Modélisation Multidimensionnelle :

Vérification Formelle de Schéma d'Hiérarchie

Ali SALEM, Faiza GHOZZI, Hanene BEN-ABDALLAH

Laboratoire Miracl
Institut Supérieur d'Informatique et de Multimédia de Sfax
Route de Tunis Km 10 B.P. 242 SFAX 3021
TUNISIE

al_salemfr@yahoo.fr , Jedidi_Faiza@yahoo.fr, Hanene.Benabdallah@fsegs.rnu.tn

RÉSUMÉ. Nous proposons une démarche formelle de vérification de schéma d'hiérarchie des modèles multidimensionnels. Cette démarche repose sur une spécification formelle en Z du méta modèle *Hiérarchie* validée à l'aide de l'outil Z/Eves. La formalisation de l'hiérarchie définit un ensemble de contraintes au niveau méta modèle que tout modèle d'hiérarchie doit satisfaire aussi bien au niveau structurel qu'au niveau des instances de l'hiérarchie. Dans cet article, nous appliquons la démarche générique pour vérifier les contraintes structurelles.

ABSTRACT. We propose a formal verification process for hierarchy schema in multidimensional models. This process rests on a formal specification in Z of the hierarchy meta model validated with the theorem prover Z/Eves. The formalization of the hierarchy defines a set of constraints at the meta model level that every hierarchy model must satisfy both at the structural level and the instance level. In this paper, we apply the generic verification process to verify the structural constraints.

MOTS-CLÉS : Modèle multidimensionnel, OLAP, Contraintes, Mèta-Modèle, Spécification formelle, Hiérarchie.

KEYWORDS: Multi-dimensional model, OLAP, Constraints, Meta-Model, Formal Specification, Hierarchy.

1. Introduction

Dans les systèmes de traitements analytiques en ligne, reconnus par les systèmes OLAP « On Line Analytical Processing », 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 d'analyse (Dimensions) représentés par des perspectives d'analyse (Hiérarchies). Par ailleurs, la représentation de plusieurs sujets d'analyse dans le même schéma multidimensionnel facilite la corrélation des analyses. Un tel modèle (Constellation) regroupe plusieurs faits partageant des dimensions.

En plus, les modèles multidimensionnels basés sur les hiérarchies multiples nécessitent la spécification de contraintes sur la structure de ces hiérarchies [1] [2]. Le respect de ces contraintes permet d'agréger les données analysées selon les différentes granularités offertes en conservant l'intégrité des résultats [1] [2]. Par ailleurs, l'expression de ces contraintes est insuffisante pour aider le concepteur à valider son schéma, une démarche de vérification du schéma multidimensionnelle s'avère incontournable pour pouvoir exploiter ces contraintes.

Dans ce cadre, nous avons proposé une spécification formelle des concepts multidimensionnels [11] en langage Z [9]. Dans cet article, nous nous intéressons à la bonne formulation du concept hiérarchie et à la proposition d'une démarche semi-automatique de vérification de ces contraintes afin de répondre à ces besoins d'analyse.

Au cours de notre étude des travaux antérieurs dans ce domaine, nous nous sommes intéressés aux contraintes et à leurs différentes classifications. Parmi ces travaux, nous distinguons le modèle GMD [3] dans lequel les auteurs présentent un ensemble de contraintes que nous classons en deux catégories : les contraintes liées à la création des cubes et les contraintes liées à l'agrégation. [4] présente un modèle multidimensionnel orienté objet conçu en UML. Les auteurs, dans ce papier, classifient les contraintes multidimensionnelles en deux catégories : les contraintes dites d'emplacement pour la construction du cube et les contraintes d'agrégation. Ces contraintes sont formulées en langage naturel. Dans [1], il s'agit d'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. [5] présente une extension d'UML qui permet de représenter les propriétés structurelles des modèles Multidimensionnels au niveau conceptuel. Les auteurs expriment ces contraintes en OCL (Object Constraint Language) mais ne proposent pas de démarche de vérification de ces contraintes. [6] présente une classification assez claire des contraintes multidimensionnelles.

Les contraintes exprimées dans ces travaux diffèrent, d'une part, dans le niveau d'expression des contraintes (méta modèle ou modèle) et, d'autre part, dans le niveau de

vérification ou de préservation des contraintes. En outre, il n'y a pas de consensus sur l'ensemble des contraintes. Ce qui peut mener à des résultats d'analyse incohérents.

L'objectif du présent travail est d'aboutir à une démarche formelle de vérification de schéma de hiérarchie. Ainsi, nous proposons une spécification formelle consistante du concept hiérarchie et une démarche de vérification de schéma d'hiérarchie.

2. Spécification Formelle en Langage Z

La spécification formelle de notre modèle 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 [9] vu son pouvoir d'expression et la disponibilité gratuite de son outil de vérification Z/eves.

Avant d'introduire la formalisation du concept d'hiérarchie, 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 :

$$Determine: AttDim \leftrightarrow AttDim$$

La relation *Relval* met en correspondance les valeurs des attributs: $Relval: Dom \leftrightarrow Dom$

Chaque attribut d'une dimension comprend un ensemble fini de valeurs (*Dom*) et un poids reflétant son importance :

AttDim

$val: \mathbb{F}\ Dom$

$poids: Type$

Ainsi, la formalisation du schéma *Hierarchie* : où : *N* est le nom de la hiérarchie ; *Att* est un ensemble fini d'attributs dimension *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.

La partie prédicat du schéma *Hierarchie* regroupe les contraintes exprimées au niveau méta modèle, *i.e.*, qui doivent être satisfaites par tout modèle d'hiérarchie. Ces contraintes peuvent être classées en deux catégories : 1) Les contraintes liées à la structure décrivant les règles d'ordonnancement des attributs dans une hiérarchie ; et 2) les contraintes liées aux instances décrivant les relations entre les différentes valeurs des attributs. Le Tableau 1 explique les catégories de contraintes.

Hierarchie

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

(1) # ParamH ⩾ 2
(2) $\exists_1$ x: Att • x . poids = ID
(3) $\exists_1$ x: Att • x . poids = All
(4) ran ParamH = Att \ { y: Att | y . poids = Faible }
(5) ∀x: Att • x ∈ ran ParamH ∧ ParamH 1 = x ⇒ x . poids = ID
(6) ∀x: Att • x ∈ ran ParamH ∧ ParamH (# ParamH) = x ⇒ x . poids = All
(7) ∀i, j: 1 .. # ParamH | i ≠ j • ParamH i ≠ ParamH j
(8) ∀x, y: Att; i: 1 .. # ParamH - 2 | ParamH i = x ∧ ParamH (i + 1) = y • (x, y) ∈ Determine
(9) ∀i, j: 1 .. # ParamH | j = i + 1 • ∃v, w: Dom | v ∈ (ParamH i) . val ∧ w ∈ (ParamH j) . val • (v, w) ∈ Relval
(10) ∀i, j: 1 .. # ParamH; v: Dom | j = i + 1 ∧ v ∈ (ParamH i) . val • ∃w1, w2: Dom • w1 ∈ (ParamH j) . val ∧ w2 ∈ (ParamH j) . val ∧ (v, w1) ∈ Relval ∧ (v, w2) ∈ Relval ⇒ w1 = w2
(11) ∀v: Dom; i: 1 .. # ParamH - 1 | v ∈ (ParamH i) . val • ∃w: Dom | w ∈ (ParamH (i + 1)) . val • (v, w) ∈ Relval
(12) ∀v, w: Dom; x, y: Att • v ∈ x . val ∧ w ∈ y . val ∧ (v, w) ∈ Relval ∧ x . poids ≠ Faible ∧ x . poids ≠ All ∧ y . poids ≠ Faible ⇒ x ∈ ran ParamH ∧ y ∈ ran ParamH

Contraintes liées à la structure d'une hiérarchie	**Contraintes liées aux instances d'une hiérarchie**
(1) Hiérarchie non vide [7] : Chaque hiérarchie possède au moins deux niveaux de paramètres ; *ID* et l'attribut *All*. (2) Unicité de l'identifiant [7] : Une hiérarchie possède un et un seul identifiant. Il est l'attribut de granularité la plus fine (5). (3) Unicité de l'attribut All [7] : Il est défini pour clôturer une hiérarchie. All est l'attribut de plus haute granularité (6). (7) Acyclicité [3] [4] [1] [8] [6] : Elle interdit l'existence d'un cycle au niveau des hiérarchies. (8) Connexion vers le haut [1] [6] : cette contrainte exprime que tous les paramètres, sauf All, possèdent au moins un père.	(9) Contrainte de dépendance hiérarchique (Soundness) [6] : Pour chaque couple de paramètres de la hiérarchie, il existe au moins deux valeurs appartenant à ce couple tel que ces deux valeurs sont dépendantes. (10) (11) Partition [1] [6] : À chaque valeur d'un paramètre correspond une et une seule valeur parmi celles de chaque paramètre successeur dans la hiérarchie. (12) Connectivité [1] [6] (Completness) : Cette contrainte exprime que l'existence d'une relation entre les valeurs des attributs implique l'existence d'une relation hiérarchique entre les paramètres.

Tableau 1. *Explication des contraintes d'hiérarchie.*

3. Démarche de Vérification de schémas

Dans cette phase, nous présentons la démarche de vérification qu'un schéma particulier (modèle) est correct par rapport au méta-modèle contraint. L'idée de base consiste en une démonstration du théorème d'initialisation [9]. Le principe de ce théorème consiste à instancier un exemple et à prouver qu'il est correct.

3.1 Description d'un schéma d'une hiérarchie en langage Z

Nous prenons l'exemple de la hiérarchie Clas_Cons de la dimension Véhicule (Figure 1). Parmi les attributs de la dimension Véhicule, nous citons Immat, Modèle, Marque, Puissance et l'attribut All. L'attribut Immat est l'identifiant de la hiérarchie ; les attributs Modèle et Marque sont des paramètres ; et Puissance est un attribut faible. Chaque attribut contient un ensemble de valeurs liées les uns aux autres suivant la relation Relval.

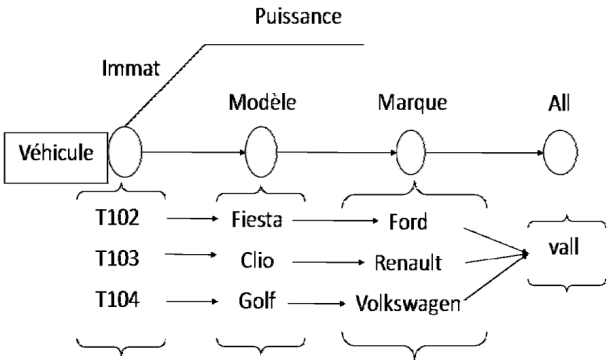

Figure 1. *La hiérarchie Clas_Cons de la dimension Véhicule.*

La description de ce schéma en Z se fait en deux phases. La première phase déclare les différents attributs et leurs contraintes. Nous réalisons cette phase à travers un axiome box (Figure 2) contenant deux parties : 1) la déclaration des constituants de l'hiérarchie ; et 2) les prédicats représentant les contraintes qui seront définies par Z\Eves comme des axiomes. Ces axiomes guident la démonstration du théorème d'initialisation. Dans la deuxième phase, nous définissons le schéma *InstanceHierarchie* (Figure 3) qui jouera le rôle d'une instance de *Hiérarchie*. Par ailleurs, nous devons affecter les différentes valeurs nécessaires aux différents ensembles déjà déclarés dans la spécification du méta modèle *Hierarchie*.

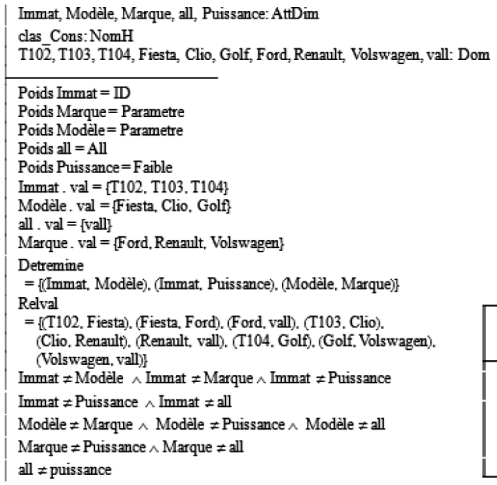

Figure 2. *Axiome box*

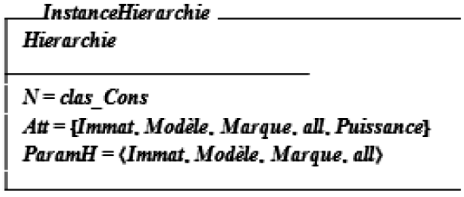

Figure 3. *Instance d'hiérarchie*

3.2 Preuve du théorème d'initialisation

Le théorème d'initialisation est le suivant :

theorem *HierarchieClas_Cons*
$\exists Hierarchie \bullet InstanceHierarchie$

La preuve du théorème d'initialisation valide qu'un exemple particulier d'hiérarchie satisfait les contraintes du méta modèle (une à une dans leur ordre de définition). Ainsi, nous avons déclaré les contraintes de façon à optimiser la preuve de ce théorème. Par ailleurs, cette preuve reste toujours très épineuse pour un concepteur multidimensionnel, non expert en Z.

Pour guider cette preuve, nous avons défini une démarche générique. Cette démarche prend en entrée le méta modèle validé et un exemple d'hiérarchie à vérifier. Elle extrait à partir de la base de théorèmes générés automatiquement par Z/eves les axiomes pertinents aux concepts de l'exemple. En plus, elle réécrit un ensemble d'axiomes génériques que nous avons dégagés durant la preuve de plusieurs exemples (*e.g.,* les axiomes de déclaration des poids des attributs). Ces axiomes seront appelés en fonction de l'ordre des contraintes et selon les étapes illustrées sur l'exemple de la Figure 1. La démonstration commence par la commande *invoke* suivie par *prove by reduce*, qui vérifient automatiquement la contrainte (1) d'hiérarchie non vide. Par conséquent, notre démarche commence réellement à partir de la contrainte (2). Les commandes *case* et *next* servent à traiter les contraintes une par une. Ainsi, le passage de vérification d'une contrainte à une autre est assuré par la commande *next*.

- Invoke
- Prove by reduce
- Case
- Vérification de C2:
  - Introduire les axiomes de poids en utilisant la commande use
  - Instancier l'axiome de l'identifiant par la commande instantiate
  - prove
- Vérification de C3
  - La même démarche que C2 mais en instanciant l'axiome de l'attribut all
- Vérification de C4
  - Introduire les axiomes de poids en utilisant la commande use
  - Withe normalization reduce
- Vérification de C5
  - Introduire l'axiome du poids del'identifiant en utilisant la commande use
  - Prove
- Vérification de C6
  - Introduire l'axiome du poids de l'attribut all en utilisant la commande use
  - Prove
- Vérification de C7
  - Introduire les axiomes de distinction des attributs en utilisant la commande use
  - Withe normalization reduce
- Vérification de C8
  - Introduire l'axiome de dépendance fonctionnelle nommé *détermine* en utilisant la commande use
  - Apply extensionality2 to prédicate *détermine*
  - Withe normalization reduce
- Refaire C7
- Refaire C8

Cette démarche est générique. Ainsi, en l'appliquant sur l'exemple de la Figure 1, nous avons démontré que la hiérarchie *Clas_cons* satisfait les contraintes structurelles.

5. Conclusion

Dans ce papier, nous avons rassemblé toutes les contraintes Méta-Modèle du concept hiérarchie. Ces contraintes sont jugées indispensables pour maintenir la cohérence des données à agréger et pour assurer l'intégrité des structures et des données

multidimensionnelles par rapport à ces contraintes. Par ailleurs, nous avons présenté une démarche de validation de schéma de hiérarchie de modèles multidimensionnels.

Nous sommes entrain de développer un environnement permettant la conception d'un schéma multidimensionnel et la vérification assistée de sa consistance. Cet environnement se base sur la formalisation de tous les concepts multidimensionnels. En plus, il reposera sur la définition formelle de règles de transformation de modèles : du multidimensionnel vers Z.

6. Bibliographies

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

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

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

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

[5] Luján S, Trujillo J., Song, "Extending the UML for Multidimensional Modeling" The Unified Modeling Language : Proc. 5th International Conference, Dresden, Germany, September 30 - October 4, 2002.

[6] Ghozzi F., Ravat F., Teste O., Zurfluh G., "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, 2003.

[7] Ghozzi F., "Conception et Manipulation de Base de Données Dimensionnelles à Contraintes". Thèse de l'Université Paul Sabatier - Toulouse III, novembre 2004.

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

[9] J.M. Spivey." The Z Notation : a Reference Manual". Prentice-Hall,1992.

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

[11] Salem A.,Ghozzi F., Ben Abdallah H., Zurfluh G., "Spécification Formelle du Modèle Multidimensionnel à Contraints", Atelier Systèmes d'Information Décisionnels INFORSID 2006, Hammamet, Tunisie, p. 22-27 31Mai 2006.