

# Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

Ali Salem<sup>\*</sup>, Faiza Ghozzi<sup>\*\*</sup>, Hanene Ben-Abdallah<sup>\*\*\*</sup>

<sup>\*</sup> Faculté des Sciences de Gafsa, Tunisie

[salem.aly@gmail.com](mailto:salem.aly@gmail.com)

<sup>\*\*</sup> Institut Supérieur d'Informatique et de Multimédia de Gabès, Tunisie

[Faiza.Jedidi@planet.tn](mailto:Faiza.Jedidi@planet.tn)

<sup>\*\*\*</sup> Faculté des Sciences Economiques et de Gestion de Sfax, Tunisie

[hanene.benabdallah@fsegs.rnu.tn](mailto:hanene.benabdallah@fsegs.rnu.tn)

**Résumé.** Comme tout autre modèle, les modèles multidimensionnels doivent adhérer à un ensemble de contraintes de bonne formation structurelle et sémantique afin de garantir l'exactitude des analyses. Une partie de ces contraintes régissent la relation d'un modèle multidimensionnel avec sa source de données. Dans ce papier, nous proposons un cadre formel pour la spécification, la vérification syntaxique et la validation d'un schéma multidimensionnel par rapport à celui de sa source de données.

## 1 Introduction

Il est évident que le succès du développement d'un système OLAP dépend, entre autre, de la qualité de son modèle conceptuel. En effet, comme tout autre modèle, les modèles multidimensionnels doivent adhérer à un ensemble de contraintes de bonne formation structurelle et sémantique. Le respect de ces contraintes permet, par exemple, d'agrèger les données analysées selon les différentes granularités offertes en conservant l'intégrité des résultats (Hurtado et al., 2002) (Lechtenbörgel et al., 2003).

Dans ce contexte, plusieurs travaux ont dégagé des contraintes dans leur majorité informelles (cf., (Hurtado et al., 2002) (Luján et al., 2002)). Les quelques travaux qui ont formalisé des contraintes (cf., (Franconi et al., 2004) (Abelló et al., 2006)) n'offrent pas de support méthodique pour démontrer la satisfaction des contraintes. Cependant, outre l'expression formelle précise des contraintes, le concepteur décisionnel a besoin d'une démarche de vérification sans pour autant devenir expert dans la méthode formelle exprimant les contraintes. Autrement dit, le choix du langage formel utilisé pour exprimer les contraintes doit tenir compte de deux critères : le pouvoir d'expression du langage et le support en termes de démarche et d'outils d'analyse.

Comme argumenté dans nos travaux antérieurs (Salem et al., 2006) (Salem et al., 2008), le langage formel Z (Spivey et al., 1992) jouit d'un pouvoir d'expression qui permet d'exprimer tous les concepts multidimensionnels. De plus, grâce à son outil démonstrateur Z/Eves (Saaltink, 1999), un concepteur peut analyser une spécification formelle. Notre objectif est de définir une aide méthodique pour l'analyse de la satisfaction des contraintes des schémas multidimensionnels. Cette aide exploite Z/Eves sans imposer au concepteur multidimensionnel des compétences en Z. Dans cette perspective, nous avons proposée une démarche pour la vérification de la satisfaction des contraintes structurelles (Salem et al., 2006) (Salem et al., 2008). L'objectif de ce papier est de compléter cette démarche pour

## Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

permettre la vérification de la chargeabilité d'un schéma multidimensionnel à partir d'une source relationnelle.

En effet, la vérification de la bonne formation structurelle d'un schéma multidimensionnel ne suffit pas pour juger que ce dernier puisse être alimenté à partir d'une source. Nous définissons un entrepôt de données comme *chargeable* si le processus d'alimentation de l'entrepôt ne rencontre pas des données dans la source qui violent l'intégrité des contraintes du schéma multidimensionnel. Afin de vérifier qu'un entrepôt est chargeable, nous devons, d'une part, contrôler les contraintes sémantiques et, d'autre part, vérifier la conformité du schéma par rapport à sa source globale. (Ainsi, dans nos travaux, nous ne traitons pas le problème d'intégrité de sources.) Nous appelons le deuxième type de vérification la *validation* du schéma multidimensionnel par rapport à une source.

Note démarche de validation repose sur une confrontation du schéma multidimensionnel, basée sur les besoins des décideurs par rapport à celui de sa source. Elle ne reconstruit pas le premier en adoptant une démarche ascendante tel qu'il est reconnu au niveau des approches mixtes de conception de modèle multidimensionnel. Plutôt, la validation se réduit à la vérification de certaines contraintes, nommées contraintes de conformité. Nous définissons ces contraintes par le biais des travaux traitant la conception de modèle multidimensionnel en adoptant une démarche ascendante.

Afin de mettre notre démarche de validation dans son contexte, nous passons en revue, dans la section 2, les travaux pertinents aux contraintes des modèles multidimensionnels et à leur formalisation. Ensuite, dans la section 3, nous présentons une spécification formelle en Z du méta modèle multidimensionnel ; cette spécification formelle intègre les contraintes de bonne formation structurelle de tout schéma multidimensionnel. Dans la section 4, nous introduisons les contraintes de conformité, comme proposées par les méthodes de conception ascendantes. Enfin, dans la section 5, nous formalisons ces contraintes en Z et nous illustrons les étapes de la validation par le biais d'un exemple.

## 2 Etat de l'art

Durant nos études des travaux antérieurs dans ce domaine, dans un premier temps, nous avons focalisé nos efforts sur la collecte des contraintes multidimensionnelles. Nous résumons ces contraintes dans le tableau Tab1.

Dans un deuxième temps, nous nous sommes intéressés aux travaux de conception de modèle multidimensionnel adoptant une démarche ascendante et ceux adoptant une démarche mixte afin de pouvoir extraire certaines contraintes servant à la vérification de la conformité du schéma multidimensionnel par rapport à celui de la source. Par ailleurs, en adoptant une méthode de conception descendante, le concepteur néglige le schéma de la source. Ainsi, il est fort probable d'avoir un schéma multidimensionnel non conforme à celui de la source. Ce qui engendre des problèmes lors de l'extraction des données. En effet, en se basant uniquement sur les besoins des décideurs dans la modélisation multidimensionnelle nous risquons d'obtenir un schéma instable (Moody et al, 2000).

Dans (Pang et al, 2004), les auteurs génèrent les mesures, les dimensions et les hiérarchies à partir des requêtes de base données opérationnelles et de la dépendance fonctionnelle entre les attributs. Le schéma multidimensionnel résultant respecte certaines contraintes telles que l'orthogonalité et l'acyclicité.

Dans (Moody et al, 2000), les auteurs décrivent une méthode de conception de modèles multidimensionnels à partir de schémas entité-association. Elle consiste, en une première étape, au classement des entités selon différentes catégories. Ensuite, il s'agit d'identifier les hiérarchies. Enfin, les auteurs regroupent ces hiérarchies et agrègent les données transactionnelles pour former le modèle multidimensionnel.

Concept	Contraintes	Description
Hiérarchie	Unicité de l'identifiant (Ghozzi et al., 2004)	La hiérarchie contient un identifiant unique.
	Unicité de l'attribut <i>All</i> (Ghozzi et al., 2004)	La hiérarchie contient un paramètre unique libellé <i>All</i> .
	L'identifiant a la granularité la plus fine (Ghozzi et al., 2004)	Dans une hiérarchie, les paramètres sont classés de la granularité la plus fine à la plus haute. L'identifiant est le paramètre de granularité la plus fine.
	<i>All</i> de plus haute granularité (Ghozzi et al., 2004)	<i>All</i> est le paramètre de plus haute granularité. Il sert à clôturer une hiérarchie.
	Hiérarchie non vide (Ghozzi et al., 2004)	Chaque hiérarchie a au moins deux niveaux de paramètres : l'identifiant et le paramètre <i>All</i> .
	Acyclité (Franconi et al., 2004)(Abelló et al., 2006)(Hurtado et al., 2002) (Ghozzi et al., 2004)	Cette contrainte rejette l'existence d'un cycle dans la hiérarchie.
	Connexion vers le haut (Hurtado et al., 2002) (Ghozzi et al., 2004)	Tous les paramètres, sauf <i>All</i> , possèdent au moins un père (un paramètre de granularité moins fine).
Dimension	Non dimension vide (Hurtado et al., 2002) (Ghozzi et al., 2004)	Chaque dimension possède au moins une hiérarchie.
	Identifiant unique	La dimension contient un identifiant unique.
	Unicité de <i>All</i>	La dimension contient un unique paramètre <i>All</i> .
Fait	Orthogonalité des dimensions (Hurtado et al., 2002)	Les dimensions d'un fait doivent être indépendantes fonctionnellement.
	Fait non vide (Ghozzi et al., 2004)	Chaque fait contient au moins une mesure.
	Fait non isolé (Ghozzi et al., 2004)	Chaque fait est relié à au moins une dimension.

TAB. 1 – Liste des contraintes multidimensionnelles du schéma en étoile.

Dans (Cabibbo et al, 1998) et (Golfarelli et al. 1998), les auteurs proposent également une méthode de conception visant à construire un schéma MD à partir d'un schéma entité-association. Ils construisent les faits à partir des associations n-aires et les dimensions à partir des entités.

Dans (Feki et al, 2007) les auteurs présentent une méthode ascendante de construction de schémas en étoile à partir d'une source relationnelle. Ils étudient la structure des relations et proposent une classification en relation-associations et relation-entités permettant de construire le schéma multidimensionnel.

Dans (Mazon et al, 2007), les auteurs proposent une approche mixte de développement de DW. Ils construisent le modèle multidimensionnel à partir des besoins des utilisateurs et

ensuite ils vérifient l'exactitude vis à vis des sources de données en utilisant un ensemble de règles en QVT (Query/View/Transformation) basées sur les formes normales multidimensionnelles.

Selon (Romero et al, 2009), la meilleure façon pour la recherche automatique des sujets et des dimensions d'analyse est l'identification des dépendances fonctionnelles à partir des sources de données. Ainsi, Ils proposent un algorithme permettant de découvrir les dépendances fonctionnelles à partir de l'ontologie de domaine.

Selon notre étude, nous avons remarqué l'absence de consensus sur l'ensemble de contraintes exprimées. Aussi, l'absence d'un moyen rigoureux de vérification de ces contraintes. Pour vérifier la conformité des modèles par rapport à la source la solution est généralement dans la confrontation du schéma résultant de la démarche ascendante et celui résultant de la démarche descendante. Dans cet article, nous offrons un moyen rigoureux de vérification et de validation de modèles multidimensionnels. Un cadre formel facilitant la lisibilité et le raisonnement sur un tel modèle. En outre, pour tester la conformité des modèles par rapport à la source, selon notre approche, il n'est plus nécessaire de confronter le schéma résultant de la démarche ascendante et celui résultant de la démarche descendante. Le problème se réduit à vérifier certaines contraintes que nous avons nommé contraintes de conformité.

### 3 Spécification formelle en langage Z du méta-modèle

Parmi les modèles multidimensionnels, nous distinguons les modèles en flocon de neige, les modèles en constellation et les modèles en étoile. Dans nos travaux, nous nous intéressons aux modèles en étoile. Les concepts de base de ce modèle sont le concept de *Fait* qui représente le sujet d'analyse, le concept de *Dimension* qui représente un axe d'analyse incluant des *Hiérarchies* qui construisent les perspectives d'analyse.

La spécification formelle du méta-modèle des schémas en étoile permet d'exprimer les contraintes de manière exacte et précise offrant, par conséquent, le moyen de les vérifier. Le langage de spécification choisi est le langage Z (Spivey et al., 1992) vu son pouvoir d'expression et la disponibilité gratuite de son outil de vérification Z/Eves (Saaltink, 1999).

Avant d'introduire la formalisation des différents concepts, introduisons quelques définitions utiles : Les deux types *Name* et *Dom* désignent les ensembles de noms des différentes instances de concepts et de valeurs d'attributs. Le type libre *Weight* traduit la classe des attributs des dimensions et *TYPE* indique le type de l'attribut :

$$\begin{aligned} & [Name, Dom] \\ & Weights ::= Weak \mid Parametre \mid ID \mid All \\ & TYPE ::= Textuel \mid Number \mid Temporal \mid Boolean \end{aligned}$$

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

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

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

*ATTRIBUT*

```

val: F Dom
weigh: Weights
type: TYPE

```

### 3.1 Formalisation du concept hiérarchie

La formalisation du concept hiérarchie se traduit par le schéma *Z* nommé *Hierarchy*. La première partie du *schema-box* déclare les composants : *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 d'entiers.

*Hierarchy*

```

N: Name
Att: F ATTRIBUT
ParamH: seq ATTRIBUT
WeakH: ATTRIBUT ↔ ATTRIBUT

```

- (1)  $\#ParamH \geq 2$
- (2)  $\exists! x: Att \cdot x \cdot weigh = ID$
- (3)  $\exists! x: Att \cdot x \cdot weigh = All$
- (4)  $ran\ ParamH = Att \setminus \{ y: Att \mid y \cdot weigh = Weak \}$
- (5)  $\forall x: Att \cdot x \in ran\ ParamH \wedge ParamH\ 1 = x \Rightarrow x \cdot weigh = ID$
- (6)  $\forall x: Att \cdot x \in ran\ ParamH \wedge ParamH\ (\#ParamH) = x \Rightarrow x \cdot weigh = All$
- (7)  $\forall i, j: 1.. \#ParamH \mid i \neq j \cdot ParamH\ i \neq ParamH\ j$
- (8)  $\forall x, y: Att; i: 1.. \#ParamH - 2 \mid ParamH\ i = x \wedge ParamH\ (i + 1) = y$   
  - $(x, y) \in Determine$

La deuxième partie (prédicative) du schéma *Hierarchy* regroupe les contraintes exprimées au niveau du méta modèle, i.e., qui doivent être satisfaites par tout schéma de hiérarchie. Ces contraintes sont : (1) Hiérarchie non vide, (2) Unicité de l'identifiant, (5) qui est l'attribut de granularité la plus fine, (3) Unicité de l'attribut All et (6) qui est l'attribut de plus haute granularité, (7) Acyclicité, et (8) Connexion vers le haut.

### 3.2 Formalisation du concept dimension

La spécification formelle en langage *Z* du concept dimension se traduit par le schéma *Z* *Dimension* où la partie déclarative spécifie: *N* le nom de Dimension, *Hier* un ensemble fini d'hiérarchies, et *Att* un ensemble fini d'attributs formant les hiérarchies (*AttDim*).

Dans la partie déclarative, nous définissons les contraintes suivantes : (1) dimension non vide, (2) unicité de l'identifiant d'une dimension, (3) unicité de l'attribut All, (4) et (5) assurent que l'ensemble des attributs d'une dimension est l'union des attributs de ses hiérarchies.

## Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

<i>Dimension</i>	
	<i>N</i> : Name <i>Hier</i> : F <i>Hierarchy</i> <i>Att</i> : F <i>ATTRIBUT</i>
(1)	$Hier \neq \emptyset$
(2)	$\exists 1x: Att \cdot x \cdot weighth = ID$
(3)	$\exists 1x: Att \cdot x \cdot weighth = All$
(4)	$\forall x: Att \cdot \exists h: Hier \cdot x \in h \cdot Att$
(5)	$\forall h: Hier \cdot \forall x: h \cdot Att \cdot x \in Att$

### 3.3 Formalisation du concept fait

La formalisation du concept du fait (et par conséquent du schéma en étoile) se traduit par le schéma *Z Fact*. Dans sa partie déclarative, nous déclarons : *N* pour le nom du fait, *AttFact* pour l'ensemble de ses mesures, *DimAssocie* pour l'ensemble fini de ses dimensions. Dans la partie prédictive, nous définissons les trois contraintes : (1) et (2) fait non vide, (3) fait non isolé, et (4) orthogonalité des dimensions.

<i>Fact</i>	
	<i>N</i> : Name <i>AttFact</i> : F <i>ATTRIBUT</i> <i>Mesure</i> : F <i>ATTRIBUT</i> <i>DimAssocie</i> : F <i>Dimension</i>
(1)	$AttFact \neq \emptyset$
(2)	$Mesure \neq \emptyset$
(3)	$DimAssocie \neq \emptyset$
(4)	$\forall d: Dimension \mid d \in DimAssocie$ <ul style="list-style-type: none"> <li>• <math>\forall d2: Dimension; x1: d \cdot Att</math>  <math>\mid d2 \in DimAssocie \wedge d \neq d2 \wedge x1 \cdot weighth = ID</math></li> <li>• <math>\forall x2: d2 \cdot Att \mid x2 \cdot weighth \neq All \cdot (x1, x2) \notin Determine</math></li> </ul>

### 3.4 Validation des différents schémas Z

La validation d'un schéma *Z* consiste en la démonstration que les types de données sont correctement définis et que les contraintes (prédicats) ne sont pas contradictoires. Ceci est accompli habituellement en prouvant un théorème d'initialisation à travers lequel on démontre au moins l'existence d'un état initial.

A ce niveau, nous testons donc la consistance de notre formalisation du méta-modèle. La démarche consiste à instancier un exemple correct pour chaque concept et à prouver qu'il respecte les contraintes exprimées au niveau du schéma *Z* du concept.

Ainsi, dans une première étape, nous déclarons les différentes instances en utilisant les schémas *Z* appelés Axiome-box. Ce dernier se compose de deux parties : une partie déclarative et une partie prédicat. Dans la partie déclarative, nous définissons les différents objets et dans le prédicat nous définissons les différentes contraintes liées à ces objets. Chaque contrainte sera traduite par le démonstrateur de théorèmes *Z*\Eves en un axiome.

Ensuite, dans une deuxième étape, nous définissons le schéma  $Z$  *ConceptInstance* qui jouera le rôle d'une instance du concept. Par ailleurs, nous devons associer les différentes valeurs nécessaires aux différents ensembles déclarés au niveau du schéma  $Z$  de chaque concept.

Enfin, nous démontrons le théorème d'initialisation suivant:

**theorem** *ConsistenceConcept*  
 $\exists \text{Concept} \cdot \text{ConceptInstance}$

par le biais de l'outil Z/Eve qui est un démonstrateur semi-automatique.

## 4 Contraintes de conformité

Dans le cadre d'une démarche formelle de validation de schéma multidimensionnel par rapport à sa source, nous fixons comme objectif de tester la chargeabilité du schéma basé sur les besoins des décideurs. Ainsi, nous pallions la principale limite de la démarche descendante, à savoir, la prise en compte des données de la source. Nous constatons que les démarches mixtes proposées dans la littérature et visant à pallier cette limite (Cavero et al, 2001) (Carneiro et al, 2002) nécessitent une phase de reconstruction de schéma multidimensionnel en adoptant une démarche ascendante, ainsi qu'une phase de confrontation. Notre démarche simplifie ces phases et réduit la tâche du concepteur en réalisant automatiquement le test de chargeabilité du schéma multidimensionnel obtenu à partir de la démarche descendante et ceci en se basant sur les contraintes de conformité. Ces contraintes sont extraites des heuristiques de construction de schéma multidimensionnel selon une démarche ascendante proposées dans la littérature (Golfarelli et al,1998) (Moody et Kortink, 2000).

Le rôle de cette catégorie de contraintes est de contrôler la conformité des différents éléments (Fait, Dimension et Hiérarchie) d'un schéma multidimensionnel par rapport au schéma de la source.

Dans le cadre des travaux de notre équipe (Feki et al, 2007) adoptant une démarche ascendante de conception de schéma multidimensionnel, les auteurs ont proposé un ensemble d'heuristiques basé, d'une part, sur les règles de transformation du schéma entité-association vers le schéma relationnel, et d'autre part, sur les travaux de conception de schéma multidimensionnel à partir d'un schéma entité-association (Golfarelli et al,1998), (Moody et Kortink, 2000) (Pang et al., 2006).

Dans notre proposition, nous nous reposons sur les travaux de (Feki et al, 2007) pour définir les contraintes de conformité. Nous considérons dans ces travaux les sources qui se basent sur un schéma relationnel en troisième forme normale. Nous justifions notre choix du fait qu'une telle représentation élimine toute sorte de redondance de données et d'incohérence entre les informations gérées par le système d'information. Aussi, nous supposons que les noms des concepts utilisés dans le schéma multidimensionnel coïncident avec les concepts correspondants extraits de la source.

### 4.1 Contraintes de conformité liées au concept fait

Dans (Feki et al, 2007), les auteurs classe les faits en des faits pertinents et d'autres à pertinence faible. Ainsi, toute relation-association  $R$  contenant au moins un attribut numérique non clé (primaire ou étrangère) est un fait candidat pertinent nommé  $R$ . Toute

relation-entité R contenant au moins un attribut numérique non clé (primaire ou étrangère) est un fait candidat de faible pertinence nommé R. La différence entre une relation association et une relation entité se manifeste par le fait que la première est caractérisée par les clés étrangères.

Les mesures d'un fait correspondent aux attributs numériques non clé de la relation fait et n'appartiennent pas à d'autres relations. En outre, un attribut numérique appartenant à une relation parallèle à la relation fait peut être considéré comme une mesure. Une relation R1 est parallèle à R2 si les clés primaires de la première sont incluses ou égales à ceux de la deuxième.

Nous dégageons, ainsi, les contraintes suivantes :

- Contrainte de conformité de *Fait* : Le nom du fait coïncide avec le nom d'une relation contenant au moins un attribut numérique non clé (ni primaires, ni étrangères).
- Contrainte de conformité des *Mesures* : Une mesure d'un fait est un attribut numérique non clé et ne doit pas appartenir à d'autres relations. Si cet attribut appartient à une autre relation, alors cette dernière doit être parallèle à la relation du fait.

## 4.2 Contraintes de conformité liées au concept dimension

Une fois les faits sont extraits, la deuxième étape consiste à extraire les dimensions. Dans le cas où le schéma de la source est relationnel, (Feki et al, 2007) proposent les heuristiques suivantes :

- Toute relation-entité R directement référencée par une relation-fait F est une dimension candidate pour F. Le nom de cette dimension est celui de R, son identifiant est le transformé de la clé primaire de R.
- Tout attribut booléen appartenant à une relation-fait F donne naissance à une dimension candidate pour F dont il devient l'identifiant.
- Tout attribut temporel (date ou temps) appartenant à une relation-fait F estampille le fait F et construit alors une dimension temporelle dont il devient l'identifiant.
- Si un attribut de la clé primaire d'une relation-fait F de classe R-a n'est pas une clé étrangère, alors cet attribut construit une dimension candidate dont il est l'identifiant.
- Tout attribut A non clé (primaire ou étrangère) appartenant à une relation-fait F et à d'autre(s) relation(s) est un identifiant candidat d'une dimension de F construite sur A.

Ainsi, une dimension est conforme par rapport à une source si elle respecte l'une des contraintes suivantes :

- Le nom de la dimension correspond au nom d'une relation, et son identifiant est à la fois une clé primaire de cette relation et une clé étrangère pour la relation fait.
- L'identifiant de la dimension est un attribut non clé (primaire ou étrangère) de la relation fait appartenant à d'autre relation.
- L'identifiant de la dimension correspond à un attribut booléen, temporel ou à une clé primaire et non étrangère de la relation fait.



### 4.3 Contraintes de conformité liées au concept hiérarchie

Une hiérarchie constitue une perspective d'analyse au sein d'une dimension. Son rôle est d'ordonner les attributs (paramètres et attributs faibles) d'une dimension. La conformité des paramètres est validée, si pour chaque paramètre extrait à partir d'un attribut, l'attribut remplit l'une des conditions suivantes :

- Il est une clé primaire d'une relation directement référencée par la relation dimension.
- Il est un attribut temporel ou booléen non clé.
- Il est un attribut non clé de la relation dimension et appartenant à une autre relation.

En ce qui concerne les attributs faibles : Un attribut désigné comme faible est valide s'il est textuel (ou numérique) non clé appartenant à une relation dont la clé est un paramètre et n'appartenant pas à d'autres relations.

Ainsi, une hiérarchie est valide si elle respecte les deux contraintes suivantes :

- Chaque paramètre doit correspondre à une clé étrangère de la relation dimension, un attribut booléen ou temporel non clé, ou bien, non clé et appartenant à une autre relation.
- Chaque attribut faible est un attribut non clé et appartenant uniquement à la relation dont sa clé primaire correspond au paramètre auquel il est associé.

## 5 Formalisation des contraintes

Dans cette section, nous formalisons en Z les contraintes de validation. Ces contraintes sont les contraintes de conformité et les contraintes sémantiques. Ces dernières sont dégagées des travaux de modélisation multidimensionnelle intégrant des contraintes :

- Contrainte de dépendance hiérarchique (*Soundness*) (Ghozzi, 2004) : Pour chaque couple de paramètres de la Hiérarchie il existe deux valeurs appartenant à ce couple tel que ces deux valeurs sont dépendantes.
- Contrainte d'orthogonalité (Abello et al, 2006) : cette contrainte vérifie l'absence de dépendance fonctionnelle entre les différentes dimensions. C'est-à-dire l'absence de dépendances entre les attributs d'une dimension et les attributs des autres dimensions. Ce qui est garanti par l'analyse des instances.
- Contrainte de relation Attribut faible-Paramètre (Ghozzi, 2004) : cette contrainte exige qu'un attribut faible ne puisse être connecté qu'à un seul paramètre.

Nous commençons notre spécification par décrire formellement en Z le concept Relation. Une relation est caractérisée par un ensemble de clés primaires, un ensemble de clés étrangères et un ensemble d'attributs.

<i>Relation</i>
<i>N: Name</i>
<i>P_Key: F ATTRIBUT</i>
<i>F_Key: F ATTRIBUT</i>
<i>Attribut: F ATTRIBUT</i>

## Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

En plus, nous définissons la relation transitive *path* mettant en relief les dépendances fonctionnelles entre les différents attributs

$$\frac{\text{path: } \text{ATTRIBUT} \leftrightarrow \text{ATTRIBUT}}{\forall x, y: \text{ATTRIBUT} \cdot \exists z: \text{ATTRIBUT} \mid (x, z) \in \text{path} \wedge (z, y) \in \text{path} \cdot (x, y) \in \text{path}}$$

Nous formalisons dans le paragraphe suivant ces contraintes en langage Z.

### 5.1 Formalisation des contraintes de validation du fait

Le schéma *SoundnessFact* décrit les contraintes de validation d'un fait par rapport au schéma de la source. Dans la partie déclarative du schéma, nous définissons *F* le fait à valider et *Relational* (ensemble fini de relation) le schéma de la source. Dans sa partie prédicat, nous exprimons les contraintes suivantes :

- Contrainte de conformité du fait par rapport à la source (1),
- Contrainte de conformité des mesures par rapport à la source (2)
- Contrainte d'orthogonalité exprimant l'indépendance fonctionnelle entre les différentes dimensions associées au fait (3).

$$\begin{array}{l} \text{SoundnessFact} \\ \hline F: \text{Fact} \\ \text{Relational}: F \text{ Relation} \\ \hline (1) \exists R: \text{Relational}; a: F. \text{Mesure} \\ \quad \cdot F.N = R.N \wedge a \in R. \text{Attribut} \wedge a.type = \text{Number} \\ (2) \forall m: F. \text{Mesure}; R_i: \text{Relational} \mid R_i.N = F.N \\ \quad \cdot \forall R_j: \text{Relational} \setminus \{R_i\} \mid \neg R_i.P\_Key \subseteq R_j.P\_Key \cdot m \notin R_j. \text{Attribut} \\ (3) \forall D_i: F. \text{DimAssocie} \\ \quad \cdot \forall D_j: F. \text{DimAssocie}; m: D_i. \text{Att} \mid D_j \neq D_i \wedge m.weight = ID \\ \quad \cdot \forall n: D_j. \text{Att} \mid n.weight \neq All \cdot (m, n) \notin \text{path} \end{array}$$

### 5.2 Formalisation des contraintes de validation de dimension

$$\begin{array}{l} \text{SoundnessDimension} \\ \hline F: \text{Fact} \\ \text{Relational}: F \text{ Relation} \\ \hline \forall R_f: \text{Relational}; D: F. \text{DimAssocie} \mid R_f.N = F.N \\ \quad \cdot \forall m: D. \text{Att} \mid m.weight = ID \\ \quad \cdot (\exists R_d: \text{Relational} \setminus \{R_f\} \\ \quad \quad \cdot R_d.N = D.N \wedge m \in R_d.P\_Key \wedge m \in R_f.F\_Key \\ \quad \quad \vee m \in R_f. \text{Attribut} \wedge m \in R_d. \text{Attribut}) \\ \quad \vee m \in R_f. \text{Attribut} \wedge (m.type = \text{Temporal} \vee m.type = \text{Boolean}) \\ \quad \vee m \in R_f.P\_Key \wedge m \notin R_f.F\_Key \end{array}$$

En gardant toujours les mêmes déclarations au niveau de la partie déclarative schéma box, la formalisation des contraintes de validation de dimension se traduit par

*SoundnessDimension*. Dans la partie prédicat, nous décrivons la contrainte de conformité de dimension.

### 5.3 Formalisation des contraintes de validation d'une hiérarchie

Le schéma box *SoundnessHierarchy* exprime la formalisation en Z des contraintes de validation d'une hiérarchie par rapport au schéma de la source. Dans la partie déclarative, nous déclarons *D* la dimension contenant la hiérarchie à valider et *Relational* la variable regroupant les relations du schéma de la source. Dans la partie prédicat, nous définissons les contraintes suivantes :

- La contrainte de dépendance hiérarchique (1).
- La contrainte de conformité des paramètres d'une hiérarchie (2).
- La contrainte de conformité des attributs faibles (3).

<i>SoundnessHierarchy</i>	
<i>D</i> : Dimension	
<i>Relational</i> : F Relation	
(1)	$\forall H: D. Hier$ <ul style="list-style-type: none"> <li>• <math>\forall a: H. Att \mid a. weighth = Parametre</math></li> <li>• <math>\exists R1, R2: Relational \mid R1 \neq R2</math></li> <li>• <math>a \in R1. P\_Key</math></li> <li>• <math>\forall a \in R1. Attribut</math></li> <li>• <math>\wedge (a. type = Temporal</math></li> <li>• <math>\vee a. type = Boolean</math></li> <li>• <math>\vee a \in R2. Attribut</math></li> <li>• <math>\vee a \in R2. P\_Key)</math></li> </ul>
(2)	$\forall H: D. Hier$ <ul style="list-style-type: none"> <li>• <math>\forall a, b: H. Att; i: \mathbb{N} \mid a. weighth \neq All \wedge b. weighth \neq All</math></li> <li>• <math>(i, a) \in H. ParamH \wedge (i + 1, b) \in H. ParamH \Rightarrow (a, b) \in path</math></li> </ul>
(3)	$\forall H: D. Hier$ <ul style="list-style-type: none"> <li>• <math>\forall a, b: H. Att</math></li> <li>• <math>(a. weighth = ID \vee a. weighth = Parametre)</math></li> <li>• <math>\wedge b. weighth = Weak</math></li> <li>• <math>\wedge (a, b) \in H. WeakH \cdot (a, b) \in path</math></li> </ul>

### 5.4 Validation de la spécification

La validation d'un schéma Z consiste en la démonstration que les types de données sont correctement définis et que les contraintes (prédicats) ne sont pas contradictoires. La première est vérifiée par l'outil Z/Eves par de simple click. La deuxième consiste en la démonstration du théorème d'initialisation. Il suffit d'instancier un exemple respectant les contraintes décrites au niveau des schémas Z et de vérifier que ce dernier respecte ces contraintes. Cette phase se compose des étapes suivantes :

- Dans une première étape, nous déclarons les différents objets constituant le schéma multidimensionnel et le schéma de la source en utilisant les axiomes-box.
- Dans la deuxième étape, nous définissons le schéma-box *SchémaInstance* qui jouera le rôle d'une instance de schéma Z déjà défini. Par ailleurs, nous devons associer les différentes valeurs nécessaires aux différents ensembles déclarés au niveau du schéma de validation.
- La dernière étape consiste en la démonstration du théorème d'initialisation.

## Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

Dans cette section, nous validons notre spécification par l'instanciation de l'exemple illustré dans les travaux de (Feki et al, 2007).

La figure (Figure 1) illustre le schéma en étoile du fait *Education\_Assure* selon les dimensions *Teacher* et *Field*. *Nbr\_Gr\_Tea* correspond à la mesure du fait. La dimension *Teacher* est caractérisée par son identifiant *Num\_Tea*, le paramètre *Grad\_Tea* et les attributs faibles *Name\_Tea* et *Fname\_Tea*. La dimension *Field* est caractérisée par son identifiant *Cod\_Fiel*, deux paramètres *Cod\_Sec* et *Cod\_Aud* et la les attributs faibles *Int\_Fiel*, *Vol\_Hou\_Fiel*, *Int\_Aud* et *Int\_Sec*.

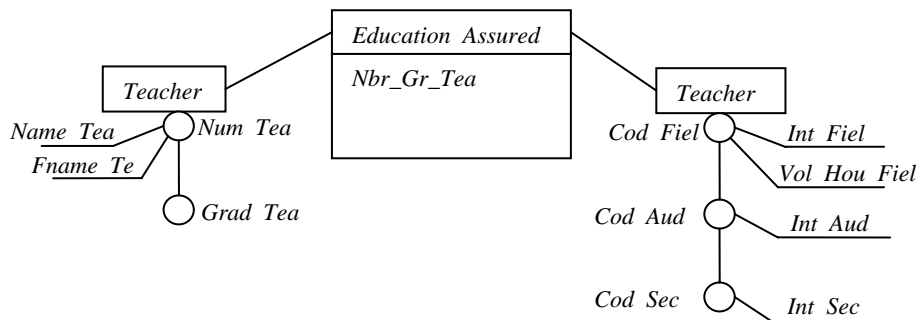


FIG. 1 – Schéma du fait *Education\_Assure*

Nous validerons ce schéma multidimensionnel par rapport au schéma de sa source. Ce dernier est un schéma relationnel en 3ème forme normale :

SECTION (Cod\_Sec, Int\_Sec)

AUDIENCE (Cod\_Aud, Int\_Aud, Cod\_Sec : SECTION)

FIELD (Cod\_Fiel, Cod\_Aud : AUDIENCE, Int\_Fiel, Vol\_Hou\_Fiel)

TEACHER (Num\_Tea, Name\_Tea, Fname\_Tea, Grad\_Tea)

EDUCATION\_ASSURED (Cod\_Fiel : FIELD, Num\_Tae : TEACHER, Num\_Sem, Yea\_Univ, Nrb\_Grp\_Tea)

CHAGE\_REQUIRED (Grad\_Tea, Charg\_Hou\_Req)

Commençons par vérifier si l'exemple traité satisfait les conditions nécessaires pour la validation de la spécification. Premièrement, nous remarquons que notre fait respecte la contrainte de conformité de fait par l'existence d'une relation portant le même nom « EDUCATION\_ASSURED » contenant un attribut numérique non clé. Deuxièmement, la mesure du fait respecte la contrainte de conformité des mesures. Troisièmement, les dimensions associées au fait sont indépendantes fonctionnellement, d'où la contrainte d'orthogonalité est respectée. Passant à la vérification des dimensions, nous constatons que les dimensions « Teacher » et « Field » respectent la contrainte de conformité de dimension. Ceci est garanti par l'existence de deux relations portant les mêmes noms que ces dimensions et leurs identifiants se sont des clés primaires pour ces dimensions et des clés étrangères pour la relation « EDUCATION\_ASSURED ». Nous terminons par les hiérarchies dont les paramètres et les attributs faibles respectent la contrainte de conformité de la hiérarchie.

Pour Commencer la procédure de validation, nous déclarons les différentes entités associées au schéma multidimensionnel ainsi que celles associées au schéma relationnel. La figure 2 illustre la déclaration des entités associées au schéma relationnel. Cette déclaration doit être précéder d'une autre pareille pour les entités associées au schéma multidimensionnel.

```

R_Education_Assured, R_Section, R_Audience, R_Field, R_Teacher,
R_Charge_Required: Relation
-----
R_Education_Assured . N = EDUCATION_ASSURED
R_Education_Assured . P_Key = {Cod_Fiel, Num_Tea, Num_Sem, Tea_Univ}
R_Education_Assured . F_Key = {Cod_Fiel, Num_Tea}
R_Education_Assured . Attribut = {Nbr_Gr_Tea}
R_Section . N = SECTION
R_Section . P_Key = {Cod_Sec}
R_Section . F_Key = {}
R_Section . Attribut = {Int_Sec}
R_Audience . N = AUDIENCE
R_Audience . P_Key = {Cod_Aud}
R_Audience . F_Key = {Cod_Sec}
R_Audience . Attribut = {Int_Aud}
R_Field . N = FIELD
R_Field . P_Key = {Cod_Fiel}
R_Field . F_Key = {Cod_Aud}
R_Field . Attribut = {Int_Fiel, Vol_Hou_Fiel}
R_Teacher . N = TEACHER
R_Teacher . P_Key = {Num_Tea}
R_Teacher . F_Key = {}
R_Teacher . Attribut = {Name_Tea, Fname_Tea, Grad_Tea}
R_Charge_Required . N = CHARGE_REQUIRED
R_Charge_Required . P_Key = {Grad_Tea}
R_Charge_Required . F_Key = {}
R_Charge_Required . Attribut = {Charg_Hou_Req}
path
= {(Cod_Fiel, Cod_Aud), (Cod_Fiel, Int_Fiel), (Cod_Fiel, Vol_Hou_Fiel),
  (Num_Tea, Name_Tea), (Num_Tea, Fname_Tea), (Num_Tea, Grad_Tea),
  (Cod_Sec, Int_Sec), (Cod_Aud, Cod_Sec), (Cod_Aud, Int_Aud)}
Nbr_Gr_Tea . type = Number

```

FIG. 2 – Déclaration des objets du schéma relationnel

La phase suivante consiste en l'instanciation des schémas *Z SoundnessFact*, *SoundnessDimension* et *SoundnessHierarchy*. Concernant *SoundnessHierarchy*, il faut instancier ce schéma pour chaque dimension. Dans notre cas, nous avons besoin de deux instanciations : la première sert à valider les hiérarchies de la dimension *Teacher* et la deuxième permet de valider les hiérarchies de la dimension *Field*.

La figure (Figure 3) illustre l'instanciation du schéma *SoundnessFact*. Nous instancions de la même manière les schémas *SoundnessDimension* et *SoundnessHierarchy*. Nous devons vérifier les hiérarchies de chaque dimension à part.

```

InstanceEducationAssured
-----
SoundnessFact
-----
F = FactEducation_Assured
Relational
= {R_Education_Assured, R_Section, R_Audience, R_Field, R_Teacher,
  R_Charge_Required}

```

FIG. 3 – Instance du schéma *SoundnessFact*

## Validation Formelle de Schéma Multidimensionnel vis à vis de sa Source

Enfin, nous devons démontrer le théorème d'initialisation associé à chaque instance. Ce théorème est de la forme suivante :

$$\mathbf{theorem\ ValidationSchemaInstance}$$
$$\exists Schema \cdot SchemaInstance$$

Par la démonstration de ce théorème (Figure 4), nous garantissons la consistance de notre spécification. En outre, les contraintes exprimées au niveau d'un schéma ne sont pas contradictoires.

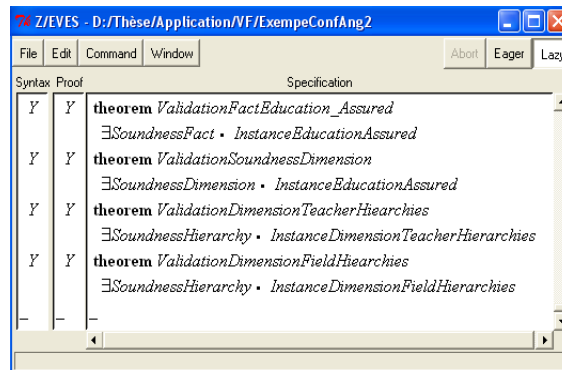


FIG. 4 – Preuve des différents théorèmes d'initialisation par l'outil Z/Eve

## 6 Conclusion

Dans cet article nous avons proposé un cadre formel de validation de schémas multidimensionnels. Il réalise le contrôle de la conformité du schéma multidimensionnel par rapport à celui de sa source. Notre approche offre un moyen rigoureux de validation de modèles multidimensionnels. Un cadre formel facilitant la lisibilité et le raisonnement sur un tel modèle. En outre, pour tester la conformité des modèles par rapport à la source, selon notre approche, il n'est plus nécessaire de confronter le schéma résultant de la démarche ascendante et celui basée sur les besoins des décideurs. Le problème se réduit à vérifier certaines contraintes que nous avons nommé contraintes de conformité. Nous sommes entrain de mettre en place une démarche de validation automatique similaire à celle proposée dans (Salem et al, 2008) afin d'offrir au concepteur non expert en méthode formelle un moyen de valider leurs schémas multidimensionnels.

## Références

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

- Cavero J., Piattini M., Marcos E., "MIDEA : A multidimensional Data Warehouse Methodology". Dans 3rd international Conference on Enterprise Information Systems (ICEIS'01), Setubal, Portugal, p. 138-144, juillet 2001.
- Carneiro L., Brayner A., "X-META : A Methodology for Data Warehouse Design with Metadata Management". Dans 4th International Workshop on Design and Management of Data Warehouses (DMDW'02), Toronto, Canada, p. 13-22, mai 2002.
- 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.
- Hui J., Dong L. and Xiren X., "Using formal specification language in industrial software development". In Proceedings of the IEEE International Conference on Intelligent Processing Systems, pages 1847-1851, Beijing, China, October 1997.
- 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.
- Ghozzi F., "Conception et Manipulation de Base de Données Dimensionnelles à Contraintes". Thèse de l'Université Paul Sabatier - Toulouse III, novembre 2004.
- Golfarelli, M., Maio, D., and Rizzi, S., "Conceptual Design of Data Warehouses from E/R Schemas". Conference on System Sciences, Vol. VII, Kona, Hawaii. 1998.
- Lechtenböcker J., Vossen G. "Multidimensional normal forms for data warehouse design". Dans Revue Information Systems, Vol. 28, N. 5, p. 415-434, juillet 2003.
- 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.
- Mazón J. N., Trujillo J., Lechtenböcker J., "Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms". Data Knowl. Eng. 63(3): 725-751, 2007
- Moody, L.D., and Kortink, M.A.R.. "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouses and Data Mart Design." Proc. of the Int'l Workshop on Design and Management of Data Warehouses, Stockholm, Sweden. 2000.
- Pang, C., Taylor, K., Zhang, X. et Cameron, M. "Generating Multidimensional Schemata from Relational Aggregation Queries". LNCS 3306, pp. 584-589, 2004.
- Romero O., Calvanese D., Abelló A., Rodríguez-Muro M., "Discovering functional dependencies for multidimensional design". DOLAP, 1-8, 2009
- Spivey J.M.. "The Z Notation : a Reference Manual". Prentice-Hall, 1992.
- Saaltink M.. "The Z/EVES 2.0 User's Guide". ORA Canada, One Nicholas Street, Suite 1208, Ottawa (Ontario), K1N 7B7, octobre 1999.
- Salem A., Ghozzi F., Ben Abdallah H., Zurfluh G., "Spécification Formelle du Modèle Multidimensionnel à Contraintes", Atelier Systèmes d'Information Décisionnels, INFORSID 2006, Hammamet, Tunisie, p. 22-27 31 Mai 2006.
- Salem A., Ghozzi F., Ben-Abdallah H. "Modélisation multidimensionnelle : vérification formelle de schéma de hiérarchie", 9ème Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées (CARI'08), Rabat, Maroc, p727-734, 27 octobre 2008.

