant :

$$AX = b \qquad (S_g)$$

Où A est une matrice carrée triangulaire inférieure a diagonale strictement positive:

$$A = \begin{pmatrix} a_1 & 0 & 0 & & 0 & 0 \\ a_1 & a_2 & 0 & 0 & 0 & 0 \\ \cdots & & & & & \\ \cdots & & & & & \\ a_1 & a_2 & a_3 & & a_{k-1} & 0 \\ a_1 & a_2 & a_3 & & & a_k \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_k \end{pmatrix} \text{ un vecteur donné avec}$$

$0 < b_i < b_{i+1} : \ i \in \{1, 2, ..., k\}$ et $b_0 = 0$ posé dans toute la suite par convention.
Proposition1: Le système (S) admet une solution unique X^* de composantes

$$x_i^* = (b_i - b_{i-1})(a_i)^{-1} \quad i = 1, 2..., k$$

Preuve: Puisque $det(A) = \prod_{i=1}^{k} a_i$ et $a_i \neq 0$ pour tout i
Alors la matrice A est inversible et le problème (S) admet une solution unique:

$$X^* = A^{-1}b$$

Par un calcul simple de la matrice inverse A^{-1} (Brezinski et Redivo-Zaglia),(Bronson et Richard,1997), on trouve que pour $i; j$ dans$\{1, 2, .., k\}$, on a:

$$A_{ij}^{-1} = a_i^{-1}(\delta_{ij} - \delta_{(i-1)j})$$

Où δ_{ij} étant le symbole de Kroneker: $\delta_{ij} = 1$ si $i = j$ et 0 sinon.
Ainsi le vecteur X^* de $\mathbb{R}_+^*$ a pour composantes $(x_i^*)_{i=1,2,...,k}$ tel que:
$x_i^* = (b_i - b_{i-1})(a_i)^{-1}$ *pour tout* $i \in \{1, 2, ..., k\}$

Définitions 4: variation-entrée & variation-note

1/ Pour tout vecteur b nous définissons les composantes du vecteur **variation-entrée** Δb comme suit: $\Delta b_i = b_i - b_{i-1}$, $i \in \{1, 2, ..k\}$
2/ Pour toute matrice A, comme définie au problème (S), nous définissons les composantes du vecteur **variation-note** ε par: $\varepsilon_i = a_{i+1}(a_i)^{-1}$, $i \in \{1, 2, ..k-1\}$
Lemme3: Si le vecteur b donné au problème (S) est tel que:

$$b_i < \varepsilon_{i-1}\Delta b_{i-1} + b_{i-1} \text{ pour tout } i \in \{2, 3, ...k\} \qquad (6)$$

Alors X^* la solution de (S) est de composantes décroissantes:

$$x_i^* > x_j^* \text{ pour tout } i < j \text{ dans } \{1, 2, ...k\} \qquad (7)$$

Preuve: Supposons que (7) est vérifiée, cela implique que $b_i - b_{i-1} < \varepsilon_{i-1}\Delta b_{i-1}$ or de la définition(4) les termes généraux ε_{i-1} et Δb_{i-1} sont de la forme:

$$\varepsilon_{i-1} = a_i a_{i-1}^{-1} \quad et \quad \Delta b_{i-1} = b_{i-1} - b_{i-2}$$

ce qui implique de (7) que $(b_i - b_{i-1})(a_i)^{-1} < (b_{i-1} - b_{i-2})(a_{i-1})^{-1}$
de la proposition1, on a: $x_i^* = (b_i - b_{i-1})(a_i)^{-1}$
par suite: $x_i^* < x_{i-1}^*$ pour tout $i \in \{2, 3, ..., k\}$

Proposition2 Si le vecteur b définie dans (S) est de composantes vérifiant:

$$b_i \ < \ (1 \ - \sum_{j=1}^{j=i-1} x_j \ ) \ a_i \ + \ b_{i-1} \quad \textit{pour tout } i \in \{1, 2, ..k\} \tag{8}$$

Alors X^* la solution de (S) est telle que :

$$x_i^* < 1 \quad et \quad \sum_{j=1}^{j=i} x_j^* < 1 \ \ \textit{pour tout } i \in \{1, 2, ..k\} \tag{9}$$

Preuve: Supposons que (9) est vérifié, alors $b_i - b_{i-1} < \ (1 \ - \sum_{j=1}^{j=i-1} x_j \ ) \ a_i$
Par hypothèse de (S) on a $a_i > 0$ pour tout i, l'inégalité précédente devient:
$(b_i \ - b_{i-1})(a_i)^{-1} < \ 1 \ - \sum_{j=1}^{j=i-1} x_j$
De la proposition1, nous déduisons que: $x_i^* = b_{i-1}(a_i)^{-1} > 0$
ce qui implique: $x_i^* < 1 \ - \sum_{j=1}^{j=i-1} x_j^*$

par suite :
$$\sum_{j=1}^{j=i} x_j^* \ = \ \sum_{j=1}^{j=i-1} x_j^* \ + \ x_i^* \ < \ 1$$

et comme $x_i^* < \sum_{j=1}^{j=i} x_j^*$ alors $x_i^* < 1$ *pour tout* i
d'où les résultats de la proposition.

Lemme4 : Soit b un vecteur du problème (S), Si l'inégalité :

$$b_{k-1} > b_{k-2} + \frac{a_{k-1}}{2}(1 - \sum_{i=1}^{k-2} x_i) \tag{10}$$

est vérifiée, alors X^* la solution de (S) est telle que:

$$1 - \sum_{i=1}^{k-1} x_i^* < x_{k-1}^* \tag{11}$$

Preuve: Supposons que (11) est vérifiée
Comme $a_{k-1} > 0$ par hypothèse dans le problème (S), alors

$$(b_{k-1} - b_{k-2})(a_{k-1})^{-1} > \frac{1}{2}(1 - \sum_{i=1}^{k-2} x_i)$$

de la proposition1, la solution de (S) est de composantes $x_{k-1}^* = (b_{k-1} - b_{k-2})(a_{k-1})^{-1}$
ce qui implique que: $2x_{k-1}^* > 1 - \sum_{i=1}^{k-2} x_i$, d'où

$$x_{k-1}^* > 1 - \sum_{i=1}^{k-2} x_i - x_{k-1}^* = 1 - \sum_{i=1}^{k-1} x_i$$

3.2 Choix des échelles d'évaluation

3.2.1 Normalisation des échelles

D'une manière générale, le contexte d'étude et la nature de la propriété "P" imposent pour chaque critère une échelle de notation. Sans perdre de généralité, nous proposons dans ce paragraphe d'unifier ces échelles en normalisant ces notations afin d'avoir toutes les valeurs entre 0 et 1.

Soient $\tilde{N}_i^1, \tilde{N}_i^2, ..., \tilde{N}_i^l$ les notations possibles dans l'échelle de mesure d'un critère donné i, alors les nouvelles notations normalisées que peut prendre ce critère seront de la forme,

$$N_i^j = (\max_{j=1}^{j=l} \tilde{N}_i^j)^{-1} \tilde{N_i^j} \quad , \quad \textit{et on a bien} \ : \ 0 < N_i^j \leq 1 \ \textit{pour tout } j$$

Il est évident que $\max_{j=1}^{j=l} \tilde{N}_i^j \neq 0$ sinon il n'y a aucune raison de parler d'une échelle.

3.2.2 Indication pratique

Pour éviter le calcul avec des valeurs comprises entre 0 et 1; au niveau de l'implantation sur machine; nous proposons de multiplier[4] les graduations N_i par 100, ou de de les remplacer par $\exp(N_i)$

3.3 Matrice-moyenne &Vecteur-entrée

3.3.1 Construction de la matrice-moyenne

Pour un calcul significatif des pondérations basé sur la réalité, nous introduisons les moyennes $\bar{N}_i$ pour chaque critère i. Nous considérons un échantillon de M cas étudiés. Pour chacun de ces cas nous évaluons les k critères (tâche effectuée par les experts), donc pour chaque critère i nous aurons une réponse des experts formalisée par un vecteur de notes $(N_i^1, N_i^2, ...N_i^M)$ à partir des quelles nous calculons la moyenne empirique $\bar{N}_i$ associée au critère i telle que :

$$\bar{N}_i = M^{-1} \sum_{j=1}^{M} N_i^j$$

Dans le cas où les notes N_i^j $(j \in \{1, 2, ..M\}$ suivent une lois de probabilité $P(N_i^j) = p_i^j$, alors la moyenne empirique est remplacée par l'espérance mathématique:

$$\bar{N_i^j} = E(N_i^j) = \sum_{i=1}^{M} N_i^j p_i^j$$

Remarque 1: Il est évident pour les statisticiens de calculer $\sigma_{N_i}^2$ pour mesurer l'erreur effectuée lorsqu'on substitue une variable par sa moyenne (Masiéri, 2001), cette

[4]Si on remplace N_i par $N_i' = 100 N_i$, le benchmark $B_P = \sum_{i=1}^{k} \alpha_i N_i'$ exprimera la mesure de "P" en pourcentage.

variance est donnée par:

$$\sigma^2_{N_i^j} = \sum_{i=1}^{M} N_i^{j\,2} p_i^j q_i^j + 2 \sum_{1\le i<t\le M} N_i^j N_i^t p_i^j p_i^t \quad avec \quad q_i^j = 1 - p_i^j$$

Définition5: Soit un échantillon de M cas étudiés, pour lesquels on mesure k critères donnés. On appelle matrice-moyenne $\widetilde{N}$ la matrice définie par les moyennes des k critères comme suite:

$$\widetilde{N} = \begin{pmatrix} \bar{N}_1 & 0 & 0 & 0 & 0 & & 0 \\ \bar{N}_1 & \bar{N}_2 & 0 & 0 & 0 & & 0 \\ & & & & & & \\ \bar{N}_1 & \bar{N}_2 & & & & \bar{N}_{k-1} & 0 \\ \bar{N}_1 & \bar{N}_2 & & & & \bar{N}_{k-1} & \bar{N}_k \end{pmatrix}$$

Cette matrice est de même nature que la matrice A du système (S).

3.3.2 Construction du vecteur-entrée

Rappelons que pour définir un benchmark B_P nous nous basons sur k critères choisis et ordonnés suivant leur priorité et causalité historique. Cet ordre de causalité nous permet de supposer à chaque étape n (n allant de 1 à k-1) que les n premiers critères sont vérifiés et que les autres n'ont pas encore eu lieu. Cette situation vaut une évaluation B_n^e de la propriété "P" par les experts; qui n' est qu'une valeur du benchmark partiel B_n pour $N_i = \bar{N}_i$ pour tout $i \in \{1, 2, ...n\}$

$$Par\ suite \quad B_n^e = \sum_{i=1}^{n} \alpha_i^* \bar{N}_i \quad pour\ tout\ n \in \{1, 2, k-1\} \tag{12}$$

Remarquons que le passage d'une étape n à une étape $n+1$ se fait par la réalisation en plus du critère $n+1$, donc c'est une amélioration de la mesure de l'information ou de la propriété "P", d'où on a: $B_e^n < B_e^{n+1}$ *pour tout* $n \in \{1, 2..., K-2\}$
D'après (13) on a:

$$B_n^e = \sum_{i=1}^{n} \alpha_i^* \bar{N}_i \ = \ \sum_{i=1}^{n-1} \alpha_i^* \bar{N}_i \ + \ \alpha_n^* \bar{N}_n \quad pour\ tout\ n \in \{1, 2, k-1\}$$

donc $B_n^e = B_{n-1}^e + \alpha_n^* \bar{N}_n$ *pour tout* $n \in \{1, 2, k-1\}$ et $B_0^e = 0$ (par convention[5])

$$d'où \qquad \alpha_n^* = \frac{B_n^e - B_{n-1}^e}{\bar{N}_n} \quad pour\ tout\ n \in \{1, 2, k-1\} \tag{13}$$

Posons $B^e = \begin{pmatrix} B_1^e \\ B_2^e \\ ... \\ B_k^e \end{pmatrix}$ le vecteur obtenu à partir des évaluations B_e^n (n allant de 1 à k-1) des experts comme décrit auparavant, et B_k^e prise dans $\mathbb{R}^+$ telle que $B_e^k > B_{k-1}^e$.

[5]Cette convention traduit qu'aucun des critères n'est vérifié.

Puisque ces valeurs ne dépendent d'aucun système de calcul, nous les considérons des entrées du problème, ainsi le vecteur B^e est dit vecteur-entrée.

3.3.3 Résolution du problème (Pb) et Algorithme de calcul

D'après (13), on peut écrire $\widetilde{N}\alpha^* = B^e$, où $\widetilde{N}$ la matrice moyenne, et α^* est le vecteur solution du système $\widetilde{N}\alpha = B^e$. Pour que ce vecteur soit solution du problème (Pb) il suffit qu'il satisfait les contraintes suivantes:

$$\begin{cases} \sum_{i=1}^{k} \alpha_i = 1 \\ \alpha_i > 0 \ \ \textit{pour tout } i \ \in \{1, 2...k\} \\ \alpha_i > \alpha_j \ \ \textit{pour tout } i < j \ \textit{dans } \{1, 2...k\} \end{cases}$$

Or le problème $\widetilde{N}\alpha = B^e$ est identique au système (S)
Alors les résultats des lemmes (3),(4) et la proposition(2) nous permet de satisfaire ces contraintes.
Théorème Soit B^e le vecteur-entrée. Considérons le problème $\widetilde{N}\alpha = B^e$
Sous les conditions suivantes:

$$B_i^e < \varepsilon_{i-1}\Delta B_{i-1}^e + B_{i-1}^e \ \textit{avec} \ \varepsilon_{i-1} = \frac{\bar{N}_i}{\bar{N}_{i-1}} \ \ \textit{pour tout } i \in \{2, 3, ...k\} \tag{14}$$

$$B_i^e \ < \ (1 \ - \sum_{j=1}^{j=i-1} x_j \ ) \ \bar{N}_i \ + \ B_{i-1}^e \quad \textit{pour tout } i \in \{1, 2, ..k\} \tag{15}$$

$$B_{k-1}^e > B_{k-2}^e + \frac{\bar{N}_{k-1}}{2}(1 - \sum_{i=1}^{k-2} x_i) \tag{16}$$

Le vecteur α^* solution unique du problème (Pb) est de composantes:
$\alpha_i^* = (B_i^e - B_{i-1}^e)(\overline{N_i})^{-1}$ *pour tout* $i \in \{1, 2, ..., k-1\}$
$\alpha_k^* = 1 - \sum_{i=1}^{k-1} \alpha_i^*$
Preuve: Ce théorème est un résultat direct des lemmes (3), (4) et la proposition(2) **.**
Remarque 2: Les conditions du théorème sont réalisables et homogènes car elles expriment le fait d'améliorer la note chaque fois que l'expert a des informations de plus sur la propriété, la compatibilité des inégalités de ce théorème est assurée d'une manière implicite du fait que $1/2 < 1$.

Algorithme de calcul :

Soit E la réponse de l'expert à chaque itération.
Au départ; étape 0; aucun critère n'est vérifié donc : $B_e^0 = 0$
✓ A l'itération 1, $E = B_e^1$
- Si $B_e^1 \geq N_{max}$ alors "erreur d'évaluation"
- Sinon lors, $\alpha_1^* = \frac{B_1^1}{\bar{N}_1}$ et passer à l'itération suivante.
...
✓ A l'itération $2 \leq i \leq k-1$, $E = B_e^i$ -

$$Si \begin{cases} B_i^e < \varepsilon_{i-1} \Delta B_{i-1}^e + B_{i-1}^e \; avec \; \varepsilon_{i-1} = \frac{\bar{N}_i}{\bar{N}_{i-1}} \\ B_i^e \; < \; (1 \; - \sum_{j=1}^{j=i-1} \alpha_j^* \;) \; \bar{N}_i \; + \; B_{i-1}^e \end{cases}$$

Alors $\alpha_i^* = \frac{B_e^i - B_e^{i-1}}{\bar{N}_i}$

$$\begin{cases} \bullet \; Si \; \alpha_i^*(k-1) > \theta \\ alors \; passer \; à \; l'itération \; suivante \\ \bullet \; \text{Sinon arrêt de l'algorithme.} \end{cases}$$

- Sinon, alors "erreur d'évaluation"
...
✓ A l'itération $i = k$, - Si $B_{k-1}^e > B_{k-2}^e + \frac{\bar{N}_{k-1}}{2}(1 - \sum_{i=1}^{k-2} \alpha_i^*)$
Alors $\alpha_k^* = 1 - \sum_{j=1}^{k-1} \alpha_j^*$ et fin algorithme.
- Sinon "erreur d'évaluation"
Fin Algorithme

A la fin de cet algorithme nous aurons comme résultat les composantes du vecteur α^*, et par suite l'écriture explicite de B_p que nous utiliserons directement à chaque fois qu'on veut évaluer la propriété "P" en étude.

4 Conclusion

Le but principal de ce papier était d'obtenir via une modélisation mathématique un Benchmark B_P mesurant une propriété P engendrée par k critères, et construit d'une manière unique comme fruit de la démarche suivante:
Premièrement, nous avons résolu le problème de satisfaction de contraintes (CSP) obtenu à travers une écriture du système représenté par la matrice moyenne des notes des critères qui reflète l'exploitation de l'historique des données de l'entreprise, et un vecteur entré exprimant l'intégration du savoir faire des experts.
Deuxièmement, nous avons présenté un algorithme simple permettant d'obtenir pour k critères choisis l'écriture unique explicite du benchmark B_p qui ne demande que les valeurs des évaluations des critères dans la structure en étude (AGR, entreprise, organisation...) pour pouvoir l'exploiter.
Afin de comparer les résultats de notre benchmark avec d'autres, nous avons fait une recherche dans la littérature existante en économie et en économétrie, et nous n'avons

trouvé aucune modélisation mathématique d'un benchmark, mais vu l'intégrité d'un historique de données de l'entreprise et aussi du savoir faire des experts, juste dans la construction de B_p et non pas à chaque mesure, nous pouvons conclure que cet indice demeure efficace et objectif
Dans un prochain travail nous généraliserons les résultats par la relaxation de la contrainte de causalité, via l'utilisation des classes de critères.

Références

Brezinski. C , Redivo-Zaglia. M, Méthodes numériques directes de l'algèbre matricielle, Ellipses 1ère edition -Collection : Mathématiques à l'université

Bronson, Richard (1997). Calcul Matriciel - Cours Et Problèmes, France Mac Graw Hill Collection : Série Schaum

Codling,S. (1996). Best practice benchmarking, Gulph Pub. Col,Houston, Texas

Fikri.M , Elkhomssi.M, Saoud.S (2007). Benchmarking par des outils mathématiques dans le cadre des structures micro-économiques.PRLMC FST-Fès

Masiéri.W (2001). Statistique et calcul des probabilités Editions Dalloz

Matheson, D.(2000). Achieving performance excellence, New Zealand Management

Minoux.M (1983). Programmation Mathématique, Théorie et algorithmes, Tome1-Dunod

Schoettl .J (2005). Réaliser un benchmarking : se comparer aux meilleurs pour progresser , INSEP Consulting Editions

Summary:

In this work, the originality can be seen on two levels:
– Defining a writing mathematical of a benchmark B_p measuring a property "P" generated by k criteria.
– Coupling between two different concepts, causality and independence between the criteria, in order to model the constraints of construction of this benchmark, which is a tool for comparison with the leaders existing in the domain of business.
Thus the interest and the particularity of this work expressed by natural assumptions and compatible with the business, which we raised in the theorem given thereafter to obtain a unique solution of a problem, except that here we have integrated the knowledge of experts and the historical results are already known by the company, which reinforces the value and objectivity of the benchmark B_p.

WCSS: un système cellulaire d'extraction et de gestion des connaissances

Mohamed Benamina, Baghdad Atmani

Département Informatique, Faculté des Sciences, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie
benamina.mohamed@gmail.com, atmani.baghdad@univ-oran.dz

Résumé. WCSS est un système cellulaire d'extraction et de gestion des connaissances dédié à la fouille des données. WCSS s'adresse à deux types de publics. D'un côté, il exploite l'environnement de fouille des données offert par la plate forme Weka qui le rend ainsi accessible à une utilisation sur des données réelles. De l'autre, du fait que les règles générées sont conjonctives, il se prête à une utilisation directe avec les systèmes experts classiques. WCSS est écrit en java et s'appuie sur les graphes d'induction générés par le principe de la méthode SIPINA.
Mots-clés : Fouille des données, apprentissage automatique, graphe d'induction, automate cellulaire, extraction de règles.

1 Introduction

Parmi les méthodes de fouille des données (Zighed et Rakotomalala, 2000), l'induction des règles à partir d'exemples tient une place particulière parce qu'elle est celle qui réalise le meilleur compromis entre performances en précision et compréhensibilité (Lefebure et Venturi, 2001). De plus, il s'agit certainement de la forme de connaissance la plus utilisée dans les systèmes experts. Il existe de nombreuses méthodes qui produisent des modèles d'induction en utilisant explicitement la forme de règle "Si *Prémisse* Alors *Conclusion*" (Quinlan, 1986, 1993). On distingue notamment :

- la famille des algorithmes *AQ* (Michalski et al., 1986) parmi lesquels nous citons *AQ11*, *AQ15* et *CN2* (Clark et Boswell, 1991) qui consistent à explorer par spécialisation une ou plusieurs règles entièrement consistantes pour chaque classe.
- la famille des algorithmes des *listes de décision* (Rivest, 1987). Le modèle se présente de la manière suivante : Si *Prémisse1* Alors *Conclusion1*
  Sinon Si *Prémisse2* Alors *Conclusion2*
  Sinon Si...

Malgré les qualités des stratégies précédentes, nous avons opté pour une autre méthode : l'induction par graphe (Rabaseda et Zighed, 1996). Cette méthode présente l'avantage d'utiliser un système de représentation qui respect parfaitement les critères de précision et de compréhensibilité (Rakotomalala et Zighed, 1999).

La représentation et le traitement de la connaissance sont les deux questions rebutantes dans la conception de n'importe quel système d'apprentissage automatique (Clark, 1989, 1990). C'est peut-être également la considération fondamentale dans la conception de n'importe quel système de fouille des données, parce que la représentation utilisée peut réduire la *complexité* de *stockage* et diminue ainsi la *complexité* du *traitement* (Atmani et

Beldjilali, 2007a, 2007b). L'implémentation et l'intégration de la machine cellulaire ACSIPINA dans la plate forme Weka (Witten et al., 1999) est double. D'une part, WCSS, qui est issu des travaux de Atmani et Beldjilali (2007b) propose un nouvel environnement cellulaire de génération, de représentation et d'optimisation des graphes d'induction. En outre, pour alimenter la base de connaissance d'un système expert cellulaire, WCSS produit des règles conjonctives à partir d'une représentation booléenne des graphes d'induction. La représentation cellulaire, d'un graphe d'induction, facilite la transformation des règles dans des expressions booléennes équivalentes, ainsi on peut compter sur l'algèbre booléenne élémentaire pour vérifier plusieurs simplifications. En effet, l'utilisation directe des règles extraites d'un graphe d'induction n'est pas possible, et ce pour plusieurs raisons. D'une part, les règles de production sont généralement des règles conjonctives et non disjonctives-conjonctives comme celles issues d'un graphe (Atmani et Beldjilali, 2007a). D'autre part, les variables manipulées peuvent apparaître plusieurs fois dans le graphe. Les graphes obtenus peuvent être de grande taille et comportent des informations redondantes, incohérentes, …etc. Il semble nécessaire de simplifier les graphes générés, par conséquent les règles avant de les utiliser dans un système expert.

WCSS est un système cellulaire d'extraction et de gestion des règles conjonctives. WCSS est entièrement écrit en JAVA et s'appuie sur les graphes d'induction générés par le principe de la méthode SIPINA élaborée par Zighed (1986). WCSS peut être utilisé de deux manières différentes. La première, en générant des règles dites conjonctives il se prête facilement à une utilisation directe avec les systèmes experts classiques. La deuxième, en exploitant l'environnement de traitement de données offert par la plate-forme Weka, il devient, accessible à une utilisation de type « chargé d'études » sur des données réelles. Seul le premier objectif est arboré dans cet article : l'extraction et la gestion cellulaire des règles.

Cet article est structuré comme suit. La section 2 est consacrée en générale l'apprentissage automatique supervisé par induction, et en particulier à la présentation du schéma général du système WCSS. Dans la section 3 nous illustrons à travers un exemple le principe de la méthode SIPINA. La machine cellulaire est abordée dans la section 4. Nous présentons son organisation générale et nous détaillons son mode de fonctionnement. Enfin, nous concluons par la section 5.

2 Apprentissage supervisé par induction

Soit $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ une population des individus ou d'objets concernés par le problème d'apprentissage par induction. A cette population est associé un attribut particulier appelé *attribut classe* noté *Y*. A chaque individu ω peut être associé sa classe $Y(\omega)$. On dit que la variable *Y* prend ses valeurs dans l'ensemble des étiquettes, appelé également *ensemble des classes* et noté ***C***.

$$Y : \Omega \rightarrow C = \{c_1, c_2, \ldots, c_m\}$$
$$\omega_i \rightarrow Y(\omega_i) = c_j$$

D'après Zighed et Rakotomalala (2000), si la population Ω est celle des observations des prévisions météorologiques et *Y* le résultat du comportement par rapport à un jeu : *Joué* noté c_1 ou *ne pas joué* noté c_2 ; alors $Y(\omega)$ sera le comportement *affecté* à l'observation ω. La détermination du modèle d'affectation φ est liée à l'hypothèse selon laquelle les valeurs prises par la variable *Y* ne relèvent pas du hasard, mais de certaines situations particulières que l'on peut caractériser. Pour cela l'expert du domaine concerné établit une liste à priori de

variables statistiques, appelées *variables exogènes* et notées pour chaque $\omega \in \Omega$ par : $X(\omega) = (X_1(\omega), X_2(\omega), .., X_p(\omega))$.

Les variables exogènes prennent leurs valeurs dans un domaine de représentation noté $\boldsymbol{D}$ qui ne possède pas de structure mathématique particulière : $X : \Omega \rightarrow D$. Le tableau 1 récapitule un ensemble de 4 variables exogènes.

X_j (l_j)	Signification	Modalité ou valeur
X_1 ($l_1 = 3$)	Ciel	Soleil, Couvert, Pluie
X_2 ($l_2 = 3$)	Température	Chaude, Tiède, Fraîche
X_3 ($l_3 = 2$)	Humidité	Elevée, Normale
X_4 ($l_4 = 3$)	Vent	Fort, Faible

TAB. 1 – *Notation et définition d'une variable exogène.*

La valeur prise par $X_j(\omega)$ est appelée la *modalité* ou la *valeur* de la variable X_j pour chaque individu ω. Nous désignons par l_j le nombre des différentes modalités affectées à la variable X_j. Pour mieux illustrer cette forme de notation, considérons le problème des prévisions météorologiques et supposons qu'une observation peut être décrite par quatre variables exogènes. Dans ce cas, la population des observations concernées par l'apprentissage par induction est un ensemble de tuples composés des quatre variables exogènes $\{X_1, X_2, X_3, X_4\}$ et leurs classes *jouer* ou *ne pas jouer*.

L'objectif est de rechercher un modèle φ de prédiction, appelé aussi d'affectation ou de classification, permettant, pour une observation ω issu de Ω, pour laquelle nous ne connaissons pas la classe $Y(\omega)$ mais dont nous connaissons l'état de toutes ses variables exogènes $X(\omega)$, de prédire cette valeur grâce à φ. La mise au point de φ nécessite de prélever dans la population Ω deux échantillons notés Ω_A et Ω_T. Le premier dit d'*apprentissage* servira à la construction de φ et le second dit de *test* servira à tester la validité de φ (Zighed et Rakotomalala, 2000). Ainsi, pour tout individu $\omega \in (\Omega_A \cup \Omega_T)$, nous supposons connues à la fois ses valeurs $X(\omega)$ dans l'espace de représentation D et sa classe $Y(\omega)$ dans l'espace des étiquettes C. Si φ est jugée cohérente, alors nous pourrons généraliser son emploi à toutes les observations de la population Ω. Ainsi, grâce à φ, nous pourrons calculer $Y(\omega)$, pour chaque $\omega \in \Omega - (\Omega_A \cup \Omega_T)$, connaissant seulement $X(\omega)$.

L'apprentissage automatique supervisé par induction se propose donc de fournir des outils permettant d'extraire, à partir de l'information dont on dispose sur l'échantillon d'apprentissage Ω_A, le modèle de prédiction φ. Ce modèle φ peut prendre la forme d'un réseau de neurones φ^{RN}(Atmani et Beldjilali, 2007a), d'un graphe d'induction φ^{GI} (Kohavi et Quinlan, 2002) ou d'un automate cellulaire φ^{AC} (Atmani et Beldjilali, 2007b). WCSS est basé sur le modèle φ^{AC}.

Le processus général d'apprentissage comporte généralement trois étapes que nous récapitulons ci-dessous :

1. **Elaboration du modèle:** C'est l'étape qui fait appel à un échantillon d'apprentissage noté Ω_A, dont tous les individus sont décrits dans un espace de représentation noté $\boldsymbol{D}$ et appartiennent à l'une des $\boldsymbol{m}$ classes notées c_j ; $j = 1, \ldots, m$. Il s'agit de construire l'application φ qui permet de calculer la classe à partir des différentes représentations des individus ω_i : appelés exemples.
2. **Validation du modèle:** Il s'agit de vérifier, sur un échantillon test Ω_T et dont nous connaissons pour chacun de ses individus, la représentation et la classe, si,

le modèle de prédiction φ issue de l'étape précédente donne bien la classe attendue.

3. **Généralisation du modèle :** C'est l'étape qui consiste à étendre l'application du modèle à tous les individus de la population Ω.

Le processus général d'apprentissage que notre système cellulaire WCSS applique à une population Ω est organisé sur quatre étapes :

1. Acquisition et préparation des données par Weka qui consiste à utiliser les différentes techniques de prétraitement des données déjà intégrées dans l'environnement Weka ;
2. Elaboration du Modèle φ^{AC} par ACSIPINA qui se résume sur 4 étapes :
   a. Initialisation du graphe d'induction par automate cellulaire (coopération *COG* et *CIE*) ;
   b. Génération des règles de production (coopération *COG* et *CIE*) ;
   c. Validation des règles cellulaires (coopération *CV* et *CIE*) ;
3. Validation du Modèle φ^{AC} par Weka qui consiste à exploiter toutes les méthodes de visualisation et d'analyse déjà intégrées dans la plate forme Weka.

La figure 1 récapitule le diagramme général de notre système cellulaire WCSS.

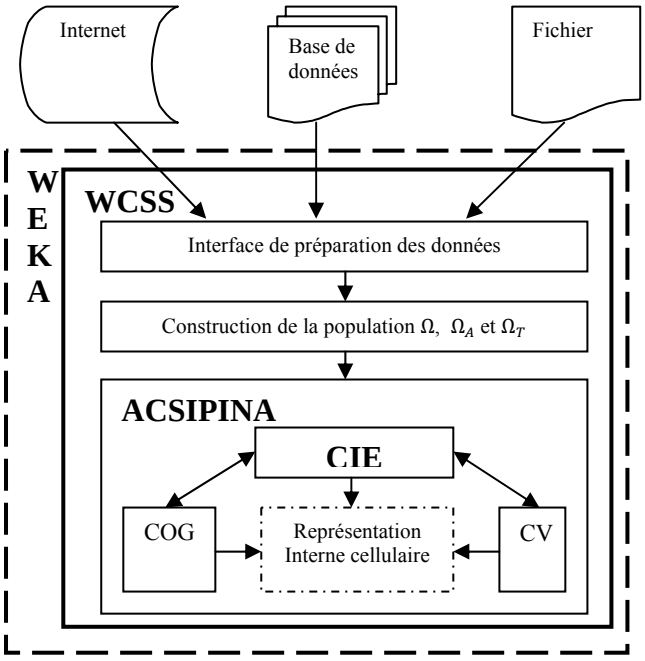


FIG. 1 – *Diagramme général du système WCSS.*

3 Apprentissage symbolique à partir de données

L'algorithme de la méthode *SIPINA* (Zighed, 1996) est une heuristique non arborescente pour la construction d'un graphe d'induction. Son principe consiste à générer une succession

de partitions par fusion et/ou éclatement des nœuds du graphe. Son objectif est d'optimiser un critère τ_λ. Dans ce qui suit nous allons décrire le déroulement du processus sur un exemple fictif. Supposons que notre échantillon d'apprentissage Ω_A se compose de 14 observations des prévisions météorologiques qui se répartissent en deux classes *«oui* pour *joué »* et *«non* pour *ne pas joué »* (voir table 2). A partir de l'échantillon Ω_A la méthode *SIPINA* commence le traitement symbolique pour la construction du graphe d'induction :

- Choisir la mesure d'incertitude (Shannon ou quadratique).
- Initialiser les paramètres λ, μ et la partition initiale S_0.
- Appliquer la méthode *SIPINA* pour passer de la partition S_t à S_{t+1} et générer le graphe d'induction.
- Enfin, génération des règles de prédiction qui sont en générale disjonctives-conjonctives.

Les paramètres λ, μ, les partitions et toutes les autres notions utilisées dans ce processus, sont présentés à l'aide d'exemples dans l'ouvrage de Zighed et Rakotomalala (2000).

ω	$X_1(\omega)$	$X_2(\omega)$	$X_3(\omega)$	$X_4(\omega)$	$Y(\omega)$
1	*Soleil*	*Chaude*	*Elevée*	*Faible*	*non*
2	*Soleil*	*Chaude*	*Elevée*	*Fort*	*non*
3	*Couvert*	*Chaude*	*Elevée*	*Faible*	*oui*
4	*Pluie*	*Tiède*	*Elevée*	*Faible*	*oui*
5	*Pluie*	*Fraîche*	*Normale*	*Faible*	*oui*
6	*Pluie*	*Fraîche*	*Normale*	*Fort*	*non*
7	*Couvert*	*Fraîche*	*Normale*	*Fort*	*oui*
8	*Soleil*	*Tiède*	*Elevée*	*Faible*	*non*
9	*Soleil*	*Fraîche*	*Normale*	*Faible*	*oui*
10	*Pluie*	*Tiède*	*Normale*	*Faible*	*oui*
11	*Soleil*	*Tiède*	*Normale*	*Fort*	*oui*
12	*Couvert*	*Tiède*	*Elevée*	*Fort*	*oui*
13	*Couvert*	*Chaude*	*Normale*	*Faible*	*oui*
14	*Pluie*	*Tiède*	*Elevée*	*Fort*	*non*

TAB. 2 – *L'échantillon d'apprentissage* Ω_A.

3.1 Définition d'une partition

La partition initiale S_0 comporte un seul élément noté s_0, qui regroupe tout l'échantillon d'apprentissage avec 9 individus appartenant à la classe *« oui »* et 5 appartenant à la classe *« non »*. La partition suivante S_1 est engendrée par la variable X_3 et les individus dans chaque nœud s_D sont définis comme suit :

- $s_1 = \{\omega \in \Omega_A | X_3(\omega) = Elevée\}$ et $s_2 = \{\omega \in \Omega_A | X_3(\omega) = Normale\}$.

De même que dans le nœud s_0, on distingue dans s_1 et s_2, les individus des deux classes. La figure 2, extraite de l'interface graphique développée spécialement pour WCSS et intégrée avec succès dans Weka, résume les étapes de construction des sommets s_0, s_1 et s_2.

De la partition S_1, le processus est réitéré à la recherche d'une partition S_2 qui serait meilleure selon le critère τ_λ. Sur la figure 3, sont résumées les principales étapes de construction du graphe d'induction généré par le principe ACSIPINA proposé par Atmani et Beldjilali (2007b). Regardons ce graphique comme s'il s'agissait d'un résultat final sans se

préoccuper de vérifier dans le détail tous les calculs qui ont conduit à ce graphe. La partition suivante S_2 est engendrée par la variable X_1. La partition suivante S_3 est obtenue par la fusion des sommets s_2 et s_4. Enfin, la partition suivante S_4 est engendrée par la variable X_4.

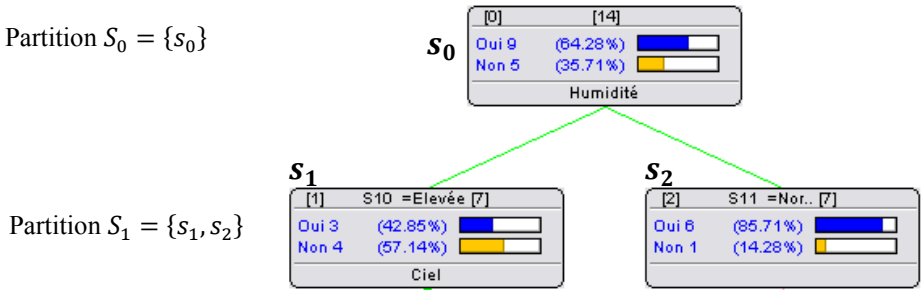


FIG. 2 – *Construction des sommets* s_0, s_1 *et* s_2.

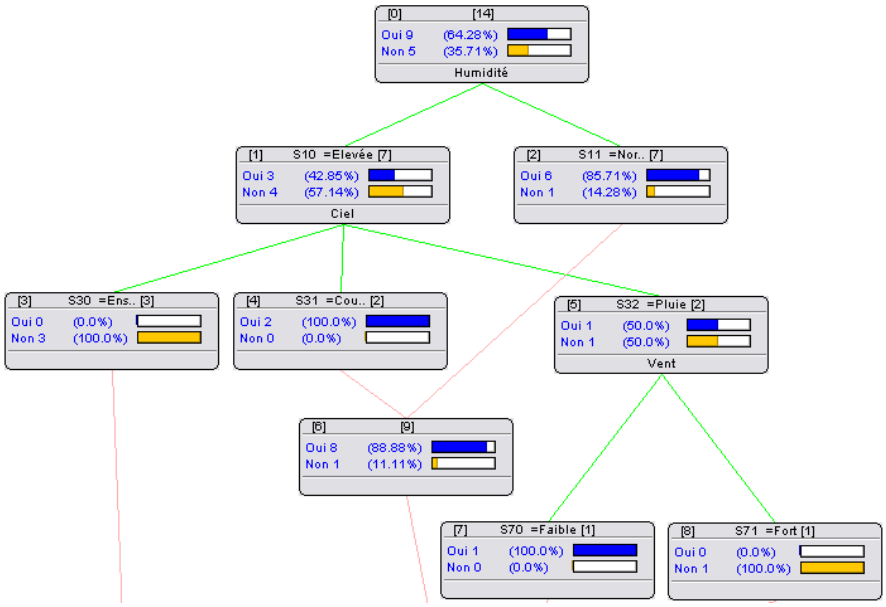


FIG. 3 – *Construction des partitions* S_0, S_1, S_2, S_3 *et* S_4.

3.2 Génération des règles disjonctives-conjonctives

Sous réserve que notre échantillon soit représentatif de la population originelle nous pouvons donc déduire quatre règles R_1, R_2, R_3 et R_4 qui sont de la forme: Si *Condition* Alors *Conclusion*. La *Condition* est une expression logique composée de disjonction de

conjonction que l'on nommera *Prémisse* et *Conclusion* la classe majoritaire dans le sommet décrit par la condition. Par exemple, dans la figure 2, la classe majoritaire de s_1 est 4 (classe *« non »*), par contre la classe majoritaire de s_2 est 6 (classe *« oui »*). Les quatre règles de classification issues du modèle graphe d'induction de la figure 3 sont :

1. Si (Humidité=élevée) et (Ciel=soleil) alors $c_2 = non$; « selon s_3 ».
2. Si ((Humidité=élevée) et (Ciel=couvert)) ou ((Humidité=normale)) alors $c_1 = oui$; « selon le sommet final s_6 obtenu par fusion ».
3. Si (Humidité=élevée) et (Ciel=pluie) et (Vent=faible) alors $c_1 = oui$;
4. Si (Humidité=élevée) et (Ciel=pluie) et (Vent=fort) alors $c_2 = non$.

4 Apprentissage cellulaire à partir des données

4.1 Moteur d'inférence cellulaire –*CIE*–

Le module *CIE* (*Cellular Inference Engine*) simule le fonctionnement du cycle de base d'un moteur d'inférence en utilisant deux couches finies d'automates finis. La première couche, *CELFACT*, pour la base des faits et, la deuxième couche, *CELRULE*, pour la base de règles. Chaque cellule au temps *t+1* ne dépend que de l'état des ses voisines et du sien au temps *t*. Dans chaque couche, le contenu d'une cellule détermine si et comment elle participe à chaque étape d'inférence : à chaque étape, une cellule peut être active (1) ou passive (0), c'est-à-dire participe ou non à l'inférence.

Atmani et Beldjilali (2007b) ont supposés qu'il y a *l* cellules dans la couche *CELFACT*, et *r* cellules dans la couche *CELRULE*. Toute cellule *i* de la première couche *CELFACT* est considérée comme fait établi si sa valeur est 1, sinon, elle est considérée comme fait à établir. Toute cellule *j* de la deuxième couche *CELRULE* est considérée comme une règle candidate si sa valeur est 1, sinon, elle est considérée comme une règle qui ne doit pas participer à l'inférence.

Les états des cellules se composent de trois parties : *EF*, *IF* et *SF*, respectivement *ER*, *IR* et *SR*, sont l'entrée, l'état interne et la sortie d'une cellule de *CELFACT*, respectivement d'une cellule de *CELRULE*. L'état interne, *IF* d'une cellule de *CELFACT* indique le rôle du fait : dans le cas d'un graphe d'induction *IF* = 0 correspond à un fait du type sommet (s_i), *IF* = 1 correspond à un fait du type *attribut=valeur* ($X_i = valeur$). Pour une cellule de *CELRULE*, l'état interne *IR* peut être utilisé comme coefficient de probabilité que nous n'aborderons pas dans cet article[1].

Pour illustrer l'architecture et le principe de fonctionnement du module *CIE*, cœur du module ACSIPINA, nous considérons la partie du graphe, extraite de la figure 3, obtenue en utilisant les partitions $S_0 = \{s_0\}$, $S_1 = \{s_1, s_2\}$ et $S_2 = \{s_3, s_4, s_5\}$ (voir la figure 4).

La figure 5 montre comment la base de connaissance extraite à partir de ce graphe d'induction est représentée par les couches *CELFACT* et *CELRULE*. Initialement, toutes les entrées des cellules dans la couche *CELFACT* sont passives (*EF* = 0), exceptées celles qui représentent la base des faits initiale (*EF*(1) = 1).

[1] Cas où la variable cible *Y(w)* à prédire a des modalités non équilibrées, priorité entre règles, etc…

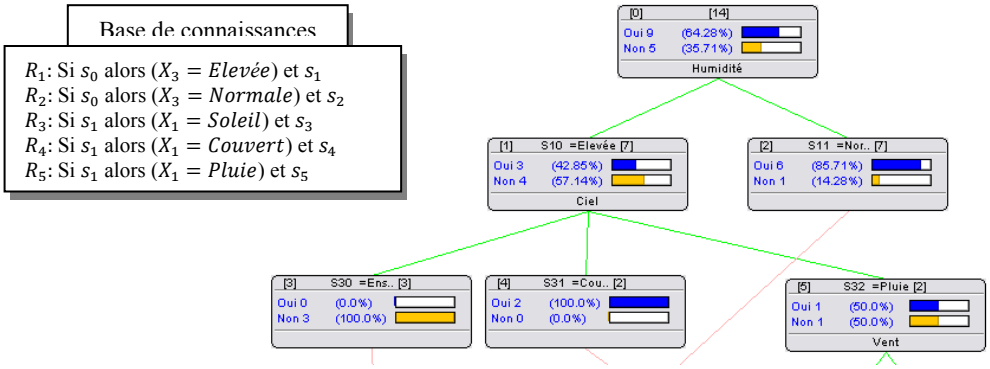


FIG. 4 – *Les partitions S_0, S_1 et S_2.*

Les matrices d'incidence R_E et R_S représentent la relation *entrée/sortie* des Faits et sont utilisées en *chaînage avant*. On peut également utiliser R_S comme relation d'entrée et R_E comme relation de sortie pour lancer une inférence en chaînage arrière. Notez qu'aucune cellule du voisinage d'une cellule qui appartient à *CELFACT* (respectivement à *CELRULE*) n'appartient pas à la couche *CELFACT* (respectivement à *CELRULE*).

CELFACT(Faits)	*EF*	*IF*	*SF*
s_0	1	0	0
$X_3 = Elevée$	0	1	0
s_1	0	0	0
$X_3 = Normale$	0	1	0
s_2	0	0	0
$X_1 = Soleil$	0	1	0
s_3	0	0	0
$X_1 = Couvert$	0	1	0
s_4	0	0	0
$X_1 = Pluie$	0	1	0
s_5	0	0	0

CELRULE (Règles)	*ER*	*IR*	*SR*
R_1	0	1	1
R_2	0	1	1
R_3	0	1	1
R_4	0	1	1
R_5	0	1	1

FIG. 5 – *Représentation cellulaire de la Base des connaissances de la figure 4.*

Le voisinage est introduit par la notion de matrice d'incidence. Dans la figure 6 sont respectivement représentées les matrices d'incidence d'entrée R_E et de sortie R_S de l'automate cellulaire. La relation d'entrée, notée iR_Ej, est formulée comme suit : $\forall i \in [1, l]$, $\forall j \in [1, r]$, si (le Fait $i \in$ à la *Prémisse* de la règle j) alors $R_E(i, j) \leftarrow 1$. De même la relation de sortie, notée iR_Sj, est formulée comme suit : $\forall i \in [1, l]$, $\forall j \in [1, r]$, si (le Fait $i \in$ à la *Conclusion* de la règle j) alors $R_S(i, j) \leftarrow 1$.

Pour définir la dynamique du CIE, nous allons rappeler que le cycle de base d'un moteur d'inférence, pour établir un fait F en chaînage avant, fonctionne traditionnellement comme suit :

- ✓ Recherche des règles applicables (évaluation et sélection) ;
- ✓ Choisir une parmi ces règles, par exemple R (filtrage) ;
- ✓ Appliquer et ajouter la partie conclusion de R à la base des faits (exécution).

Le cycle est répété jusqu'à ce que le fait *F* soit ajouté à la base des faits, ou s'arrête lorsqu'aucune règle n'est applicable.

$\boldsymbol{R_E}$	R_1	R_2	R_3	R_4	R_5
s_0	1	1			
$X_3 = Elevée$					
s_1			1	1	1
$X_3 = Normale$					
s_2					
$X_1 = Soleil$					
s_3					
$X_1 = Couvert$					
s_4					
$X_1 = Pluie$					
s_5					

$\boldsymbol{R_S}$	R_1	R_2	R_3	R_4	R_5
s_0					
$X_3 = Elevée$	1				
s_1	1				
$X_3 = Norma e$		1			
s_2		1			
$X_1 = Soleil$			1		
s_3			1		
$X_1 = Couvert$				1	
s_4				1	
$X_1 = Pluie$					1
s_5					1

FIG. 6 – *Les matrices d'incidences d'entrée R_E et de sortie R_S pour la figure 4.*

La dynamique de l'automate cellulaire *CIE*, pour simuler le fonctionnement d'un *Moteur* d'*Inférence cellulaire*, utilise deux fonctions de transitions δ_{fact} et δ_{rule}, où δ_{fact} correspond à la phase d'*évaluation*, de *sélection* et de *filtrage*, et δ_{rule} correspond à la phase d'*exécution*. La fonction de transition δ_{fact}:

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{fact}} (EF, IF, \boldsymbol{EF}, \boldsymbol{ER} + (\boldsymbol{R_E^T} \cdot \boldsymbol{EF}), IR, SR)$$

La fonction de transition δ_{rule}:

$$(EF, IF, SF, ER, IR, SR) \xrightarrow{\delta_{rule}} (\boldsymbol{EF} + (\boldsymbol{R_S} \cdot \boldsymbol{ER}), IF, SF, ER, IR, \overline{\boldsymbol{ER}})$$

Où la matrice R_E^T désigne la transposé de la matrice R_E.

Nous considérons G_0 la configuration initiale de l'automate cellulaire et, que $\Delta = \delta_{rule} \circ \delta_{fact}$ la fonction de transition globale : $\Delta(G_0) = G_1$ si $G_0 \xrightarrow{\delta_{fact}} G'_0$ et $G'_0 \xrightarrow{\delta_{rule}} G_1$

Supposons que $G = \{G_0, G_1, \ldots, G_q\}$ est l'ensemble des configurations de notre automate cellulaire. L'évolution discrète de l'automate, d'une génération à une autre, est définie par la séquence G_0, G_1, ..., G_q, où $G_{i+1} = \Delta(G_i)$.

4.2 Elaboration du modèle φ^{AC} par ACSIPINA

L'élaboration du modèle φ^{AC} par ACSIPINA se résume comme suit :

- **Initialisation du graphe d'induction par automate cellulaire :** c'est le rôle du module *COG* en coopération avec *CIE* ; Cela revient à initialiser les couches *CELFACT* et *CELRULE* et générer les matrices d'incidences R_E et R_S. Le résultat de cette phase est illustré par les figures 5 et 6.
- **Génération des règles de production conjonctives :** C'est toujours une coopération entre les modules *COG* et *CIE* ; Pour produire automatiquement des règles conjonctives, le module *COG* coopère avec le moteur d'inférence cellulaire (*CIE*) qui utilise les mêmes fonctions de transition δ_{fact} et δ_{rule} avec la permutation de R_E et de R_S. En partant du noeud terminal vers la racine s_0, et en utilisant le moteur d'inférence cellulaire *CIE* en chaînage arrière, avec le mode asynchrone en profondeur.
- **Validation des règles cellulaires :** En se basant sur la base de règle générées pendant la phase précédente, le module *CV* initialise à nouveau les couches *CELFACT* et *CELRULE*

et génère les deux matrices R_E et R_S. Sur l'accomplissement de ce processus, le module *CV* (voir figure 1) est prêt à lancer la phase de validation. En employant le même principe de base du moteur d'inférence cellulaire *CIE*, et les mêmes fonctions de transition δ_{fact} et δ_{rule}, l'automate cellulaire avance d'une configuration vers une autre, dans le but de produire l'ensemble Ω_E (prévision fausse).

5 Les résultats expérimentaux obtenus avec WCSS

Quand il s'agit de l'apprentissage par induction cellulaire, nous devons impérativement passer par les étapes suivantes :

1. Importer les données dans le système WCSS (voir figure 1) ;
2. Définir le problème à résoudre, c'est-à-dire sélectionner les descripteurs et la variable à prédire ;
3. Subdiviser les données en ensemble d'apprentissage Ω_A et de test Ω_T ;
4. Lancer l'induction en utilisant le principe de *ACSIPINA* sur les données d'apprentissage Ω_A:
   - Choisir la mesure d'incertitude (*Shannon* ou *Quadratique*).
   - Initialiser les paramètres λ, μ et la partition initiale *S*0.
   - Initialiser les couches *CELFACT* et *CELRULE* ;
   - Appliquer le principe *ACSIPINA* pour passer de la partition *St* à *St*+1 et générer le graphe d'induction cellulaire.
5. Evaluer le modèle cellulaire par déduction sur l'ensemble test Ω_T.

Une manière classique d'évaluer la qualité de l'apprentissage du système WCSS est de confronter la prédiction du modèle avec les valeurs observées sur un échantillon de la population. Cette confrontation est résumée dans un tableau croisé appelé matrice de confusion (voir table 3). Il est possible d'en extraire des indicateurs synthétiques, le plus connu étant le taux d'erreur ou taux de mauvais classement qu'on note ξ.

Nous pouvons donc dire qu'en classant un individu pris au hasard dans la population, nous avons 8.89 chances sur 100 de réaliser une mauvaise affectation. Le principe intérêt du taux d'erreur est qu'il est objectif ; il sert généralement à comparer les méthodes d'apprentissage sur un problème donné. Pour obtenir un indicateur non biaisé, il est impératif, en pratique, de ne pas le mesurer sur l'échantillon qui a servi à élaborer le modèle. A cet effet, le praticien met souvent de coté un échantillon, dit de test, qui servira à évaluer et à comparer les modèles.

ξ=0.0889	***Type 1***	***Type 2***	***Total***
Type 1	*835*	*39*	*874*
Type 2	*91*	*496*	*587*
Total	*926*	*535*	*1461*

TAB. 3 – *Matrice de confusion.*

La validation par le module CV est la phase qui consiste à calculer le taux ξ, sur un échantillon test Ω_E, en utilisant les mêmes fonctions de transition δ_{fact} et δ_{rule}. La généralisation est la dernière phase qui consiste à calculer de nouveau la valeur de ξ en appliquant le modèle à tous les individus de la population $\Omega = \Omega_A + \Omega_T$. La généralisation a

été effectuée sur plusieurs bases de grande taille en particulier : diabetes avec 1461 individus contenant 10 variables dont 2 continues ; breast avec 699 individus contenant seulement des variables continues ; et une troisième base de 2201 individus contenant seulement des variables discrètes.

Enfin, pour évaluer notre système WCSS, nous avons utilisé plusieurs méthodes d'apprentissage automatique par induction déjà implémenté dans la plate forme Weka, à savoir quatre algorithmes (*ID3*, *C4.5*, *CART* and *Kppv*), pour conduire un autre type de généralisation. Nous avons obtenu, après filtrage avec Neuro-IG (Atmani et Beldjilali, 2007a), les résultats de comparaison que nous avons résumée dans la table 4.

	WCSS	***Weka (plate forme de fouille de données)***			
	ξ	***ξ (ID3)***	***ξ (C4.5)***	***ξ (CART)***	***ξ (kppv)***
diabetes	*0.0082*	*0.0130*	*0.0034*	*0.0034*	*0.0185*
breast	*0.0100*	*0.0758*	*0.0358*	*0.0572*	*0.0272*
titanic	*0.1814*	*0.2240*	*0.2240*	*0.2240*	*0.2240*

TAB. 4 – *Les résultats de comparaison.*

Les expériences ont prouvé que la nouvelle représentation booléenne permet l'optimisation du graphe d'induction et, n'influe pas sur le taux de classification. L'utilisation du système WCSS permet de réduire la taille de stockage du graphe d'induction de plus de 50% ainsi que le temps de validation : utilisation de δ_{fact} et δ_{rule} .

6 Conclusion

Deux motivations concurrentes nous ont amenés à adopté le principe des automates cellulaire, proposé par Atmani et Beldjilali (2007b), pour la génération, la représentation, l'optimisation et l'utilisation d'une base de Règles. En effet, nous avons non seulement souhaité avoir une base de règles optimale, mais aussi, nous avons également souhaité améliorer l'acquisition automatique des règles pour alimenter un système expert en intégrant cette nouvelle technique cellulaire dans la plate forme Weka. Les avantages de cette méthode basée sur l'automate cellulaire peuvent être récapitulés comme suit :

– L'acquisition de l'information ainsi que son contrôle sont simples, sous forme de matrices binaires exigeant un prétraitement minimal.

– La facilité de l'implémentation des fonctions de transition δ_{act} et δ_{rule} qui sont de basse complexité, efficaces et robustes pour des valeurs extrêmes. D'ailleurs, elles sont bien adaptées aux situations avec beaucoup d'attributs.

– Les résultats sont simples pour être insérer et utiliser par un système expert.

– Le système de prédiction obtenu est un modèle cellulaire composé d'un ensemble simple de fonctions de transition et de règles de production, qui permettent non seulement de décrire le problème actuel mais d'établir également une fonction de classification pour la prévision.

– La matrice d'incidence, R_E, facilite la transformation de règles dans des expressions équivalentes booléennes, qui nous permet d'utiliser l'algèbre de Boole élémentaire pour examiner d'autres simplifications[2].

[2] Fait l'objet d'un autre article.

Références

Atmani, B., Beldjilali, B. (2007a). *Neuro-IG : A Hybrid System for Selection and Elimination of Predictor Variables and non Relevant Individuals*. Informatica, Journal International, Vol. 18, N°2 163-186.

Atmani, B., Beldjilali, B. (2007b). *Knowledge Discovery in Database: Induction Graph and Cellular Automaton*. Computing and Informatics Journal, Vol.26, N°2 171-197.

Clark, P. (1989). *Knowledge representation in machine learning*, Machine and Human Learning, Eds : Y. Kodratoff and A. Hutchison, London.

Clark, P. (1990). *Machine learning, techniques and recent developments*, In AR Mirzai, Editor, Artificial Intelligence : Concepts and Applications in Engineering, Chapman and Hall 655-93.

Clark, P., Boswell, R. (1991). *Rules induction with CN2, Some recent improvements*, In ML-Procedding of the 5th European Conference, 151-163.

Kohavi, R., Quinlan, J. (2002). *Decision tree discovery*, in Handbook of Data Mining and Knowledge Discovery, Klosgen and Zytkow Editors 267-276.

Lefebure, R., Venturi, G. (2001). *Data Mining*, Paris, EYROLLES.

Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N. (1986). *The multi-purpose incremental learning system AQ15 and its testing application to three medical domains*, In Procedding of AAAI-86, Philadelphia 1041-1045.

Quinlan, J.R. (1986). *Induction of Decision Trees*. Machine Learning.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Rabaseda, S., Zighed, D.A (1996). *Generation and simplification of rules in graphs of induction*, Acts of the 25th symposium of the economic structures, Econometrics and data processing.

Rakotomalala, R., Zighed, D.A. (1999). *Feschet, Characterization of production rules in a process of induction*, Hermes Science Publication, Paris.

Rivest, R. (1987). *Learning Decision Lists*, Machine Learning, 2, 223-246.

Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G. et Cunningham, S.J. (1999). "*Weka: Practical machine learning tools and techniques with Java implementations,"* in H. Kasabov and K. Ko, eds., ICONIP/ ANZIIS/ ANNES '99 International Workshop, Dunedin.

Zighed, D.A. (1996). *SIPINA for Windows, ver 2.5*. Laboratory ERIC, University of lyon 2.

Zighed, D.A., Rakotomalala, R. (2000). *Graphs of induction, Training and Data Mining*, Hermes Science Publication, 21-23.

Summary

WCSS is a Cellular knowledge management system dedicated to the data mining. WCSS addresses two types of audiences. On the one hand, it exploits the data mining environment provided by the Weka platform which makes it accessible to use a type responsible for studies on real data. On the other hand, it generated a conjonctive rules, it lends itself to direct use with conventional systems experts. WCSS is written in Java and based on the induction graphs generated by the principle of the SIPINA method.

Décision floue et modèle de simulation des piétons virtuels

Meriem Mandar , Azedine Boulmakoul

Faculté des Sciences et Techniques de Mohammedia
FSTM – Département informatique - B.P. 146 Mohammedia Maroc
azedine.boulmakoul@yahoo.fr
meriem.mandar@gmail.com

Résumé. Dans ce papier, la décision floue et les techniques de simulation à base d'automates cellulaires sont utilisées afin de modéliser la dynamique des piétons virtuels, en se basant sur le modèle de Schadschneider. Aux principes de ce modèle de base, nous intégrons un processus de calcul de la matrice de préférence fondé sur le modèle du choix discret en vue de minimiser le coût d'atteignabilité de la destination. La fuzzification du modèle a été aussi proposée pour permettre aussi de mieux modéliser la préférence et la trace chimique.

1 Introduction

Les architectes et planificateurs urbains sont souvent confrontés au problème d'évaluation de la manière dont leurs architecture ou conception affecteront le comportement des individus. Une façon de traiter ce problème est de développer des modèles reliant les comportements des piétons aux paramètres de conception d'architecture (voir Dijkstra *et al.* 2000). Plusieurs modèles de simulation ont été proposés afin de prédire le comportement des piétons dans des situations normales aussi bien que dans celles de panique. Ces modèles peuvent être classés en trois catégories. La première traite les individus d'une manière uniforme, en les comparants aux particules par analogie avec les fluides ou avec la dynamique des gaz. La deuxième approche consiste à discrétiser l'espace en cellules, où le déplacement des individus et l'occupation physique d'espace sont déterminés par les règles de comportement. Finalement la dernière catégorie est basée sur un environnement continu et décrit le comportement des piétons par des forces qui le régissent. Dans l'espace des automates cellulaires, le temps et l'état des variables sont discrets. Ceci permet d'avoir des simulations performantes. Les automates cellulaires sont fondés sur le principe des automates occupant des cellules selon un ensemble de règles locales, pouvant décrire le comportement de chaque automate et créant une approximation du comportement individuel. Le comportement collectif émergeant est un résultat de l'interaction de la règle de la micro-simulation sur les voisinages locaux. Les modèles de simulation traditionnels appliquent des équations plutôt que des règles comportementales. L'auto-organisation dans le comportement de la «Vie Artificielle» provient de la décentralisation des sources de prise de décision. La simulation des piétons par les automates cellulaires est une approche parallèle et distribuée. Les modèles simples de la simulation de type «automates cellulaires» sont capables de capturer les traits

essentiels du système (Back 1996). La structure du papier est organisée comme suit. Après une introduction, la section 2 aborde l'état de l'art en matière de simulation des piétons virtuels. La section 3 propose des rappels succincts de la théorie des ensembles flous. Dans la section 4 est formulé le modèle de simulation que nous avons élaboré. La section 5 traite les aspects conception du simulateur. Enfin la section 6 propose les conclusions et les perspectives de ce travail.

2 Etat de l'art – Piétons virtuels

L'état de l'art de la modélisation des comportements des piétons regroupe trois types de modèles : (a) les modèles de files d'attente dont les travaux sont donnés par (Constantine, 2006). Dans ces modèles des temps d'attente sont introduits pour la considération des phénomènes de congestion dus à une demande du trafic plus grande que la capacité de l'infrastructure. Les travaux développés par (Cavens et al. 2005) utilisent la simulation développée par (Helbing, 2002). Ces modèles servent à décrire le phénomène d'évacuation des piétons. (b) Les modèles des automates cellulaires discrétisent l'espace en treillis de cellules identiques, ayant chacune un ensemble de règles régissant leur état à chaque pas de temps. Nagel et Schreckenberg (2002) ont introduit le modèle N-S. Il décrit les déplacements des véhicules en utilisant un ensemble de règles dynamiques données ci-dessous :

1. l'accélération $v_i \leftarrow \min(v_i + 1, v_{\max})$;
2. le freinage $v_i \leftarrow \min(v_i, g_i - 1)$;
3. Randomisation $v_i \leftarrow \max(v_i - 1, 0)$, avec une probabilité p ;
4. et le déplacement $x_i \leftarrow x_i + v_i$

Le modèle de Blue et Alderb (2000) définit trois éléments fondamentaux du mouvement des piétons : (i) le pas à effectuer, (ii) le mouvement désiré (accélération ou freinage), (iii) la gestion de conflits. Alors que Zhang et Wang (2004) ont introduit une notion de points d'arrêts dans un trafic mixte. A chaque période de temps, un mouvement est choisit selon une probabilité de transition. Laquelle est composée de deux parties, une statique incluant les priorités de la prochaine direction, l'effet d'obstacle, et l'existence d'un refuge pour les piétons qu'ils atteignent dans le cas où l'état de leur feu de circulation est au rouge. La partie dynamique inclue l'effet du feu de circulation des piétons, l'occupation de la cellule destination, et l'état du point d'arrêt correspondant. La probabilité de transition de la simulation développée par Kirchner et Schadschneider (2002) dépend des composants statiques et dynamiques Kd et Ks définissant l'espace. La variation de ces paramètres influence le temps d'évacuation des piétons. Le modèle de l'utilité aléatoire développé par Antonini et al. (2004) combine la théorie du choix discret, la modélisation du comportement des piétons et les techniques de traitement d'images. Ce modèle permet de simuler le problème de détection des piétons dans des scénarios aussi réels et complexes que possible. Le terme choix discret provient d'une discrétisation du comportement des piétons en un ensemble de choix où l'alternative avec la plus haute utilité est sélectionnée. Mauron (2002) a développé dans sa thèse deux approches pour simuler le comportement des piétons face aux obstacles. La première utilise l'idée de base de Hogendoorn et al. (2002). Les cellules sont de différents types (lien, traversée, bordure, obstacle). Les forces et les directions de marche de chaque cellule,

sont calculées en début de simulation. La seconde simule l'environnement avec un ensemble de points de liens. Les agents utilisent ces points pour trouver le chemin à leurs destinations.

(c) Les modèles continus sont basés sur des équations différentielles. Le modèle de force sociale développé par Helbing et al. (2000) repose sur l'introduction de forces décrivant le mouvement des piétons. L'équation de la motivation totale $\vec{F}_\alpha(t)$ d'un piéton est la somme des termes d'accélération, effets répulsifs des autres piétons et des obstacles ainsi que des effets attractifs. Le travail présenté par Kirchner et al. (2003) repose sur les systèmes multi agents, se répartissant en plusieurs modules communicants entre eux via un réseau. Ces modules se groupent en trois catégories : (a) des modules mentaux déterminent le comportement des agents pour atteindre leurs buts en se basant sur leurs propres expériences ; (b) des modules de simulation physique régissent les interactions de l'agent avec son environnement physique ; (c) un module de contrôle chargé de coordonner la communication entre les autres modules et de garder la trace de l'état de simulation. Le modèle MISC (Lacroix *et al.* 2006) tente de simuler des déplacements d'individus dans un environnement contraint en mettant le point sur la nécessité de la synthèse entre la gestion du temps et celle de l'espace.

3 Théorie du floue

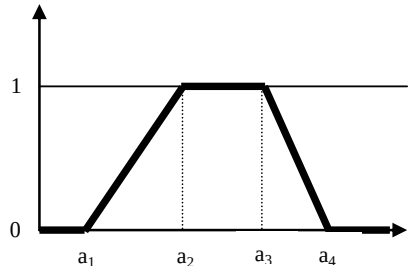
Il y a plusieurs activités humaines dans lesquelles l'imprécision se produit en association avec les quantités numériques. Le concept d'ensemble flou a été présenté par Zadeh (1965) où il a décrit les mathématiques de la théorie des ensembles flous, et par extension la logique floue. Les approches conventionnelles de la représentation de la connaissance manquent des moyens de représenter les concepts flous. Par conséquent, les approches basées sur la logique et la théorie des probabilités classiques ne fournissent pas un cadre conceptuel approprié pour traiter la représentation de la connaissance, puisqu'une telle connaissance est par nature lexicalement imprécise et non catégorique. Cette théorie a proposé de faire opérer les valeurs V et F sur l'intervalle des nombres réels [0,1]. De nouvelles opérations pour les calculs de la logique ont été proposées, et se sont avérées en principe une généralisation de la logique classique.

Définition 1. (Ensemble flou) Soit X un ensemble non vide. Un ensemble flou A définit sur X, est caractérisé par sa fonction d'appartenance : $\mu_A : X \to [0,1]$

$\mu_A(x)$ est perçue comme le degré d'appartenance de l'élément x dans l'ensemble flou A. Il est claire que A est complètement déterminé par l'ensemble des tuples $A = \{(u, \mu_A(u)) \mid u \in X\}$.

Définition 2. (Dubois et Prade, 1983) définissent la fonction d'appartenance d'un Nombre Flou Trapézoïdale (NFT) $\tilde{A}(a_1, a_2, a_3, a_4)$, comme suit:

$$\mu_{\tilde{A}}(x)=\begin{cases} L(x), & Si \quad a_1 \le x < a_2; \\ 1, & Si \quad a_2 \le x < a_3; \\ R(x), & Si \quad a_3 \le x < a_4; \\ 0, & \sin on. \end{cases} \qquad (1)$$



Où L(x) est une fonction croissante monotone continue et $0 \le L(x) \le 1$; et R(x) est une fonction décroissante monotone continue et $0 \le R(x) \le 1$, $a_1 < a_2 \le a_3 < a_4$.

$\forall \alpha \in [0,1]$, le α-coupe de $u_{\tilde{A}}(x)$ est : $I_{\tilde{A}}(\alpha)=\left[L^{-1}(\alpha), R^{-1}(\alpha)\right]$ (2)

Spécialement, si $\alpha = 1$ alors : $I_{\tilde{A}}(1)=\left[L^{-1}(1), R^{-1}(1)\right]=[a_2, a_3]$. (3)

La distance de O à $I_{\tilde{A}}(\alpha)$ est : $d_{I_{\tilde{A}}}=\dfrac{\sqrt{2}\left|L^{-1}(\alpha)R^{-1}(\alpha)\right|}{\sqrt{L^{-1}(\alpha)^2+R^{-1}(\alpha)^2}}$ (4)

En se basant sur la théorie des ensembles flous (FST), on peut obtenir une mesure d'un NFT $\tilde{A}(a_1,a_2,a_3,a_4)$ comme suit :

$$R_{\tilde{A}}=\frac{\int_0^1 \alpha \; d_{I_{A(\alpha)}} d\alpha}{\int_0^1 \alpha \quad d\alpha}=\frac{\sqrt{2}\int_0^1 \dfrac{\alpha\left|L^{-1}(\alpha)R^{-1}(\alpha)\right|}{\sqrt{L^{-1}(\alpha)^2+R^{-1}(\alpha)^2}} \quad d\alpha}{\int_0^1 \alpha \quad d\alpha} \qquad (5)$$

De (5) on obtient une correspondance entre un nombre flou $\tilde{A}(a_1,a_2,a_3,a_4)$ et un nombre réel. Spécialement si $\tilde{A}(a_1,a_2,a_3,a_4)$ est un nombre flou trapézoïdale, $a_1 < a_2 < a_3 < a_4$

Où $L(x)=\dfrac{x-a_1}{a_2-a_1}$, $R(x)=\dfrac{a_4-x}{a_4-a_3}$ (6)

En se basant sur (5) et (6), on peut calculer $R_{\tilde{A}}$ comme suit :

$$R_{\tilde{A}}=\sqrt{2}\left|A\left(\frac{1}{3}\left[(1+2B+D)^{\frac{3}{2}}-D^{\frac{3}{2}}\right]+\left(2E-2D+5BC-11B^2\right)\bullet\left(\sqrt{1+2B+D}-\sqrt{D}\right)\right.\right.$$
$$+(C-3B)\sqrt{1+2B+D}+\frac{1}{2}\left(3CD-5BD-2BE-5B^2C+11B^2\right)$$
$$\left.\left.\bullet\left(Ln\frac{\sqrt{1+2B+D}+1+B}{\sqrt{1+2B+D}-1-B}-Ln\frac{\sqrt{D}+B}{\sqrt{D}-A}\right)\right|\right.$$

Où, $A = \frac{(a_2 - a_1)(a_3 - a_4)}{\sqrt{(a_2 - a_1)^2 + (a_3 - a_4)^2}}$, $B = \frac{a_1(a_2 - a_1) - a_4(a_3 - a_4)}{(a_2 - a_1)^2 + (a_3 - a_4)^2}$,

$C = \frac{a_4(a_2 - a_1) + a_1(a_3 - a_4)}{(a_2 - a_1)(a_3 - a_4)}$, $D = \frac{a_1^2 + a_4^2}{(a_2 - a_1)^2 + (a_4 - a_3)^2}$, $E = \frac{a_2 a_4}{(a_2 - a_1)(a_3 - a_4)}$

En se basant sur (5) on peut identifier une application du Nombre Flou Trapézoïdale F(R) vers $(0,+\infty)$:

$$f : F(R) \to (0,+\infty),$$
$$\tilde{A}(a_1, a_2, a_3, a_4) \to R_{\tilde{A}}$$

Le théorème de décomposition des ensembles flous et la méthode de défuzzification centrale, (Yingchao, 2007) énoncent que la mesure d'un intervalle peut s'étendre à devenir une mesure d'un nombre flou. Cette dernière nous permet de déduire que la comparaison de nombres flous, peut être remplacée par une comparaison des mesures de correspondance.

Si $\tilde{A}(a_1, a_2, a_3, a_4), \tilde{B}(b_1, b_2, b_3, b_4) \in F(R)$

Alors : $\tilde{A}(a_1, a_2, a_3, a_4) \leq \tilde{B}(b_1, b_2, b_3, b_4) \Leftrightarrow R_{\tilde{A}} \leq R_{\tilde{B}}$

3.1 Arithmétique floue

Les opérateurs arithmétiques sur les intervalles flous sont définit comme suit :

Soit $\tilde{A}(a_1, a_2, a_3, a_4), \tilde{B}(b_1, b_2, b_3, b_4), \tilde{C}(c_1, c_2, c_3, c_4) \in F(R)$ alors

1. $\tilde{A} + \tilde{B} = \tilde{C}(a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)$;
2. $\tilde{A} - \tilde{B} = \tilde{C}(a_1 - b_1, a_2 - b_2, a_3 - b_3, a_4 - b_4)$;
3. $\tilde{A} \bullet \tilde{B} = \tilde{C}(\min(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4), \min(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3)$
   $\max(a_2 b_2, a_2 b_3, a_3 b_2, a_3 b_3), \max(a_1 b_1, a_1 b_4, a_4 b_1, a_4 b_4))$;
4. $\tilde{A} / \tilde{B} = \tilde{C}(\min(a_1/b_1, a_2/b_2, a_3/b_3, a_4/b_4), \min(a_2/b_2, a_2/b_3, a_3/b_2, a_3/b_3)$
   $\max(a_2/b_2, a_2/b_3, a_3/b_2, a_3/b_3), \max(a_1/b_1, a_2/b_2, a_3/b_3, a_4/b_4))$.

3.2 Décision floue

Plusieurs problèmes de prise de décision se produisent dans le monde réel, quand les buts et/ou les contraintes sont définis de manière imprécise. Un but flou G peut être défini sur un ensemble X d'alternatives par un ensemble flou G inclus dans X. Par exemple, si on considère que X dénote l'ensemble des nombres réels, alors le but flou exprimé en mot par « u doit être considérablement supérieur à 10 » peut être représenté par un ensemble flou ayant une fonction d'appartenance. De la même manière, les contraintes floues se définissent comme un ensemble flou dans X (Srichander , 1998).

Le problème de prise de décision dans un environnement flou, peut être défini comme l'intersection des buts avec les contraintes. Spécifiquement, si X est un ensemble d'alternatives, alors la décision floue D est définie comme un ensemble flou dans X, donné par $D = G \cap C$. La fonction d'appartenance correspondante est définie par : $\mu_D(x) = \min[\mu_C(x), \mu_G(x)]$

4 Le modèle du piéton virtuel proposé

4.1 Définition du modèle adopté

Le modèle de Schadschneider (2001) utilise des automates cellulaires à deux dimensions $40 \times 40\, cm^2$ pour simuler une foule de piétons. Il définit des sols dynamiques $\tilde{D}_{xy}$ et statiques $\tilde{S}_{xy}$ présentant l'espace aux piétons. Le premier est considéré comme une trace virtuelle laissée par les piétons, tandis que le deuxième définit l'espace en termes d'obstacles. Chaque cellule peut être vide ou occupée par au plus un piéton. La mise à jour est synchrone pour tous les piétons, et le déplacement se fait avec une vitesse d'une cellule par unité de temps.

4.2 Fuzzification du modèle de base

Chaque piéton possède une direction de déplacement préférée. Une matrice de préférence 3×3 est construite, contenant les préférences floues de déplacement du piéton aux 8 cellules voisines (figure 1), l'élément du centre définit la position du piéton. A chaque pas de temps, une nouvelle matrice de préférences est assignée à chaque piéton.

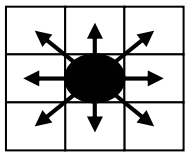


FIG. 1 – *les transitions possibles d'un piéton et sa matrice de préférence*

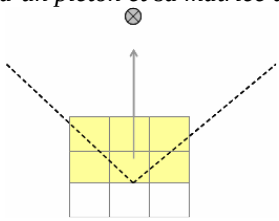


FIG. 2 – *les transitions possibles d'une particule selon son champ de vue.*

Contrairement au sol statique, le sol dynamique change par le mouvement des piétons :

1. Si un piéton a quitté la cellule (x,y), alors $\tilde{D}_{xy}$ correspondant est incrémenté par $\Delta\tilde{D}_{xy}$. Ce dernier étant fixé par le modèle et qui peut être continu ou discret ;
2. pour simuler la diffusion, un certain pourcentage du champ est distribué aux cellules voisines ;
3. la valeur $\tilde{D}_{xy}$ est décrémentée par une constante δ ;

A ce modèle de base s'ajoute le calcul de la matrice de préférence. Nous proposons le modèle de choix discret développé par Antonini et al. (2004) pour l'élaboration de la matrice de préférence minimisant la déviation par rapport à la destination à atteindre. Les cellules se trouvant dans l'intersection du champ de vision du piéton et de sa destination sont au nombre de six, et seront favorisées en augmentant leurs préférences à chaque pas de temps (figure 2).

Le mouvement du piéton dépend de quatre facteurs :

1. A chaque piéton est attachée une matrice de préférence floue $\tilde{M}_{ij}$ qui reflète la probabilité de déplacement vers une cellule voisine. Il s'agit d'une simulation orientée « objectif » d'un piéton et de comportements personnalisés;
2. Le sol dynamique flou $\tilde{D}_{xy}$ influence le mouvement collectif des autres voisins, c'est-à-dire, quand le piéton se déplace d'une cellule $\tilde{D}_{ij}$ à une cellule $\tilde{D}_{xy}$ (x différent de i ou y différent de j) il influence la valeur de la cellule $\tilde{D}_{ij}$ en l'incrémentant d'un coefficient δ (variance dynamique). Ceci influencera la probabilité de déplacement des autres piétons vers la cellule (i, j) ;
3. Le sol statique flou $\tilde{S}_{ij}$ appelé aussi influence statique des objets fixes dans la grille ;
4. La possibilité de déplacement vers une cellule dépend de son état d'occupation ($n_{ij} = 1$ ou $n_{ij} = 0$).

L'utilité générale flou du déplacement vers une cellule (i,j) est dénotée par la formule ci-dessous :

$$\tilde{U}_{ij} = \tilde{M}_{ij} \times \tilde{D}_{ij} \times \tilde{S}_{ij} \times (1 - n_{ij})$$

4.3 Gestion des conflits

La collision entre les piétons ou la possibilité de choc signifie que deux piétons se déplaceront vers la même cellule à l'étape $i+1$. La solution proposée est basée sur l'utilité de déplacement vers la cellule partagée. En effet, selon la matrice de préférence de chaque piéton nous calculons l'utilité du déplacement vers la cellule partagée comme suit :

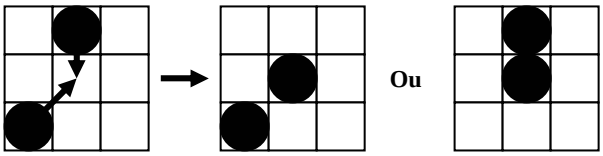


FIG. 3 – *Gestion de conflits entre piétons.*

Soient :

$\tilde{R}_1$ = la valeur de la matrice de préférence du piéton 1, pour la cellule partagée.

$\tilde{R}_2$ = la valeur de la matrice de préférence du piéton 2, pour la cellule partagée.

L'utilité relative de déplacement vers la cellule partagée de chaque piéton par rapport à l'autre (reflétant ainsi l'interaction entre les deux piétons) est donnée par :

$\tilde{F}_2^1 = \tilde{R}_1 / (\tilde{R}_1 + \tilde{R}_2)$

$\tilde{F}_1^2 = \tilde{R}_2 / (\tilde{R}_1 + \tilde{R}_2)$

Si $\tilde{F}_1^2 \succ \tilde{F}_2^1$, alors le piéton 2 est favorisé pour le déplacement, sinon c'est au piéton 1 de se déplacer.

4.4 Algorithme du modèle

- Initialisation des paramètres de simulation ;
- Pour chaque piéton, l'utilité de transition $\tilde{U}_{ij}$ pour un déplacement à une cellule voisine inoccupée (i,j) est déterminée par la matrice de préférence et les sols dynamiques et statiques locaux ;
- Chaque piéton choisit une cellule de la cible selon la matrice de transition $\tilde{P} = (\tilde{P}_{ij})$;
- Les conflits survenus entre piétons sont résolus par la procédure de gestion des conflits décrite au paragraphe précédent ;
- Les piétons autorisés à se déplacer exécutent leur pas de simulation ;
- Les piétons modifient le sol dynamique $\tilde{D}_{xy}$ de la cellule (x, y) qu'ils ont occupé avant leur déplacement.

5 Conception

5.1 Diagramme des cas d'utilisation

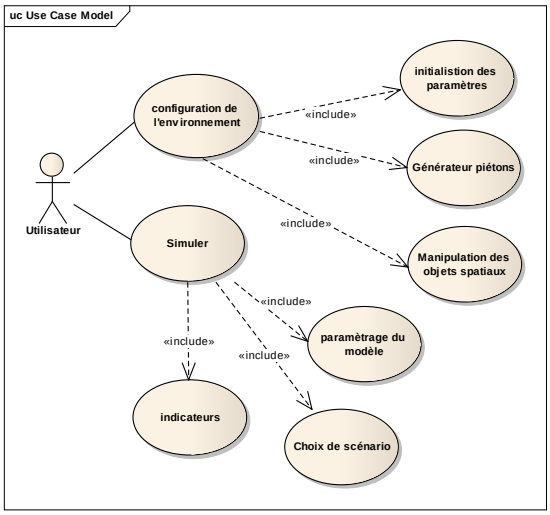


FIG. 4 – *Diagramme des cas d'utilisation.*

L'utilisateur commence par configurer l'environnement en initialisant les paramètres des champs de sols et la dimension de la carte. Puis il positionne les générateurs de piétons en définissant pour ces derniers le nombre de piétons et les matrices de préférences initiales. Cette phase comprend également la manipulation des objets spatiaux de l'environnement. En second lieu, l'utilisateur définit les paramètres du modèle. Il choisit un scénario puis lance la simulation. Cette dernière nous permet de récupérer les différents indicateurs physiques et statistiques.

5.2 Modèle de conception

Une interface homme machine est développée afin de faciliter l'utilisation du simulateur développé à base des classes MFC. Les résultats de simulation sont évalués pour les différents indicateurs physiques et statiques.

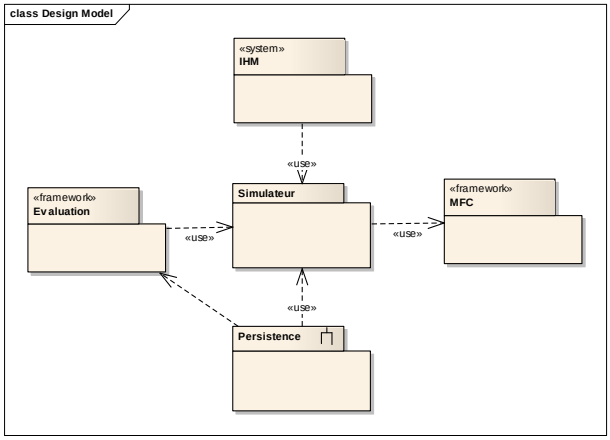


FIG. 5 – *Modèle de conception.*

5.3 Diagramme de classes

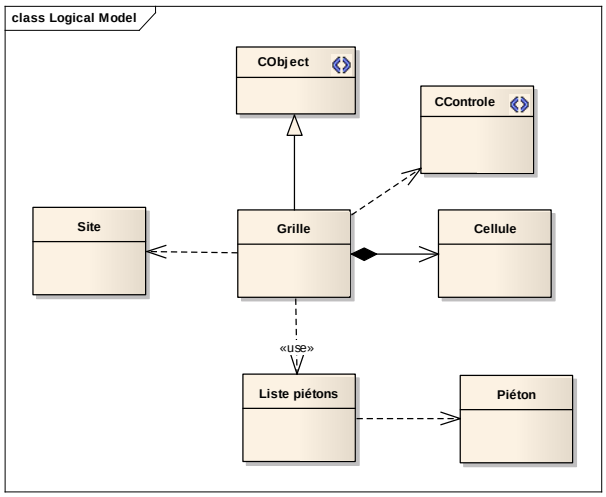


FIG. 6 – *Diagramme de classes.*

Les classes « Site », « CControle », sont associées aux boites de dialogues servant à définir respectivement le site, la dimension, les sols statiques et dynamiques. Puisqu'il s'agit des paramètres de la grille, la classe « Grille » en dépend. Elle est composée de « Cellule » et dépend aussi de la liste des piétons « Liste_P ». Nous calculons la probabilité de transition de chaque piéton dans la classe « Piéton ». La grille gère en particulier les conflits entre les piétons.

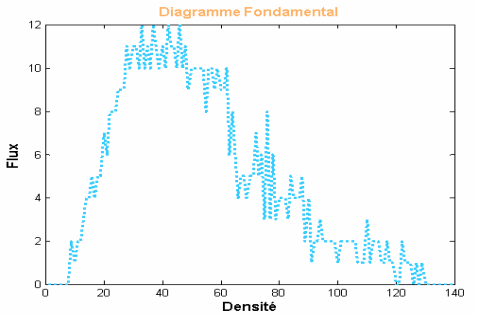


Fig.7 – *Diagramme fondamental représentant la densité des piétons en fonction de leurs flux en un site donné dans leur environnement*

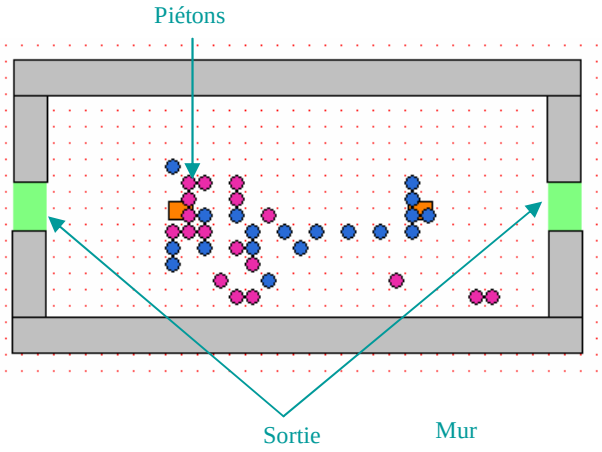


Fig. 8 – *Exemple de scénario d'évacuation*

6 Conclusion

Dans ce papier nous avons proposé un modèle simplifié de simulation des piétons virtuels. Ce modèle est fondé sur les comportements des sociétés organisées de type «fourmi», développé par Schadschneider.

A ce modèle nous avons intégré d'une part le modèle de choix discret pour le processus de calcul de la matrice de préférence. D'autre part nous avons considéré la fuzzification du modèle pour mieux modéliser la préférence et le trace chimique. Dans ce papier nous abordons les aspects liés à la conception et au déploiement du composant logiciel. Le prototype à l'état actuel donne des résultats probants en simulation. Son intégration dans un projet d'analyse de risque d'accidents en site urbain est considérée dans les projets de recherches que nous développons actuellement.

Références

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. New York: Chapman and Hall, 1984.

Antonini G., Venegas S., Thiran J.P. and Bierlaire M. *A Discrete Choice Pedestrian Behavior Model For Pedestrian Detection In Visual Tracking Systems.* Advanced Concepts for Intelligent Vision Systems, ACIVS 2004, Brussels, Belgium, 2004 IEEE. LTS-CONF-2004-018 ROSO-CONF-2004-004

Bak, P. *How Nature Works*: The Science Of Self Organized Criticality, Springer-Verlag New York, Inc. 1996.

Blue V.J. & Alderb J.L. *Cellular Automata Model of Emergent Collective Bi-Directional Pedestrian Dynamics*. Artificial Life VII, the Seventh International Conference on the Simulation and Synthesis of Living Systems. 2000.

Cavens D., Gloor C., Illenberger J., Lange E., Nagel K., Schmid W. A. *Distributed intelligence in pedestrian simulations.* pp. 201-212, Pedestrian and Evacuation Dynamics 2005, Springer Ed.

Constantine K. C. *A Simple Pedestrian Simulator Using Node-Edge Graphs for Floorplan Models.* Cambridge, Proceedings of the 2006 Summer Simulation Multiconference, July 30 - August 3, 2006, Calgary, Canada

Delgado M., J.L. Verdegay & M.A. Vila. *Fuzzy numbers, definitions and properties. Math and Soft Computing.* Vol. 1 (1994), pp. 31-43. España,

Dijkstra J., Timmermans H.J.P. and Jessurun A.J., A Multi-Agent Cellular Automata System for Visualising Simulated Pedestrian Activity, In *S. Bandini, and T. Worsch(ed.): Theoretical and Practical Issues on Cellular Automata, Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry.* Springer-Verlag, Berlin 2000. pp. 29-36.

Dubois, D. H. Prade, *Ranking of fuzzy numbers in the setting of possibility theory*. Inform. Sci., 30:183-224, 1983.

The Executive Director of the Agency. ED Decision 2003/16/RM on Certification Specifications for Large Rotorcraft. European Aviation Safety Agency, November 2003. http://www.easa.eu.int/doc/Agency, mesures/Certification Spec/decision ED 2003 16 RM.pdf

Helbing, D., Farkas, I. J., Molnar, P. and Vicsek, T. *Simulation of pedestrian crowds in normal and evacuation simulations.* In Pedestrian and evacuation dynamics, edited by M. Schreckenberg and S. Deo Sarma, 21-58. 2002, Berlin: Springer-Verlag.

Helbing D., Farkas I. J., and Vicsek T., "*Simulating dynamical features of escape panic*," *Nature*, vol. 407, pp. 487–490, 2000.

Helbing, D., Farkas, I. J., Molnar, P. and Vicsek, T. *Simulation of pedestrian crowds in normal and evacuation simulations.* In Pedestrian and evacuation dynamics, edited by M. Schreckenberg and S. Deo Sarma, 21-58. 2002, Berlin: Springer-Verlag.

Hoogendoorn, S.P. , P.H.L Bovy, and W. Daamen. *Microscopic pedestrian wayfinding and dynamics modelling*. In M. Schreckenberg and S.D. Sharma, editors, *Pedestrian and Evacuation dynamics*, pages 123–154. Springer, Berlin, 2002.

Kirchner, H. Klupfel, K. Nishinari, A. Schadschneider, and M. Schreckenberg *Simulation of competitive egress behavior: comparison with aircraft evacuation data*. Physica A, 324:689–697, Jun 2003. DOI:10.1016/S0378-4371(03)00076-1.

Kirchner A. and Schadschneider A. *Cellular automaton simulation of pedestrian dynamics and evacuation processes, in ``Traffic and Granular Flow* '01'. October 15-17, 2001, Symposion, Nagoya University, Japan, Springer 2002, ISBN 3-540-40255-1

Lacroix B., Mathieu P., Picault S. *Une gestion réaliste du temps et de l'espace dans les simulations de foules*. in : V. Chevrier (ed.), Systèmes multi-agents : articulation entre l'individuel et le collectif. Actes des 14e Journées Francophones sur les Systèmes Multi-Agents, JFSMA 2006.

Mauron, L. *Pedestrians simulation methods*, Diploma thesis, Swiss Federal Institue of Technology ETHZ, 2002.

Nagel K. and Schreckenberg M., J.Phys. A: Math. Gen. 35 (2002) L573–L577.

Schadschneider A., *Cellular Automaton Approach to Pedestrian Dynamics* – Theory, "Pedestrian and Evacuation Dynamics", M. Schreckenberg and S.D. Sharma (Eds.), pp. 87 Springer 2001.

Srichander Ramaswamy, *Portfolio Selection Using Fuzzy Decision Theory*. BIS Working Paper No. 59 November 1998 Available at SSRN: http://ssrn.com/abstract=856064

Yingchao Shao Zheng Pei, A. *Method for Ranking Fuzzy Numbers and Its Application to Game with Fuzzy Profit*. Publication: ISKE-2007 Proceedings, ISBN: 978-90-78677-04-8, ISSN: 1951-6851

Zadeh, L.A. *Fuzzy Sets, Information and Control*, 8 (1965) 338-353.

Zhang J., Wang H., Li P.. Cellular automata modelling of pedestrians crossing dynamics. Journal of Zhejiang University Sciences - ISSN 1009-3095, Vol.5, num 7, 835-840, 2004.

Summary

In this paper, cellular automata simulation techniques are used to model the virtual pedestrian dynamic, using the model proposed by Schadschneider. We integrate to this basic model on one hand a fuzzy preference computing process based on discrete choice model to minimize the reachability cost to the target. On the other hand we use the fuzzy concept to model the preference and chemical trace.

Un système de détection de fraude en téléphonie mobile à base d'un système d'inférence floue

Rachid Elmeziane*, Ilham Berrada* et Ismail Kassou*

*ENSIAS, Université Mohammed V Souissi, Laboratoire ALKHAWARIZMI,
équipe BIRONI, BP 713, Rabat Maroc
{meziane, iberrada, kassou}@ensias.ma

Résumé. La détection de fraude en téléphonie mobile est l'activité qui consiste à identifier les utilisations non autorisées du réseau mobile et de prévenir les pertes de ces utilisations pour les opérateurs de Télécommunication. Nous proposons dans ce travail un modèle de détection de fraude, en téléphonie mobile, en utilisant l'approche d'inférence floue basée sur la méthode ANFIS (Adaptative Network based Fuzzy Inference System), avec une fonction d'appartenance de niveau de consommation. Un tel système sera comparé sur des données simulées avec un système d'inférence Neuro Flou basé sur la fonction d'appartenance gaussienne. Deux critères d'évaluation ont été retenus lors de la comparaison. Le premier est un critère de performance traduit en terme de minimisation de fausses alertes par l'analyse ROC (Receiver Operating Characteristics). Le second est relié à la facilité d'interprétation par un décideur final des règles générées par les deux systèmes étudiés.

1 Introduction

Les systèmes de détection de fraude sont largement utilisés dans des secteurs aussi variés que l'assurance, la sécurité informatique, les télécommunications, etc. Des systèmes de détection de fraude en téléphonie mobile ont prouvé leur efficacité en utilisant à la fois des techniques à base de règles (i.e. des approches supervisées) et des techniques non supervisées telles que la classification Shawe-Taylor et al. (1999).

L'élaboration d'un système de détection de fraude en téléphonie mobile dépend de la capacité du système à s'adapter au changement de comportements des clients fraudeurs et légitimes et la capacité du décideur final à interpréter facilement et de manière précise un tel changement. Des approches basées sur les réseaux de neurones et d'autres approches basées sur la génération de règles sont largement utilisées dans la détection de fraude en téléphonie mobile. Il existe donc un grand intérêt pour les systèmes capables de détecter la fraude en se basant sur une analyse des changements de comportement des abonnés Barson et al. (1996), Burge et al. (1997), Fawcett et Provost (1997) et Taniguchi et al. (1998). Des approches d'analyse absolue et d'analyse différentielle ont été suggérés pour la détection de fraude à partir de l'activité d'appel Moreau et al. (1997). Dans l'analyse absolue, la détection de fraude est basée sur la comparaison des modèles de l'activité d'appel des comportements anormaux et

légitimes. L'analyse différentielle approche le problème de la détection de fraude en détectant des changements soudains dans les comportements des activités d'appel d'un abonné.

La motivation principale de cette recherche consiste à développer un modèle de détection des fraudes en téléphonie mobile, en utilisant une approche basée sur un système d'inférence Neuro Flou, tout particulièrement ANFIS (Adaptative Network based Fuzzy Inference System), avec une fonction d'appartenance du niveau de consommation développée dans Elmeziane et al. (2007). Ce système est comparé avec un système d'inférence Neuro Flou basé sur la fonction d'appartenance gaussienne développée dans Karahoca et al. (2007). Ces systèmes sont testés sur des données simulées à partir du modèle de données d'un opérateur de téléphonie mobile pour analyser l'activité de comportements frauduleux d'abonnement.

Ce papier comporte trois sections. Après une introduction dans la première section, la deuxième section introduit le concept de l'apprentissage adaptatif et décrit le système d'inférence flou ANFIS. Dans la première sous-section de la deuxième section, on discute les entrées du système ANFIS et dans la deuxième sous-section, on décrit la fonction d'appartenance de niveau de consommation en téléphonie mobile développée dans Elmeziane et al. (2007) en temps que choix de la fonction d'appartenanace pour notre système, puis dans la dernière sous section, on discute l'algorithme d'apprentissage pour le système ANFIS. La dernière section présente les résultats obtenus en utilisant dans le système ANFIS la fonction d'appartenance gaussienne d'une part et la fonction d'appartenance basée sur le niveau de consommation d'autre part et notamment la performance de tels systèmes.

2 Description du Système d'inférence floue : ANFIS

Le système ANFIS est une classe de réseau adaptatif introduit par Jang (1993, 1992). Il s'agit d'un réseau non bouclé pour lequel chaque couche est un composant d'un système neuro-flou. Comme tout Réseau de Neurone (RN), il connaît un succès fort par ses caractéristiques à modéliser et à reproduire des phénomènes sans connaissances à priori pour détecter des relations cachées entre entrées et sorties. Son adaptation à un problème donné demeure une tâche complexe puisqu'elle nécessite l'ajustement de plusieurs paramètres de performance : nombre d'entrées et de sorties, nombres de couches, fonctions d'activation, choix de la base de test, algorithmes d'apprentissage, etc. Il présente néanmoins l'avantage par rapport aux RN classiques de ne pas être utilisé comme une boite noire puisqu'il offre la possibilité d'expliciter et d'analyser les relations entre entrées et sorties.

Afin de décrire l'architecture et le fonctionnement d'un système ANFIS, considérons la figure 1 où l'on suppose que le système flou possède quatre entrées x_1, x_2, x_3, x_4 et une sortie $\hat{y}$. Le système ANFIS est composé de 5 couches exception faite de la couche d'entrée. Si deux ensembles flous sont associés à chaque variable d'entrée comme dans la figure 1, alors le système présente 16 règles d'inférence $R_j(2^4)$. Les inférences sont de type Sugeno-Takagi de premier ordre, R_j s'écrit :

$$\begin{aligned} &\text{si } (x_1 \text{ est } A_1^j) \text{ et } (x_2 \text{ est } A_2^j) \text{ et } (x_3 \text{ est } A_3^j) \text{ et } (x_4 \text{ est } A_4^j) \\ &\qquad \text{alors } \hat{y}_i = c_1^j x_1 + c_2^j x_2 + c_3^j x_3 + c_4^j x_4 + c_5 \end{aligned} \tag{1}$$

où $j = 1, .., 16$ (2^4). Notons par $c_{i,k}$ la sortie du neurone k de la couche i.

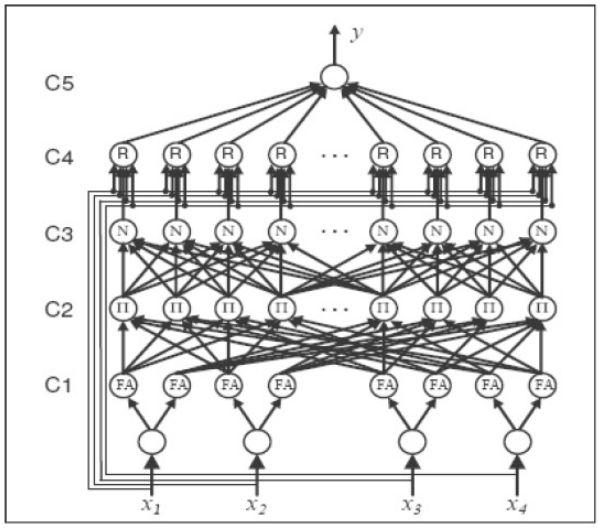


FIG. 1 – *Système ANFIS à 4 entrées et 2 fonctions d'appartenance par entrée.*

Dans la couche 1, chaque noeud se comporte comme une fonction d'appartenance (gaussienne, sigmoïde, etc.). Cette couche permet la " fuzzification " des variables x_1, x_2, x_3, x_4. A titre d'exemple, on considère la fonction d'appartenance gaussienne définie par :

$$\mu_{A_l^j}(x) = exp(-[(x - m_l^j)/b_l^j]^2) \tag{2}$$

avec $l \in 1, 2, 3, 4$ le numéro de l'entrée et m_l^j, b_l^j l'ensemble des paramètres d'une gaussienne.

Dans la couche 2, chaque noeud correspond à une T-Norme floue. L'opérateur produit est généralement utilisé.

$$\mu_j = \prod_{l \in 1,2,3,4} \mu_{A_l^j}(x_l)$$

Dans la couche 3, les sorties des T-Norme sont normalisées

$$\mu_j = \frac{\mu_j}{\mu_1 + \mu_2 + \mu_3 + \mu_4}$$

Dans la couche 4 chaque noeud est relié aux entrées initiales par une combinaison linéaire donnée selon Takagi et al. (1985) par :

$$c_1^j x_1 + c_2^j x_2 + c_3^j x_3 + c_4^j x_4 + c_5$$

Dans la couche 5 la sortie prédite obtenue est donnée par :

$$\hat{y} = \frac{\sum_{j=1}^{16} \mu_j (c_1^j x_1 + c_2^j x_2 + c_3^j x_3 + c_4^j x_4 + c_5)}{\sum_{j=1}^{16} \mu_j} \tag{3}$$

Le système neuro-flou présenté dans la figure 1 comporte 96 paramètres devant être optimisés (16 inhérents à la fonction gaussienne et 80 à la linéarisation des sorties des règles) lors de la phase d'apprentissage.

Il résulte de la présentation ci-dessus que l'objectif du système ANFIS est de générer un système à base de règles. La sortie de chaque règle est une combinaison linéaire de variables d'entrées et d'un terme constant. Le résultat final est la moyenne pondérée de chacune des règles de production. La règle d'apprentissage du réseau se base sur la méthode de descente du gradient Werbos (1974). Dans la conception du système ANFIS, le nombre de fonctions d'appartenance et le nombre de règles floues doivent être choisis de façon appropriés afin que le système d'ajustement sur les données soit en mesure d'adapter les données.

2.1 Sélection des entrées de ANFIS

On considère dans ce travail un ensemble de données simulées de télécommunication relatives aux trafics de certains abonnés dont la proportion des opérations de fraude représente 1% de l'échantillon simulé. Ces données comportent 10 039 instances qui représentent les opérations quotidiennes des abonnés d'un opérateur de télécommunication.

La transformation des données conduit au calcul d'agrégats caractérisant le trafic quotidien de chaque abonné tels que :

- le nombre total des appels par jour ;
- les durées totales d'appel par jour ;
- la durée maximale d'appel par jour et
- l'écart type des durées d'appel par jour.

Les mêmes agrégats ont été calculés en terme des montants des appels par jour pour chaque abonné.

2.2 Choix de la fonction d'appartenance

Dans Elmeziane et al. (2007) nous avons présenté sur un cas réel l'utilisation de la théorie des ensembles flous pour la définition d'une fonction d'appartenance permettant d'évaluer, de manière précise, le niveau de consommation, des abonnés en téléphonie mobile. Le niveau de consommation d'un abonné est souvent calculé à partir de la durée facturée qui s'avère insuffisante dans la plupart des cas. En effet, deux abonnés peuvent avoir la même durée d'appel pour des services différents mais sans avoir le même degré de consommation. D'où la nécessité d'introduire d'autres critères dans la détermination du niveau de consommation.

L'approche de résolution proposée dans Elmeziane et al. (2007) comporte trois étapes principales. Dans une première étape l'Analyse des Correspondances Multiple (ACM) a été appliquée sur les modalités des variables caractérisant les produits et services. L'objectif de l'étape 2 est la segmentation des abonnés par produits et services afin de discriminer entre les abonnés en se basant sur le comportement d'utilisation des produits et services. Les facteurs obtenus par l'ACM ont été utilisés comme variables d'entrée du Réseau de Kohonen. Les inerties intra classes de chaque facteur ont ensuite été utilisées dans l'étape 3 comme indicateur de la variabilité du niveau de consommation au sein de chaque classe.

La fonction d'appartenance proposée pour quantifier l'ensemble flou A= «niveau de consommation en téléphonie mobile» présente une mise à l'échelle exponentielle des agrégats caractérisant le trafic et le montant de l'appel. Cette fonction d'appartenance donnée par l'équation (4),

présente non seulement l'avantage de ne pas masquer les optima locaux de la distribution des agrégats mais aussi l'avantage d'être paramétrée par des critères liés aux produits et services.

$$f_A(d) = \sum_{j \in K} I_j(d) \sum_{i \in F_j} d^{px_{ij}} \tag{4}$$

où d : un agrégat d'appel,
$K = \{C_1, C_2, C_3, C_4\}$: l'ensemble des classes issues du réseau de Kohonen,
F_j : l'ensemble des facteurs contribuant à la construction de la classe j,
x_{ij} : l'inertie du facteur i appartenant à F_j.
I_j : est la fonction indicatrice de la classe j,
p : une constante à ajuster pour atténuer les variations de d, elle a été fixée à 0.01 suite aux différents tests numériques réalisés.

2.3 Choix de l'algorithme d'apprentissage dans ANFIS

Dans le but de réduire la complexité d'apprentissage et d'en améliorer l'efficacité, un algorithme hybride proposé par Jang (1993, 1992) combinant la méthode des moindres carrés et la méthode de descente du gradient est généralement utilisé.

On considère qu'on dispose de K entrées qui définissent les 10 039 instances. Pour une entrée donnée $x_1^k, x_2^k, x_3^k, x_4^k$ la sortie du modèle est $\hat{y}$ donnée par l'équation 3. La différence entre cette sortie et celle du système pour la même entrée est donnée par :

$$e_k = \hat{y}_k - y_k$$

On définit l'erreur quadratique en ce point k par :

$$E_k = \frac{1}{2} e_k^2 = \frac{1}{2} (\hat{y}_k - y_k)^2 \tag{5}$$

C'est une erreur instantanée sur le point k utilisée comme premier critère d'optimisation. Ce critère est généralement utilisé pour un apprentissage en ligne. L'erreur quadratique moyenne est définie par :

$$E = \frac{1}{K} \sum_{k=1}^{K} E_k \tag{6}$$

C'est une fonction objectif qui constitue un deuxième critère d'optimisation. Ce critère est utilisé pour un apprentissage hors ligne en utilisant tout l'ensemble de données d'apprentissage.

Les paramètres à identifier sont de deux types : les paramètres d'entrée et les paramètres de sortie. Les paramètres d'entrée sont ceux des fonctions d'appartenance. On suppose que chacune de ces fonctions d'appartenance peut être décrite par p paramètres où p dépend de la forme de fonction choisie. Dans le cas de la fonction gaussienne donnée par l'équation 2, deux paramètres (moyenne et écart type) suffisent, soit p=2. Dans le cas de la fonction d'appartenance de niveau de consommation, p=41 (les inerties des facteurs des classes et le paramètre d'ajustement)(voir équation 4). Les paramètres de sortie sont ceux des fonctions apparaissant dans les conclusions de règles. Si l'on considère le système ANFIS de la figure 1,

où $A_1, A_2, ..., A_4$ sont des ensembles flous respectivement pour chaque variable $x_1, x_2, ..., x_4$, le système est alors entièrement paramétré par le vecteur :

$$P = (a_{11}...a_{1A_4}\ a_{21}...a_{2A_4}...a_{41}...a_{4A_4}\ c_1...c_R)^T$$

avec a_{ij} vecteur des paramètres relatif au j^{ime} ensemble flou défini pour la i^{ime} variable, $i = 1...4$, $j = 1...4$ et c_i vecteur de 5 coefficients relatifs à la i^{ime} règle, $i = 1...m$.

L'algorithme comporte deux phases. Dans la phase avant, les paramètres a_{ij} sont fixes, et, par conséquent, les paramètres c_i sont obtenus en utilisant l'estimation moindre carrée linéaire en minimisant le critère défini par l'équation 6. Durant la phase arrière, les paramètres c_i sont fixés et l'erreur de la sortie est propagée en arrière par le biais de ce réseau, ainsi, les paramètres a_{ij} sont mis à jour récursivement, en minimisant le critère défini par l'équation 5, en utilisant la méthode de descente du gradient.

L'objectif du processus d'apprentissage est de minimiser l'erreur quadratique moyenne entre la sortie prédite par ANFIS et la variable cible qui est dans notre cas la variable indicateur de fraude. Cela permet à un système flou l'apprentissage à partir des données qu'il observe et met en oeuvre ces fonctions dans le système de règles. La figure 2 montre l'architecture ANFIS d'Inference Neuro Flou à gauche ANFIS pour la fonction d'appartenance gaussienne et à droite ANFIS pour la fonction d'appartenance de niveau de consommation. Pour ANFIS à base de la fonction d'appartenance de niveau de consommation permettant d'obtenir uniquement deux règles pour le système d'inférence flou tandis que ANFIS à base de la fonction d'appartenance gaussienne permet de générer quatre règles.

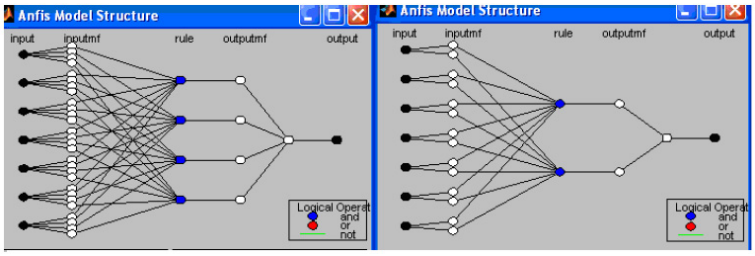


FIG. 2 – *L'architecture ANFIS d'Inference Neuro Flou, à gauche ANFIS pour la fonction d'appartenance gaussienne et à droite ANFIS pour la fonction d'appartenance de niveau de consommation.*

3 Résultats et discussions

Les divers outils de détection de la fraude proposés dans la littérature utilisent des approches d'analyse sur des données des appels des abonnés permettant de détecter les irrégularités dans le trafic. Le comportement réel des appels d'un abonné donné est comparé avec son comportement prédit par un modèle d'apprentissage sur son historique des appels. Ainsi, aussitôt qu'un changement de comportement est détecté, un système de détection de fraude déclenche immédiatement une alerte. La précision d'un tel système est souvent évaluée par des

analyses de type ROC (Receiver Operating Characteristics) permettant de comparer les faux positifs des faux négatifs.

Dans ce travail, nous retenons l'analyse ROC pour évaluer et comparer la performance de la méthode ANFIS utilisant dans la couche 1 aussi bien la fonction d'appartenance gaussienne que celle relative au niveau de consommation. Aussi, on applique sur les mêmes données deux méthodes d'apprentissage supervisé telles que un arbre de décision de type C5.0 et un réseau de neurone de type perceptron multicouche constitué de 8 unités d'entrés représentant les agrégats caractérisant le trafic quotidien de chaque abonné, deux unités cachées avec la fonction sigmoïde pour la première unité qui est constituée de cinq neurones et la fonction linéaire pour la deuxième unité qui constituée de deux neurones. La courbe ROC analyse l'information sur la performance de toutes les combinaisons possibles et les coûts de mauvaise classification, ainsi que la distribution des classes. Les matrices de confusion qui résume le nombre de cas prédits correctement ou incorrectement par un modèle de classification sont indiquées dans la figure 3.

		Classe prédite pour ANFIS niveau de consommation		Classe prédite pour ANFIS gaussienne		Classe prédite pour Réseaux de Neurones		Classe prédite pour Arbre de Décision	
		0	1	0	1	0	1	0	1
Indicateur de fraude observé	0	100	0	100	0	100	0	100	0
	1	12.871	87.129	3.960	96.040	0.990	99.010	0.990	99.010

FIG. 3 – *Matrices de confusion pour ANFIS niveau de consommation et pour ANFIS gaussienne.*

La courbe ROC sélectionnée pour les quatre méthodes est illustrée sur la figure 4. Il en résulte que la méthode ANFIS pour la fonction d'appartenance gaussienne donne un résultat meilleur que la méthode ANFIS pour la fonction d'appartenance de niveau de consommation. Néanmoins, en terme d'interprétation et comme ceci est illustré par la figure 2, l'utilisation de la fonction d'appartenance à base de niveau de consommation permet de générer moins de règles floues que la fonction d'appartenance gaussienne ce qui permettra à l'analyste humain un jugement plus réaliste.

L'utilisation des méthodes d'apprentissage supervisé telles que les réseaux de neurones et les arbres de décision appuie les résultats obtenus par ANFIS. Comme illustré sur la figure 4, le réseau de neurones et l'arbre de décision donnent de meilleurs résultats que ANFIS en terme de faux positives et faux négatives. Par contre, en terme d'interprétation ANFIS est meilleur car il offre la possibilité d'interpréter les classes. Ainsi par exemple, un abonné qui a une consommation excessive n'est pas forcement un abonné frauduleux s'il utilise des produits et services de téléphonie internationaux.

Malgré le fait que le niveau de précision fourni par les systèmes automatisés de détection des fraudes soit assez bon, les opérateurs doivent encore choisir entre une excellente prévention de la fraude et le nombre minimum de fausses alertes de fraude. La méthode ANFIS donne des résultats précis surtout quand le taux des faux positifs est un sujet de préoccupation surtout si on arrive à interpréter ces derniers. Cette propriété est importante dans les réseaux de télé-

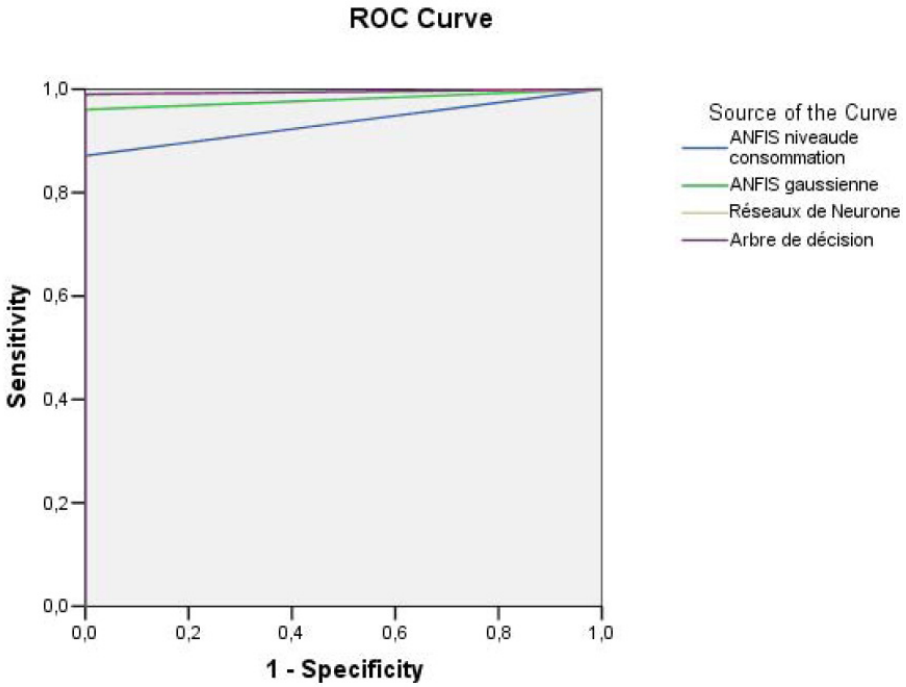


FIG. 4 – *Les courbe ROC pour ANFIS avec fonction d'appartenance de niveau de consommation et ANFIS avec une fonction d'appartenance gaussienne ainsi que pour le réseaux de neurones et l'arbre de décision*

communications car les fausses alertes de fraude peuvent entraîner des pertes importantes de revenus.

4 Conclusion

Dans le but de fournir un système capable de s'adapter au changement de mauvaise classification de cas anormaux, nous proposons d'utiliser une approche qui combine entre les réseaux de neurones qui ont prouvé leur efficacité d'adaptation et les systèmes d'inférence floue qui présentent l'avantage d'interpréter le changement soudain dans le comportement des abonnés. Les résultats montrent que, la méthode ANFIS combine à la fois la précision du système de classification fondée sur l'approche floue et l'adaptabilité (propagation en arrière) caractéristique des réseaux de neurones dans la classification des données. L'utilisation d'une fonction d'appartenance à base de niveau de consommation donne la possibilité de bien interpréter les changements dans le comportement des abonnés en se basant sur les produits et services utilisés.

Bien qu'un inconvénient majeur de la méthode ANFIS soit sa complexité croissante en fonction du nombre des enregistrements introduits dans le système, elle peut être utilisée effi-

cacement pour un ensemble important de données lorsque le système atteint une configuration optimale de la fonction d'appartenance. Le présent travail de recherche montre que les modèles ANFIS peuvent être utilisés comme une alternative aux techniques de détection actuellement utilisées pour la détection des fraudes en général et en particulier dans le domaine de la téléphonie mobile.

Références

Barson, P., S. Field, N. Davey, G. McAskie, et R. Frank (1996). The detection of fraud in mobile phone networks. *Neural Network World 6(4)*, 477–484.

Burge, P., J. Shawe-Taylor, Y. Moreau, H. Verrelst, C. Störmann, et P.Gosset (1997). Brutus - a hybrid detection tool. In *Proc. of ACTS Mobile Telecommunications Summit*.

Elmeziane, R., I. Berrada, I. Kassou, et K. Baina (2007). Détermination du niveau de consommation des abonnés en téléphonie mobile par la théorie des ensembles flous. In *RNTI E9 volume I Proceeding de7th journée francophone en extraction et gestion de connaissance*, pp. 215–216.

Fawcett, T. et F. Provost (1997). Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery Vol. 1, No. 3*, 1–28.

Jang, J.-S. (1992). Self-learning fuzzy controllers based on temporal back propagation. *IEEE Trans. Neural Networks 3(5)*, 714–723.

Jang, J.-S. (1993). Anfis : adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybernet 23(3)*, 665–685.

Karahoca, A., S. Yalcin, C. Kahraman, et M. Sanver (2007). Fraud detection using an adaptive neuro-fuzzy inference system in mobile telecommunication network. *World Scientific Publishing Co. Pte. Ltd. INFORMATION SCIENCES*, 1440–1446.

Moreau, Y., H. Verrelst, et J. Vandewalle (1997). Detection of mobile phone fraud using supervised neural networks : A first prototype. In *In International Conference on Artificial Neural Networks Proceedings (ICANN97)*, pp. 1065–1070.

Shawe-Taylor, J., K. Howkev, et P. Burge (1999). Detection of fraud in mobile telecommunications. Information Security Technical Report Vol. 4, No. 1 16-28, Royal Holloway, University of London.

Takagi, T., M. Sugeno, et C. Kang (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transaction on Systems, Man, and Cybernetics 15*, 116–132.

Taniguchi, M., M. Haft, J. Hollmén, et V. Tresp (1998). Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE Int. Conf. in Acoustics, Speech and Signal Processing (ICASSP98)*, Volume 2, pp. 1241–1244.

Werbos, P. (1974). *Beyond regression, New tools for prediction and analysis in the behavioural sciences*. Phd thesis,, Harvard University.

Summary

Fraude detection in mobile telephony is the activity to identify the unauthorized use of the mobile network and prevent losses to telecommunications operators. We propose in this work a model for detecting fraude, in mobile phone, using approach based on the method ANFIS (Adaptive Network based Fuzzy Inference System) with a membership function of level of consumption. Such a system will be compared with simulated data with a Neuro Fuzzy inference system based on the Gaussian membership function. Two evaluation criteria were identified at the comparison. The first is a performance criterion translated in terms of minimizing false alarms by ROC analyzing (Receiver Operating Characteristics). The second is connected to the ease of interpretation of rules generated by the two systems studied for final decision-maker.

"Individual Factors and E-learning Effectiveness" The Tunis Virtual University Case

Wafa Kort *, Jamel-Eddine Gharbi **

* Ligue laboratory
Ecole Supérieure de Commerce de Manouba
Résidence Essourour, app 6, Bloc F, Cité Olympique, Tunis
wafakort@yahoo.fr
** Ligue laboratory
Faculté des Sciences Juridiques, Economiques et de Gestion de Jendouba
18, rue 2 mars 8001 Jendouba
jgharbi@yahoo.com

Abstract. The Tunis virtual university was created in 2002. Its aim was spread the education on the overall universities. Until its creation five years ago, no studies were interested about its effectiveness or the effectiveness determinants. Therefore, this study is about the learning effectiveness. It tries to determining the factors influencing the e-learning effectiveness. We are adopting a causal model that make in relation involvement, need for cognition, flow state and e-learner effectiveness.
A questionnaire is addressed to a representative sample of 443 students. Our study shows that individual factors have an influence on learning effectiveness.

Key words: Involvement, Need for cognition, Flow state and E-learning effectiveness.

1 Introduction

The new economic era has changing paradigms and norms. The management of knowledge became crucial. Maurer and Saper (2001) state that over time knowledge became a more widely available commodity: by archiving it in books, by introducing schools and universities, by communication between communities of scholars. Moreover, we're talking about the information society. Nowadays, many institutions as universities, organizations and companies are looking for developing flexible mechanism to deliver knowledge. The knowledge should go beyond just delivered to the learner. The new way to deliver knowledge should delete geographic, linguistic and social barrier.

Tunis is an under developed country which want to have a place in this new system. Therefore, it wants to increase the graduates' number to attain 97940 on 2030. The actual number of student is around 346000. This number will be 492000 on 2012 but will decrease to attain 331820 on 2030. So that, the e-learning was considered as a good solution on the one hand to

manage the fluctuation of the student's number and on the other hand to use the information technology to facilitate the move of information and knowledge.

In this way, the virtual university was created to generalize this new mode in the universities. Therefore, we're trying to determine some factors influencing the effectiveness of the e-learning. We're interested of the individual factors because these factors are essential for e-learning adoption. As such our question research is:

To what extent does involvement and need for cognition influence the flow state and to what extent does the flow state influence e- learning effectiveness?

This paper will be organized as follow: the next section provides the literature review about the constructs of the study. The following section describes the methodology and the variable measures. The final section discusses the validity of the assessment and the study analysis

2 Literature review

2.1 e-Learning effectiveness

To explain learning effectiveness three theories had emerged: the behaviourist theory, the constructivist theory and the cognitive process theory. Each theory focuses on some learning characteristics.

Thorndike (1911), Watson (1978) and Skinner (1950) had founded the behaviourist. This approach explains the learning as a behavioural change which results from an answer to some external stimuli. It states that the learning mechanism has three components: the environment that stimulates someone, the stimulated organism and the behaviour or the answer following the stimulation (Wikipédia, 2006). The most important thing according to this theory is the individual behavioural change (Zairi, 2004). Therefore, learning is successful whenever it produces a behavioural change.

The constructivist theory views learning as an experience. Elworthy (2004) states that the constructivist theory is based on experience. One learns by building their own knowledge from their own experience. She adds that our experiences are unique and so is pour way of learning things. As such, according to same information the interpretations change. Zhang and al (2005) argue that the constructivism theory views the learning as a synthesis of some concepts on the learner mind which represents reality.

The cognitive process theory states that the learner attention is selective. Every one tries to construct their own model to processing information (Zairi and jallouli, 2004). This theory is an extension from the constructivism theory (Zhang et al, 2005). The most important thing according to this theory is the individual cognitive outline. Consequently, to be successful the learning shape should match this cognitive outline. Nevertheless, it's difficult to customize the learning. Therefore, e-learning offers the opportunity to learners to choose their own learning shape. In fact, they can choose the adequate rate, the adequate time or the session order.

Regarding the e-learning effectiveness dimension, the literature doesn't offer consensus. Several researches state that the learning dimension should be the same. So that, many researches made comparison between e-learning effectiveness and traditional effectiveness. Even so, the famous model cited in the literature is the Kirkpatrick model (1959) and the Bersin model (2002).

The Kirkpatrick model states that learning has four dimensions: reaction, learning, behaviour and results (Kirkpatrick, 1996). On the other hand the Bersin (2002) model states that

the e-learning effectiveness has five dimensions enrolment, activity, achievement, score and feedback.

The difference between the two models is that the Bersin model is specific to the e-learning but the Kirkpatrick model is general. In addition to that, the Bersin model focuses more on the technical factors but the Kirkpatrick model is more focused on the individual traits. However, the two models view that effectiveness is related to the knowledge change. Thus, we assume that e-learning is effective when it produce good results.

2. 2 Flow state

Csikszentmihalyi (1977) describe the flow state as: "the process of optimal experience". The process of optimal experience is attained when the user is enough motivated ant so perceives a balance between his skills and his challenges even the interaction with a medium.

Hoffman and Novak (1996) state: when in the flow state, irrelevant thoughts and perceptions are screened out and the consumer's attention is focused entirely on the interaction. So that, flow involves a merging of actions and awareness, with concentration so intense there is little attention left over to consider anything else.

Additionally, Csikszentmihalyi (1990) argues that subjects get into an experience that is entirely absorbed by the activity. He adds « …this mode is characterized by a narrowing of the focus of awareness, so that the irrelevant perceptions and thought are filtered out; by loss of self-consciousness, by a responsiveness to clear goals and unambiguous feedback; and by a sense of control over the environment …».

Recently, Hoffman and Novak (1996) extended and developed the flow construct in the context of computer-mediated environments like the Web. They identified four properties which define flow during network navigation. Flow is characterized by a seamless sequence of responses facilitated by machine-interactivity intrinsically enjoyable accompanied by a loss of self-consciousness; and self-reinforcing. They argue that the perception of skill and challenge is the first condition to get the flow state.

Originally the model of the skill and challenge was developed by Csikszentmihaly (1977). It argues that the flow state results when the person perceives a match between skill and challenge. The level of skill and challenge may be high or low. If the level of challenge exceeds that of skill, it produces an anxiety state. If the level of skill exceeds that of challenge, it produces a boredom state. Later, different models were developed but are also based on the Csikszentmihaly model (1977). These models add intermediate level to the original one. So that, Massimi and Carli (1988) and Csikszentmihalyi and Nakamura (1988) proposed the eight canal model. After that, Massimi and Carli (1988) proposed the sixty canal model. Finally, Clarke and Haworth (1994) proposed the nine canal model.

Regarding the flow state dimensions, several researches were doing. Some researches identify too dimensions: concentration and enjoyment (Ghani and Deshpande, 1994). Others researches identify four dimensions: control, intrinsic interest, attention focus and Curiosity (Trevino et al, 1992; Webster et al., 1993) add too other dimensions: control and intrinsic interest. Even though, flow state is a multidimensional construct.

Nevertheless, regarding web site context, the Csikszentmihaly model (1996) is very famous (Ettis, 2005; Chen et al, 1999; Chen 2006; Pilke, 2004). This model argue that flow state has nine dimensions that are: (1) clear goals, (2) immediate feedback, (3) personal skills well suited to given challenges, (4) merging of action and awareness, (5) concentration on the task at hand, (6) a sense of potential control, (7) a loss of self-consciousness, (8) an altered sense of time, and (9) experience which becomes autotelic. This model tends to group all dimensions that exist

on the literature. Theses dimensions may be grouped into three stages. The antecedent stages, the experience stage and the effect stage. The antecedent stage includes clear goals, immediate feedback and personal skills well suited to given challenges. The second stage includes merging of action and awareness, concentration on the task at hand, a sense of potential control and a loss of self-consciousness. The effect stage includes an altered sense of time, and experience which becomes autotelic. It concentrates on the effects that result from the flow state (Chen et al, 1999). To obtain an overall estimation about the flow state, we adopt this model for our research.

Thus, the fist hypothesis is:

H1: flow state influences positively the learning effectiveness.

2. 3 Involvement

The literature offers many definition of involvement. Arts (1999) states that the involvement notion is still swindle. She adds that the involvement may be confused with some concept like interest or engagement. Krugman (1965) defines the involvement a connection of some ideas or personal reference which someone feels about his life and a persuasive stimulus. Lastovicka (1979) defines the involvement as the relation that consumers make between products, values and brand name engagement. Additionally, Mitchell (1979) defines the involvement as an internal variable which indicates the importance of the excitation, the interest and the pulsing marked by a stimulus or a particular situation. Until the difference between the involvement definitions there is three major conclusions. Involvement depends on the relation that the subject makes with the object, to the degree that the subject gives to the object and the involvement is a stimulus result. When we talk about learning there are many stimuli. Graphics, animated picture and sound are stimuli examples.

The involvement is a dichotomy construct. The involvement is one-dimensional (Zaichowsky 1985, Lastovicka and Gardner 1978) vis-à-vis multidimensional (Laurent and Kapferer 1986, Mittal 1995). The involvement is a continuum of motivation, interest or activation (Brennan and Mavondo, 2000) vis-à-vis intrinsic. The involvement is durable vis-à-vis situational (Art, 1999). In our research, we consider that involvement is multidimensional, thus it's related to cognitive and psycho-sociologic aims (Gharbi, 1998). The involvement is durable, in fact, learner will be often exposed to the web site and spend some hours to hold the courses continue. Finally, the involvement is a dichotomy. In fact, the e-learner is either involve either no. The involvement is not related to the motivation. It's rather related to an aim search that is education.

Thus, the second hypothesis is:

H2: The involvement influences positively the flow state.

2. 4 Need for cognition

. The need for cognition concept was found by Cohen, Stotland and Wolfe (1955). They define need for cognition as the desire to structure some relevant situations significantly and integrated. The need for cognition represents the need to understand and make our environment experience intelligible. Cacioppo and Petty (1982) define the need for cognition as «.. the tendency for an individual to engage in and enjoy thinking…». Therefore, this concept implies an intrinsic interest research while someone is learning. Cacioppo and Petty (1982) state that it's a

motivational factor at the first degree. So that, the need for cognition is related to the individual internal desires. The need of cognition is a one-dimensional concept.

Therefore, the third hypothesis is:

H3: The need for cognition influences positively the flow state.

Hence, our conceptual model is presented as follow:

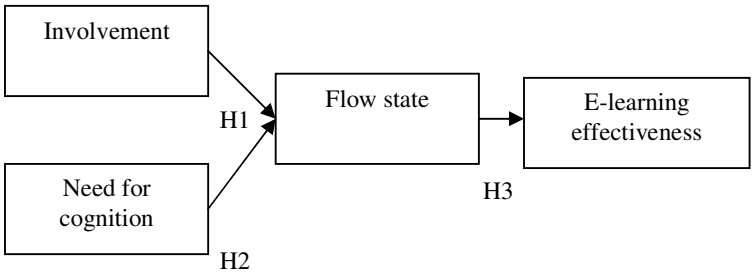


FIG. 1 – *Conceptual model*

3 Measures and Methodology

3.1 Measurement variables

To verify our conceptual model we construct a questionnaire based on scales measure according to our concepts. The scales are drawing from the literature. The following table shows the different scales chosen for our study:

VARIABLE	SCALED VARIABLE	DESCRIPTION	NUMBER OF ITEMS	ALPHA
Involvement	Zaïchkowsky scale (1994) Personal Involvement Inventory (P.I.I.) : The version corrected by the author.	Considers involvement as a multidimensional concept that has two dimensions: cognitive dimension and affective dimension.	10	[.91-.96]
Need for cognition	Cacioppo, J. T., Petty, R. E and	One-dimensional scale.	18	.90

	Chuan Feng Kao scale (1984).			
Flow state	Chen scale (2006)	Covers 7 dimensions from the Csikszentmihalyi model (1996): clear goals, immediate feedback, merging of action and awareness, concentration on the task at hand, a sense of potential control, a loss of –self-consciousness, an altered sense of time.	23	>.70

TAB. 1 – *Variables in the study*

According to the literature there is not scale measure of e-learning effectiveness. In spite of the different proposed measure, there are not models which unify these propositions. Consequently, some approaches and several tools and instruments are proposed. The famous approach is the comparison between the result emerged from the traditional learning with the results emerged from the e-learning.

Bremen et al (2003) claims that there are some (but surprisingly few) systematic studies that compare e-learning effectiveness with traditional learning and which are empirically robust. Those, that exist are mainly small-scale studies, often using a matched pairs design and are frequently of very specific instances of e-learning in which the e-learning methodologies are idiosyncratic and the conclusions are non-generalisable.

Figueira 2003 argue that globally, programme effectiveness can be evaluated through five evaluation approaches: (1) based on the programme goals, (2) based on the decision-making process, (3) goal-free approach, (4) based on an expert's knowledge, and (5) naturalistic approach. Therefore, we choose the based-program method. In fact, the aim for the learning system is that courses must be assumed by the student. To verify the achieving of this aim, we use the traditional method which is the assessment performance. Bremen et al (2003) claim that the study of performance is a good learning effectiveness indicator. This method is also used in many universities. Therefore, this method takes the student test mark as an indicator.

3.2 Methodology

The questionnaire was distributed to 423 students who belong to the high Institute of technology. The certificate obtained is "Business Management". We had given a questionnaire to the student in the classroom. The questionnaire reply was mandatory. We're interested to the level-one-student to eliminate the platform familiarity factor. We choose the site course of the "management initiation" because it's a basic course. The questionnaire consisted of 53 items which 2 gathered demographic information. The approximate time necessary to complete the questionnaire is 20 minutes. The majority of the students 75, 7% are under 22 years. After the collect of the questionnaire teachers gave us the students 'mark.

4 Analysis

4.1 Measure validity

We perform an exploratory factor analysis with varimax rotation to purify our scales. Some items were deleted because of their low communalities or they are equally correlated to two factors.

The following table shows the ACP results:

Scales	≠ Factors	Factors	α	Interpretation
P.I.I.	2	-"Cognitive involvement": item 2, 8 10, 5	,8275	The factors obtained match to the scale construct based on two dimensions
		-"Affective involvement": item 7,3, 4, 1, 9, 6	,7889	
Need for cognition	1	-Need for cognition" item 6, 10, 15, 11, 13, 14,1	,6323	We obtained one factor according to the one-dimensional construct of the scale.
Flow state	3	-"Clear goals, immediate feedback, concentration": Items 3, 1, 2, 8, 7, 4.	,7272 ,8542	The result obtained match with the Csikszentmihalyi model (1993) that state that flow state is divided into three stages. The factor one match with the antecedent stage. The factor two matches to the effect stage. The factor three match to the experience stage.
		-"Positivity of affects": Items 21, 22, 20	,6746	
		-"Telepresence, loss of self-consciousness, time distortion": Item 19, 16, 18, 15,13		

TAB. 2 – *Factor analysis*

4. 2 Test hypothesis

To test our model we used the linear regression. We have 3 hypotheses to verify. We consider that the hypothesis is verified if the probability is lower than 0,005. The results of our analysis are exposed following.

Hypothesis 1:

Learning effectiveness =, 166 Positivity of affects
(t= 3,139, p=, 002)
The relationship between learning effectiveness and flow state is only significant with "Positivity of affects". Hypothesis 1 is partially accepted.

Hypothesis 2:

Clear goals, immediate feedback, concentration =, 253 Cognitive involvement
(t= 4,682, p=, 000)
Positivity of affects =, 543 Cognitive involvement
(t=11,576, p=, 000)

The relationship between "Telepresence, loss of self-consciousness, time distortion" and involvement is not significant. On the other hand, the affective involvement has no relationship with the flow state. Only the cognitive involvement has a relationship with the flow state. So that we conclude that the course content is the most important thing for the student. This result explains also that the "Telepresence, loss of self-consciousness, time distortion" has not a significant relationship with cognitive involvement.
Therefore, the hypothesis too is partially accepted.

Hypothesis 3:

Clear goals, immediate feedback, concentration =, 332 need for cognition
(t=6,566, p=, 000)
Telepresence, loss of self-consciousness, time distortion =, 247 need for cognition
(t=4,756, p=, 000)
The relationship between "Positivity of affects" and need for cognition is not significant. So, the third hypothesis is partially accepted.

Discussion:

Our conclusion is that our overall model is partially significant. Some factors are positively related and some others are not. The cognitive involvement is positively related to the antecedent stage. This explains that in the beginning of experience or web site navigation, the first objective is learning. Students do not worry about the ergonometric aspects. On the experience stage they are submerged by the course and forget why they are in front of their PC. The most important thing is the course content. Therefore, we didn't find a significant relationship between involvement and flow state. Nevertheless, there is a significant relationship between the effect stage and the involvement. The effect stage is produced at the end of experience. So that, we can conclude that in the end the students will be not well concentrated as the experience stage. Consequently, they evaluate their experiences. They do a comparison between their objectives and their feeling. They think so about their involvement level at the beginning and their feeling at the end. The involvement-flow state relationship is therefore significant at the end stage.

On the other hand, we notice that the need for cognition has a positive relationship with the flow state at the antecedent stage and less at the experience stage. This shows that the need for

cognition is present at the beginning for experience and vanishes gradually. Moreover, the cronbach's alpha is lower between the need for cognition and "telepresence, loss of self consciousness". At the end stage there is not a significant relationship.

Regarding, the relationships between the flow state and the learning effectiveness is significant at the experience stage. In fact, at the experience stage the student is fully emerged by the Web site and there is a little attention given to other thing. The student is totally concentrated. In some situations, students can assimilate the course continues and therefore the learning will be efficient. The low cronbach's alpha value shows that learning effectiveness is influenced by other variables. In fact, student don't receive a 100% e-learning, they just received 20% e-learning.

5. Conclusion

Our study allows to determine the relationship between the learning effectiveness and some individual factors. We conclude that the flow state has a positive influence on the learning effectiveness. The flow state is influenced by the involvement and the need for cognition. Therefore, we conclude that the conceptual model is significant. The learning effectiveness is influenced by individual factor and the state lived during the learning experience.

Therefore, the UVT should find solutions to sensitize student about the importance of this new mode. Consequently, the learning effectiveness will increase if their involvement and their need for cognition are high.

Some limits are noticed. The flow state is not influenced only by personal factors. Some aspects like technical characteristics have indeed in influence. On the other hand, the learning effectiveness should be influenced by other factors beside the flow state. Further, students take just 20% e-learning. Therefore, our conclusions are limited, and can't be generalized to other context. Nevertheless, they can be generalized on the UVT context because in spite of some difference between web site courses they resemble each other.

Future researches should focus on determining the antecedents related to technical aspects. In fact, every e-learning platform has special characteristics. We should make several investigations on several platforms to determine the e-learning effectiveness antecedent. Thus, universities can insure good education.

6. References

Arts, N, (1999), Le Concept d'Implication : Une Revue de la littérature. www.scholar.google.com

Bersin,J, (2002), Measuring E-learning's Effectiveness A five-step program for success. www.google.com

Bremen, U, Hughes, J, and Attwell, G, (2003), A Framework for the Evaluation of E-Learning, European Seminars - Exploring models and partnerships for e-Learning in SMEs . www.google.com

Brennan, L and Mavondo, F, (2000), Involvement: An Unfinished Story?, Anzmac Visionary Marketing for the 21st Century: Facing the Challenge 132.

Cacioppo, J.T., Petty, R. E, and Chuan F.K, (1984), The Efficient Assessement of Need for Cognition, Journal of personality Assessement. Vol. 48, N°. 3.

Cacioppo, J.T and Petty, R.E, (1982), The need for cogntion, Journal of presonality and social Psychology, Vol. 42, N° 1, pp. 116-131.

Chen , H, (2006), flow on the net–detecting Web users positive affects and their flow states, Computers in Human Behavior, Vol. 22, pp. 221-233.
Chen, H, Wigand, R.T and Nilan, M.S, (1999), Optimal experience of Web activities, Computers in Human Behavior, Vol. 15, pp. 585-608.
Ettis. S, (2005), L'atmosphère des sites web marchand : impact de la couleur, des animations et de la musique sur les réponses du consommateur, Tutorat ALM Toulouse. Elworthy, A, (2004), La théorie constructiviste de l'apprentissage, la revue de la Fédération canadienne des services de garde à l'enfance, Vol. 18, N°. 2.
Figueira, E, (2003), Evaluating the Effectiveness of E-Learning Strategies for Small and Medium Enterprises, European Seminars - Exploring models and partnerships for eLearning in SMEs
Gharbi, J, (1998), Les facteurs qui influencent les processus décisionnels des consommateurs lors d'un achat par Internet. http://www.irec.net/publications/262.pdf.
Hoffman, D.L and Novak, T.P, (1996), Marketing in Hypermedia Computer-Mediated Environments: Conceptual Foundations, Journal of Marketing, Vol 60, pp. 50-68.
Kirkpatrick, D.L, (1996), Techniques for Evaluating Training Programs, Training & Development. www.scholar.google.com
Krugman, H, (1965), The impact of television advertising: Learning without involvement, Public Opinion Quarterly, Vol. 29, N°. 4, pp. 349-356.
Laurent, G and Kapferer, J.N , (1986), Les profits d'implication, Recherche et Application en Marketing, Vol. 1, N°. 1, pp. 41-58
Lastovicka, J. L. (1979), Questioning the concept of involvement defined product classes, Advances in Consumer Research, Vol. 6, pp. 174-179.
Lastovicka, J and Gardner, D, (1978), Low involvement versus high involvement cognitive structures, Advanced in Consumer Research, 5, éd. H.K. Hunt, Ann Arbor, Michigan, Association for consumer Research, pp.87-97.
Maurer, H and Sapper, M, (2001), E-Learning Has to be Seen as Part of General Knowledge Management. http://www.scis.nova.edu/~hafnerw/KM/KMRESOURCES/Maurer_Sapper.pdf.
Mitchell, A, (1979), Involvement: a Potentially Important Consumer Behavior, Avances in Consumer Research, 6, éd. W. L. Wilkie, Ann Arbor, Michigan, Association for consumer Research, pp. 191-196.
Mittal, B., (1995), A Comparative Analysis of Four Scales of Consumer Involvment, Psychology and Marketing, Vol. 12, N°.7, pp. 663-682.
Novak, T.P, Yung, Y and Hoffman, D.L, (1999), Measuring the Customer Experience in Online Environments: A Structural Modeling Approach. http://www2000.ogsm.vanderbilt.edu/
Pearce, J.M, Ainley, M et Howard, S, (2005), The ebb and flow of online learning, Computers in Human Behavior, Vol. 21, pp. 745–771.
Pilke, E.M , (2004), Flow experiences in information technology use, Journal of Human-Computer Studies, Vol. 61, pp. 347-357.
Trevino, L.K. and Webster, J. (1992), Flow in computer mediated communication, Communication Research, Vol. 19, No. 5, pp. 539-73.
Webster, J., Trevino, L.K. and Ryan, L. (1993), The dimensionality and correlates of flow in human-computer interactions, Computers in Human Behavior, Vol. 9, pp. 411-26.

Zhang, D, Zhou, L, Briggs , R and Nunamaker, J.F, (2005), Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness, information & Management, Vol. 43, pp.15-27.

Zairi, A and Jallouli, B , (2004), Étude comparative des modèles d'apprentissage en EAD et leur application dans l'expérience des ISET en Tunisie. www.google.com.

Zaichwosky, J.L, (1985), Measuring the involvement construct, Journal of Consumer Reasearch, Vol. 12 , N°. 3, pp. 341-352.

Zaichwosky , J. L., (1994), The Personal Involvement Inventory : Reduction, Revision, and application to Advertising, Journal of Advertising, Vol. 23, N°4 , pp. 59-70.

« Wikipédia » 2006

UVT Web site: http://pf-fc.uvt.rnu.tn

Index des auteurs

ASD'08

Troisième Atelier sur les Systèmes Décisionnels
10 et 11 octobre 2008, Mohammedia, Maroc