Extending the Multidimensional Model for Linking Cubes

Alberto Sabaini*, Esteban Zimányi**, Carlo Combi*

*Department of Computer Science,
University of Verona, Italy
{alberto.sabaini, carlo.combi}@univr.it
**Department of Computer and Decision Engineering (CoDE),
Université Libre de Bruxelles, Belgium
ezimanyi@ulb.ac.be

Abstract. Data warehouses structure data in multidimensional cubes, where dimensions specify different ways in which measures in facts may be viewed, aggregated, and sorted. It is essential for data analysts to combine data from heterogeneous multidimensional cubes to enhance their analysis capabilities. For this, users are restricted in using only shared dimensions for navigating related multidimensional cubes. In this paper, we show that this limits the analysis possibilities and introduce an explicit link that relates two multidimensional cubes, indicating that they represent different aspects of the same reality, and hence they may be connected. We argue that the standard drill-across operator is not suited to perform such operation, and we extend it by proposing a new operator called drill-across-link.

1 Introduction

Data warehouses structure data according to a multidimensional space, where dimensions specify different ways the data can be viewed, aggregated, and sorted. Events of interest for an analyst are represented as facts which are associated with cells or points in this multidimensional space and described in terms of a set of measures. Thus, every fact is based on a set of dimensions that determine the granularity adopted for representing fact measures. Dimensions are organized as hierarchies of levels that allow analysts to aggregate data at different degrees of detail. One of the major challenges that must be faced by designers of multidimensional models is to adequately represent the interactions between dimensions and facts (Song et al., 2001). Another issue is to represent the connections between different facts in the same schema. Indeed, data analysts often need to combine data from heterogeneous multidimensional cubes. A key requirement in navigating multidimensional cubes from several sources, according to Kimball and Ross (2013), is that they must have *shared* dimensions. Their widely adopted view requires shared dimensions to be either from the same instance or identical in terms of schema and data. The non-conformity problem arises when there is the need of combining multidimensional cubes and using non-shared dimension. The inclusion of non-shared dimensions in the navigation and visualization of multidimensional data from multiple sources provides the analyst with the ability to view and analyze data that would be otherwise not

available. Unfortunately, as Kimball and Ross (2013) point out, all these related data cannot be combined and put in the same cube, since separate cubes do not share all the dimensions.

In this paper, we show that this limits the analysis possibilities and introduce an explicit link that relates two multidimensional cubes, indicating that they represent different aspects of the same reality, and hence they may be connected. We argue that the standard drill-across operator is not suited to perform such operation, and we extend it by proposing a new operator called drill-across-link.

This paper is organized as follows. Section 2 summarizes some relevant related work in the area of multidimensional modeling and data aggregation. Section 3 introduces a real scenario in the medical field, which is used throughout the paper to emphasize the need of a connection between multidimensional cubes. We discuss the concept of a link between two cubes in further detail in Section 4. The multidimensional model we consider throughout the paper is then presented in Section 5. Section 6 is dedicated to the definition of a new version of the drill-across operator, that takes into account links between two cubes. Sections 7 addresses the issue of the double-counting in the context of the new operator. Finally, Section 8 draws some conclusions and addresses future work.

2 Related Work

In this section we summarize some relevant work in the area of multidimensional models and Online Analytical Processing (OLAP) operators.

Pedersen et al. (2001) use a medical case study for patient diagnosis to demonstrate the analysis requirements not supported by traditional multidimensional models. The proposed extensions aim at supporting non-summarizable hierarchies, symmetric treatment of dimensions and measures, and correct aggregation over imprecise or incomplete data. Jensen et al. (2004) present the guidelines for designing complex dimensions in the context of spatial data such as mobile, location-based services. Mansmann and Scholl (2007) analyzed the limitations of multidimensional models in handling complex dimension hierarchies and proposed extensions at the conceptual level and their relational mapping as well their implementation in a prototype front-end tool. A comprehensive classification of dimensional hierarchies, including those not addressed by current OLAP systems, formalized at both the conceptual model and the logical level, may be found in Malinowski and Zimányi (2006).

Kimball and Ross (2013) define the drill-across operation as the process of linking two or more fact tables at the same granularity, or in other words, tables with the same set of grouping columns and dimensional constraints. When multiple fact tables are tied to a dimension table, the fact tables should all link to that dimension table. When the same dimension table is being used with each of the fact tables, the dimension is "shared" to each fact table. Dimensions that are not shared (such as those that differ in grain or detail) across fact tables will prevent the application of the drill-across operation. In this paper, we point out that the drill-across operator needs to be extended in order to allow users to combine different cubes.

Another important aspect that must be taken care of is the correct aggregation of data. When merging data from different sources, the summarizability property must be ensured because, otherwise, its violation can lead to incorrect results, and therefore erroneous analysis decisions (Lenz and Shoshani, 1997; Lehner et al., 1998). The notion of summarizability was introduced by Rafanelli and Shoshani (1990) in the context of statistical databases, where it

refers to the correct computation of aggregate values with a coarser level of detail from aggregate values with a finer level of detail. Rafanelli and Shoshani (1990) observe that many-to-one associations satisfy summarizability while many-to-many associations violate it.

Abelló et al. (2002) focus their attention on the navigation among different facts. They study different kinds of object-oriented conceptual relationships between facts (namely *Derivation, Generalization, Association,* and *Flow*) that allow to drill across them. They study different kinds of semantic relationships between facts or dimensions. In particular, they point out that the drill-across operator allows users to jump from one cube to another. The closest kind of relationship to the one we are presenting in this paper, is the association between two facts. Shared dimensions are not considered in this kind of relationship. The drill-across operation is possible due to the fact that there exists an *Association* between facts. It is not necessary the dimensions in the destination fact to be related to those in the origin. It could be that selected cells in the latter determine a set of cells in the former. Thus, they just substitute *Measures* of one cell, by those of its counterpart in the other fact. As we will present in the following, our approach is different from theirs in the sense that we allow the analysis of "foreign" measures (i.e., from the destination cube), according to the origin dimensions.

Riazati et al. (2008) make a distinction between conformed dimension tables and conformed dimension attributes and discuss the advantages of relaxing the conformity requirement. The authors identify conformity within the dimension attributes, and describe methods to measure the loss resulting from the join between conformed dimension attributes with dissimilar values. They extend the definition of the drill-across operation to include (selective) non-conformed dimension attributes in the analysis. They provide users a way for analyzing multiple cubes with non-conformed dimension attributes.

Torlone (2008) propose two different approaches to the problem of integration of different data sources. The first approach refers to a scenario of loosely coupled integration, in which they identify the common information between data sources and perform join operations over the original sources. The second approach deals with the derivation of a materialized view built by merging the sources, and refers to a scenario of tightly coupled integration in which queries are performed against the view. They create a merged dimension in which they identify the common part, in order to allow users to perform drill-across operations.

3 Scenario

To illustrate the need of a link between multidimensional cubes, and to make it easier to understand the general concepts defined in the reminder of this paper, we use as example a real scenario in the pharmacovigilance domain, which is the activity related to the collection, analysis, and prevention of Adverse Drug Reactions (ADRs) induced by drugs. Due to the limitations of pre-marketing trials, such as a limited duration and a highly selected test population, often unexpected adverse reactions go undetected and only become apparent when the drug reaches the general population. For this reason, it is necessary to control drugs after their release on the market. Spontaneous reporting of suspected cases allows users (i.e., physicians, pharmacits, and citizens) to identify and send reports about unexpected reactions induced by drugs administration to the regulatory authority. This practice is invaluable, provides early warnings, and requires limited economic and organizational resources. It also has the advantage of covering all drugs on the market and of including all categories of patients. In this

scenario, we are going to use standard medical classifications for both drugs and adverse reactions. For the first one, we use the Anatomical Therapeutic Chemical (ATC) Classification System, which classifies drugs according to the organ or system on which they act and their therapeutic, pharmacological and chemical properties. For the latter one, we use the Medical Dictionary for Regulatory Activities (MedDRA) which is a clinically validated international medical terminology dictionary (and thesaurus) used by regulatory authorities for the classification of adverse reactions.

From an analysis perspective, we may identify two main facts of interest in this scenario, namely, Treatment and suspected AdverseReaction. A treatment is characterized by the patient, the administered drug, the cost of the therapy, the daily dosage, and the treatment period. An example of such a fact is depicted by the Treatment relation shown in Table 1. An adverse reaction is characterized by the reaction, the patient, and the period of illness. An

TreatmentKey	Patient	Drug	StartDate	EndDate	Cost	DailyDosage
T1	Alice	Tylenol	16-08-2013	07-09-2013	65	40
T2	Alice	Tylenol	10-09-2013	14-09-2013	20	20
T3	Alice	Aspirin	17-09-2013	08-10-2013	60	30
T4	Bob	Aspirin	01-09-2013	28-09-2013	80	30
T5	Bob	Tylenol	04-09-2013	19-09-2013	60	30
T6	Charlie	Tylenol	20-08-2013	28-08-2013	30	40
T7	Charlie	Aspirin	13-09-2013	20-09-2013	35	50
T8	Charlie	Tylenol	22-09-2013	12-10-2013	70	20

TAB. 1 – *Example of the Treatment relation contained in the Pharmacovigilance Database*

example of such a fact is depicted by the AdverseReaction relation shown in Table 2. Figure 1 shows a multidimensional schema representing these facts. The *MultiDim* model (Vaisman and Zimányi, 2014) has been used for representing the schema.

A multidimensional schema is composed of a set of dimensions and a set of facts. A dimension is composed of either one level or one or more hierarchies. A hierarchy is composed of a set of levels related by roll-up relationships.

A level describes a set of real-world concepts that, from the application perspective, have similar characteristics. A level has a set of attributes that describe the characteristics of their members. It also has one or several identifiers that uniquely identify the members of a level.

AdvReactionKey	Patient	Reaction	StartDate	EndDate	Severity
A1	Alice	Hepatitis	03-09-2013	11-09-2013	6
A2	Alice	Urticaria	12-09-2013	06-10-2013	1
A3	Bob	Hepatitis	06-09-2013	08-10-2013	4
A4	Bob	Urticaria	12-09-2013	22-09-2013	5
A5	Charlie	Urticaria	22-08-2013	06-09-2013	9
A6	Charlie	Hepatitis	15-09-2013	02-10-2013	7

TAB. 2 – *Example of the AdverseReaction relation contained in the Pharmacovigilance Database*

They may be composed of one or several attributes. In Figure 1, DrugKey is an identifier of the Drug level, while DrugName is the attribute for the drug commercial name.

A fact relates levels from several dimensions. For example, the Treatment fact in Figure 1 relates the Patient, Drug, and Time levels (ignore for the moment the gray connection between the Treatment and AdverseReaction facts). The same level can participate several times in a fact by playing different roles. Each role is identified by a name and is represented by a separate link between the corresponding level and the fact. For example, in Figure 1, the Time level participates in the Treatment fact with the roles StartDate, and EndDate. Measures contain data (usually numerical) that are analyzed using the various perspectives represented by the dimensions. The Treatment fact has two measures, namely Cost and DailyDosage.

Similarly, the AdverseReaction fact in Figure 1 relates the Patient, LowestLevelTerm, and Time levels. The latter one participates several times in the fact with different roles, like in the Treatment fact. The AdverseReaction fact has one measure called Severity (which is the intensity of the reaction from a scale from 1 to 10).

Multidimensional structures may be queried by using On-line Analytical Processing (OLAP) tools to retrieve information such as:

1. *Show the total cost for treatments started in 2013 for* Aspirin *administrations*, or
2. *Show the total number of skin-related reactions*.

In the first query drugs would be filtered and only cases of Aspirin will be kept. Then, the sum of the cost of the remaining cells would be performed. Given the data in Table 1 the result would be $(\text{Aspirin} = 175)$. In the second query, only the skin-related reactions will be kept. In our example there is only an Urticaria case. Hence, given the data in Table 2, the result would be $(\text{Urticaria} = 3)$.

4 Linking Cubes

Let us consider the following query: *What is the maximum daily dosage for drugs suspected to have induced a skin disorder reaction?* As we already mentioned, the goal of pharmacovigilance is to assess the suspected cases of adverse reactions induced by drug administrations, so it makes sense for the user to consider at the same time the occurrences of treatments and adverse reactions. The usual way to combine multidimensional cubes is through the drill-across operation, which performs a join on their shared dimensions. The drill-across operator, as defined by Kimball and Ross (2013), comes with the restriction that the two cubes must have shared dimensions in order to be combined. Shared dimensions may be identical, where the dimensions have consistent keys, attribute names, and values, or shrunken, where a dimension contains a subset of attributes of the other one. Only the shared dimensions may be used while combining two cubes. In this case, the above query cannot be expressed because it refers to the Drug dimension of the Treatment fact, and to the LowestLevelTerm dimension of the AdverseReaction fact.

By means of the drill-across operator, users are in fact exploiting an implicit connection between two cubes, which is represented by the shared dimensions. It is represented by the presence of one or more shared dimensions. We argue that an explicit connection between cubes is needed in many real-word situations. Instead of relying on common members of shared dimensions, we could rely on explicit links between the instances of two facts. For this,

a new version of the drill-across operator is needed. As we explain in the following, it takes into account the explicit link between the fact tables. The cardinality of the link can be 1-to-1, 1-to-m, or m-to-m. This cardinality must be taken into account for ensuring summarizability in the aggregation process, as we discuss in Section 7.

Let us assume that a particular definition of the drill-across operator would allow us to use all dimensions from the two source cubes. If only the shared dimensions were to be used for performing this query, the result might be incorrect and lead a potential user to erroneous conclusions. Indeed, the standard drill-across operation merges the facts from the first cube to all the facts in the second one that have the same values in the shared dimensions. This is equivalent to an equi-join. However, some of the mentioned facts from the first cube may not be related to the facts in the second one. In our running example, drug administrations referring to a patient may only relate to some of the adverse reactions experienced by the said patient, e.g., the physician ruled out one drug because he did not suspect it as cause of the reaction. Table 3 shows an example of connections between treatments and adverse reactions. T1 is only related to A1, while T2 and T3 are both related to A2 and A3. All these 5 facts refers to the patient named Alice. The same happens with the patient named Bob: T4 is only related to

TreatmentKey	**AdvReactionKey**
T1	A1
T2	A1
T2	A2
T3	A1
T3	A2
T4	A3
T5	A4
T6	A5
T7	A6
T8	A6

TAB. 3 – *Many-to-many relationship between the Treatment and AdverseReaction facts represented by a bridge table*

A3, and not to A4. By using the standard drill-across operator, some drug administrations may be tied up with some unrelated adverse reactions. Time dimensions may be indeed used for finding overlapping or adjacent periods in order to express this connection. However, temporal relationships would not be enough: treatment T1 may be temporally close to adverse reaction A2, but does not mean there exists an association between the two. To keep this information, the Treatment and AdverseReaction facts are connected by a link in the conceptual schema in Figure 1. The link, called Suspected, relates treatments that might have caused adverse reactions. The relationship between facts in a schema is represented with the same notation as the one used for the relationship between facts and levels. The cardinality of the relationship between facts indicates the minimum and the maximum number of members of one fact that can be related to members of the other fact. For example, the Treatment fact is related to the AdverseReaction fact with a many-to-many relationship, depicted in the gray area in Figure 1. Indeed, several treatments could be suspected to have caused a adverse reactions, indicating that a combination of some drugs could lead to several side effects.

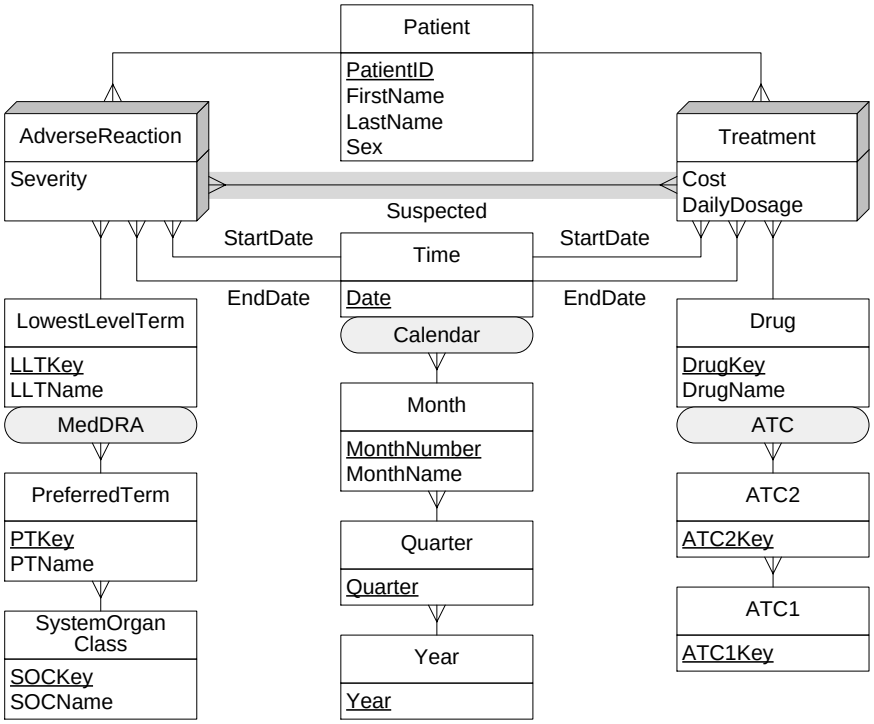

FIG. 1 – *A Multidimensional schema that represents events of drug administrations (i.e., the* Treatment *fact), and possibly related adverse reaction occurrences (i.e., the* AdverseReaction *fact). The relation between the* Treatment *and* AdverseReaction *facts is explicitly depicted by the* Suspected *link. MedDRA and ATC are two classification systems for adverse reactions and drugs, respectively*

We discuss next the mapping of a multidimensional schema to the relational model. For brevity, we only consider a star schema representation. Each dimension is mapped to a table of the same name, having as attributes the ones in all dimension levels. Likewise, each fact is translated into a table of the same name having as attributes a surrogate key, the foreign keys of the related dimension tables, and the measures. Figure 2 shows a relational implementation of the conceptual schema depicted in Figure 1. The fact tables are depicted in gray and the dimension ones in white. The Time dimension and its Calendar hierarchy are translated in the Time table, having as attributes TimeKey, Date, MonthNumber, MonthName, Quarter, and Year. The rows of the Time table represent the dimension level members. The Treatment fact depicted in Figure 1 is translated into the table of the same name having as attributes TreatmentKey, StartDateKey, EndDateKey, DrugKey, PatientKey, DailyDosage, and Cost.

In a star schema, the dimension tables are, in general, not normalized. Therefore, they may contain redundant data, especially in the presence of hierarchies. This is the case for dimension Drug since all drugs belonging to the same ATC2 class will have redundant information for the attribute describing the ATC1 class.

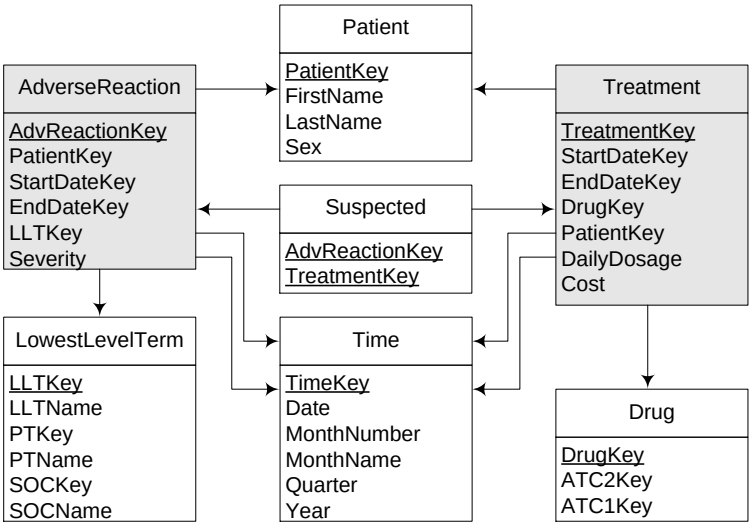

FIG. 2 – *Relational schema representation of the conceptual multidimensional schema in Figure 1*

Furthermore, in our running example we introduce a surrogate key that functionally determines all measures and foreign key attributes. The surrogate keys are used in the implementation of the link.

The mapping of links between facts depends on their cardinalities. In the case of one-to-one or one-to-many cardinalities, the surrogate key of the fact with a one cardinality is added as a foreign key of the other fact. In the case of many-to-many cardinalities, a bridge table with foreign keys to the two facts is needed. In our example, the many-to-many link between the AdverseReaction and Treatment facts is represented as a bridge table Suspected, whose content is shown in Table 3. On the other hand, consider the case where the Suspected link is one-to-many, i.e., a treatment may be suspected to have caused many adverse reactions, but a reaction may be induced by just one treatment. In this case, a bridge table would not be needed, but instead the TreatmentKey should be added to the AdverseReaction fact table.

5 The Multidimensional Model

In this section, we introduce a multidimensional data model, based on the notion of dimensions, measures, and facts. Each dimension is organized in a hierarchy of levels, which allows us summarize the measures in facts at different granularities. Within a dimension, values of a finer granularity can roll up to values of a coarser one. A multidimensional schema consists of facts defined with respect to a particular combination of levels. A multidimensional instance associates measures with dimension coordinates in each fact. For simplicity, and without loss of generality, we assume that dimensions names are unique. An example of a multidimensional schema is shown in Figure 1 by means of the *MultiDim* conceptual model (Vaisman and Zimányi, 2014).

Definition 1. *Multidimensional Schema*

- A multidimensional schema $\mathcal{S}$ is composed by a set of dimensions $\mathcal{D} = \{D_1, \ldots, D_m\}$ and a set of facts $\mathcal{F} = \{F_1, \ldots, F_n\}$.
- A dimension $D_i \in \mathcal{D}, i = 1, \ldots, m$ has a unique name and is composed of a set of levels $\mathcal{L} = \{L_1, \ldots, L_n, All\}$, a set of hierarchies $\mathcal{H} = \{H_1, \ldots, H_k\}$
- A level $L_i \in \mathcal{L}, i = 1, \ldots, n$, is defined by a schema $L_i(A_1 : dom_1, \ldots, A_n : dom_n)$, where L_i is the level name, and each attribute A_j is defined over the domain dom_j. The special level $All \in \mathcal{L}$, does not have any attribute. The level name L_i is unique in $\mathcal{L}$, and the attribute name A_j is unique in L_i.
- A hierarchy $H_i \in \mathcal{H}, i = 1, \ldots, n$ has a unique name and is composed of a subset $\mathcal{S}$ of the set $\mathcal{L}$ of levels of the dimension D, $\mathcal{S} \subseteq \mathcal{L} \in D$. It is also composed of a roll-up relation R consisting of a set of triples $\langle L_j, L_k, card\rangle$, where L_j and L_k are levels of the dimension D to which the hierarchy H_i belongs, and $card \in \{1\text{-}1, 1\text{-}m, m\text{-}m\}$ denotes the cardinality of the link between the child level L_j and the parent level L_k. The roll-up relation R, includes All as parent in some triple, and this level is, directly or transitively, accessible from all levels of the hierarchy.
- A fact $F_i \in \mathcal{F}, i = 1, \ldots, n$ is defined by a schema $F(K : dom, N_1 : \langle L_1, card\rangle, \ldots, N_m : \langle L_m, card\rangle, M_1 : dom_1, \ldots, M_n : dom_n)$ and has an attribute k that uniquely identifies a cell of the fact instance, i.e., it functionally determines all the roles N_i and M_j of each cell. N_i is a role name, L_j is the finest granularity on which measures are captured on dimension $D_j \in \mathcal{D}$, and *card* indicates the cardinality of the relationship between the dimension and the fact. It is one of $1\text{-}1, 1\text{-}m$, or $m\text{-}m$. Each measure M_i is defined over the domain dom_i. The role names N_i and the measures names M_j are unique in F.

Example 1. The schema depicted in Figure 1 can be defined as:

$\mathcal{D} = \{$Patient, Drug, Time, LowestLevelTerm$\}, \mathcal{F} = \{$AdverseReaction, Treatment$\}$

The schema for the Drug dimension is:

$\mathcal{L}_{\text{Drug}} = \{$Drug(DrugKey : Int, DrugName : String$), \ldots,$ All$\}$

$\mathcal{H}_{\text{Drug}} = \{\langle$ATC, {Drug, ATC2, ATC1}, $R_{\text{ATC}}\rangle\}$

$R_{\text{ATC}} = \{\langle$Drug, ATC2, 1-m$\rangle, \langle$ATC2, ATC1, 1-m$\rangle, \langle$ATC1, All, 1-m$\rangle\}$

The schema of the Treatment fact is described as:

Treatment = {Drug:$\langle$Drug, 1-m$\rangle$, StartDate:$\langle$Time, 1-m$\rangle, \ldots,$ Cost : Int, $\ldots\}$

Definition 2. *Multidimensional Instance.* A multidimensional instance is composed by dimension instances and fact instances. A dimension instance is composed by:

- A set $\mathcal{B}$ of members for each level L_i of each dimension $D_i \in \mathcal{D}$ in Def. 1. The level All has a unique member all.
- A finite set of relation R_i^j defined on the schema (k_i, k_j), where k_i identifies a member in level $L_i \in \mathcal{L}$. A couple $\langle k_i, k_j\rangle \in R_i^j$ if the member identified by k_i rolls up to k_j.

A fact instance is composed by a set C of cells, representing single events of a fact. A cell $c \in C$ is a tuple of levels $b_i \in \mathcal{B}$, measures values m_j, and an unique key k.

$$c = \langle k, b_1, \ldots, b_m, m_1, \ldots m_n\rangle$$

Example 2. The instance of the dimension Drug of the multidimensional schema depicted in Figure 1 can be defined as:

$\mathcal{B}_{\mathsf{Drug}} = \{(D1, Tylenol), (D2, Aspirin)\}$
$\mathcal{B}_{\mathsf{ATC2}} = \{(A1, N02), (A2, B01)\}$
$\mathcal{B}_{\mathsf{ATC1}} = \{(A1, N), (A2, B)\}$
$RL^{\mathsf{ATC2}}_{\mathsf{Drug}} = \{\langle D1, A1\rangle, \langle D2, A2\rangle\}$

The instances of the fact Treatment are in the form:
Treatment $= \{(T1, P1, D1, \text{16-08-2013}, \text{07-09-2013}, 65, 40), \ldots\}$ which corresponds to the first line of Table 1.

Definition 3. *Explicit Link* An explicit link between two facts F_1 and F_2 is a relation LK representing their connection. LK is composed by a set of couples of key attributes k_1, k_2 belonging to F_1 and F_2 respectively. Let us assume $k_1 \in K_1$ and $k_2 \in K_2$. If $\langle k_1, k_2\rangle \in LK$, then the two cells identified by those keys are connected. The cardinality of a link is determined by the uniques of each key value.

Example 3. Table 3 depicts the relation Suspected, which represent the explicit m-t-m link between the Treatmentand AdverseReaction facts. Keys from both facts are connected in order to represent an association between the two events. For instance, treatment T2 is connected with both A1 and A2 reactions.

6 The Drill-Across-Link Operator

The drill-across-link operator combines the cells from two data cubes that are connected by an explicit link. The syntax of the operator is as follows:

$$\mathsf{Cube} \leftarrow \mathsf{Drill\text{-}Across\text{-}Link}(\mathsf{Cube}_1, \mathsf{Cube}_2, \mathsf{Link})$$

that is, it performs a join of the two cubes, Cube_1 and Cube_2, across the specified Link.

Definition 4. *Drill-across-link operator.* Let C_1 and C_2 be two multidimensional instances, connected by a link LK. Then, $C = \mathsf{Drill\text{-}Across\text{-}Link}(C_1, C_2, LK)$ is defined as the combination of C_1 and C_2 across LK. The fact schema of the resulting instance C is in the form: $F(C) = F(C_1) \bowtie F(C_2)$. Intuitively, the final schema is the join of the all dimensions and measures from both the source facts. Indeed, by performing the join of all the dimensions, only one copy of the shared ones will be kept. Without loss of generality, we assume that the shared dimensions are the first k in order of appearance in both C_1 and C_2. The instance I of the resulting cube is in the form:

$$\begin{aligned} I = \quad & \{x | \exists y \in I_1 \exists z \in I_2 : y.Key = LK.C_1Key \wedge z.Key = LK.C_2Key \wedge \\ & \forall i\ 1 \ldots k : D_i(x) = D^1_i(y) = D^2_i(z) \wedge \forall i\ k \ldots m_1 : D_i(x) = D^1_i(y) \wedge \\ & \forall j\ k \ldots m_2 : D_{m_1+j}(x) = D^2_j \wedge \forall i\ 1 \ldots n_1 : M_i(x) = M^1_i(y) \wedge \\ & \forall j\ 1 \ldots n_2 : M_{n_1+j}(x) = M^2_j\} \end{aligned}$$

The first line of the instance definition represents the existence of two connected cells, belonging to their respective source cubes. The connection is expressed by the link that ties C_1 and C_2. The second row builds the dimensions shared by the two cubes (we assume that they also share the same values). After that, the non-shared dimensions and the measures are built. C_1 has m_1 dimensions and n_1 measures. Their values are copied in the dimensions and

TreatmentKey	AdvReactionKey	Patient	Drug	Cost	DailyDosage	Reaction
T1	A1	Alice	Tylenol	65	40	Hepatitis
T2	A1	Alice	Tylenol	20	20	Hepatitis
T2	A2	Alice	Tylenol	20	20	Urticaria
T3	A1	Alice	Aspirin	60	30	Hepatitis
T3	A2	Alice	Aspirin	60	30	Urticaria
T4	A3	Bob	Aspirin	80	30	Hepatitis
T5	A4	Bob	Tylenol	60	30	Urticaria
T6	A5	Charlie	Tylenol	30	40	Urticaria
T7	A6	Charlie	Aspirin	35	50	Hepatitis
T8	A6	Charlie	Tylenol	70	20	Hepatitis

TAB. 4 – *Result of the drill-across-link operator applied to the Treatment and AdverseReaction relations in Tables 1 and 2. The time dimensions have been omitted for space reasons*

measures of the resulting cube. Similarly, the non-shared dimensions and the measures of C_2 are copied in the resulting cube. Only one copy of the shared dimensions will be kept in the final result. This may be avoided by renaming the dimensions roles that should not be merged.

An example of application of this operator is: *Combine treatments suspected to have caused adverse reactions*.

$$\begin{aligned} \text{Treatment'} \leftarrow\ & \text{Rename}(\text{Treatment}, \text{StartDate} = \text{TreatStartDate}, \\ & \text{EndDate} = \text{TreatEndDate}) \\ \text{AdverseReaction'} \leftarrow\ & \text{Rename}(\text{AdverseReaction}, \text{StartDate} = \text{AdvRStartDate}, \\ & \text{EndDate} = \text{AdvREndDate}) \\ \text{AdvRTreatment} \leftarrow\ & \text{Drill-Across-Link}(\text{Treatment'}, \text{AdverseReaction'}, \text{Suspected}) \end{aligned}$$

The drill-across-link operator above returns a new cube, called AdvRTreatments, whose cells are the combination of the cells from the two cubes, according to the Suspected link. The result of the drill-across-link operator in our running example is depicted in Table 4. It is composed by the union of all dimensions from the source cubes. The natural join is performed on the dimensions, shared by the two cubes, that have the same role name. In the example depicted in Figure 1, the shared dimension Patient is naturally joined. This makes sense, since assume that the patient will be the same for both treatment and adverse reaction. On the other hand, the shared dimension Time plays different roles in both sources cubes, and they should be treated as separate dimension, and kept in the final result. Dimensions or their roles may be renamed before the application of the drill-across-link operator, to avoid them to be merged in the final result (as depicted in the above formula). Other conditions may be enforced after its application, such as $\text{AStartDate} > \text{TStartDate}$.

7 Measure Aggregation

In the presence of many-to-many relationships between cubes, the combination of their cells may introduce errors when computing aggregation. The same problem arises with traditional data warehouses, in the case of a many-to-many relationship between a cube and one

of its dimensions. Some of the cell's measures value may appear several times, due to the merging and the relationship type, leading to erroneous aggregation results.

Let us consider the following query: *Show the total cost and the max daily dosage of treatments per adverse reaction.* Intuitively, the Treatment and AdverseReaction cubes must be joined in order to answer this query. The standard drill-across operator would not be able to retrieve the requested information: by joining the two cubes over the shared dimensions, the adverse reaction dimension would not be available in the merged cube. The adverse reactions are needed for creating the aggregation groups. The Drill-Across-Link operator, on the other hand, could be used for answering such a query. Its application on the example data depicted in Tables 1 and 2 would lead to the data shown in Table 4. As the reader might notice, treatments T2 and T3 appear in the result. The same issue happens also for the adverse reactions: reactions A1, A2, and A6 appear multiple times. The max aggregation operator would not be affected by the data repetition. On the other hand, the application the sum aggregation operator would lead to erroneous results. For example, the cost of treatment T2 would appear in two aggregation groups, i.e., the A1 and A2 groups, which may induce the user to think that the T2 cost has been accounted twice, which is not the case.

This aggregation error is due to the many-to-many relationship among the source cubes. The *double counting* issue introduced above can be analyzed through the concept of multidimensional normal forms (MNFs) (Lechtenbörger and Vossen, 2003). MNFs determine the conditions that ensure correct measure aggregation, also called summarizability. The first multidimensional normal form (1MNF) requires each measure to be functionally determined by the set of associated leaf levels. In order to analyze the result of the Drill-Across-Link operator in terms of the 1MNF, we first need to find out functional dependencies that exist between the leaf levels and the measures. It can be easily seen that the measures of a treatment only depend on patient, drug, and administration time. Thus, the resulting cube does not satisfy the 1MNF, since the measures are not functionally determined by all leaf levels.

This issue also arises when there is a many-to-many relationships between parent and child levels in a dimension hierarchy. A hierarchy that has at least one many-to-many relationship is called *non-strict* (Vaisman and Zimányi, 2014). Non-strict hierarchies may induce the double counting of measures when a roll-up operation reaches a many-to-many relationship. Let us consider the case in which the cube example has also a dimension for the Physician that prescribed the treatment, that rolls-up to the physician's Specialty. Since a doctor may have more than one specialization, we would be dealing with a many-to-many dimension. If a user were to query the system by asking the total cost of therapies aggregated by the physician's specialty, there might be a case in which the drug *Aspirin* has been administered in two occasions, one for treating a headache and one for a cardiac disorder. Both treatments may be prescribed by the same physician that happens to be a cardiologist and a neurosurgeon.

One of the solutions to the double-counting problem consists in indicating how measures are distributed between several parent members for many-to-many relationships (Vaisman and Zimányi, 2014). The way a measure may be distributed depends on what it represents, or on its semantics. For some measures it is reasonable to think that their value may be splitted to various members. In the example discussed above, the treatments Cost may be divided according to the number of specialities of the physician that prescribed a particular drug. This is strictly related to the aggregation operation that is going to be performed. The treatment Cost is likely going to be summed to other treatments, and it would be incorrect to take it

into account many times. On the other hand, other aggregate operations will be used with the daily dosage (e.g., max or min). In this case, the value may be replicated without any splitting operation. In our running example, the Cost measure will be equally splitted among multiple cells in case of the sum operations.

We describe next how cubes related by a link may be queried. Consider again the query above: *Show the total cost and the max daily dosage of treatments per adverse reaction.* For answering this query, the measures of the cube need to be adjusted when the sum aggregation operator is applied. To this purpose, the number of treatment replications is counted by grouping the data according to treatment keys, as shown in Table 1 . The result for the Treatment is shown in Table 5. The second row of this table indicates that the treatment T2 appears 2 times in the joined table. The number of the treatment repetitions, depicted in Table 5, is used to adjust the cost value of 20 for T2 in Table 1. This will assure that the sum aggregation will return the correct value. The same adjustment is performed for the T3 treatment.

TreatmentKey	**ReplicationCount**
T1	1
T2	2
T3	2
T4	1
T5	1
T6	1
T7	1
T8	1

TAB. 5 – *Number of occurrences of each treatment in the join of the two cubes*

Finally, we may now compute the aggregate results by applying the sum and max operations on Cost and DailyDosage, respectively. The result of the query is shown in Table 6. The result is the correct one, and this can be easily verified by computing the sum of all the treatments cost in Table 1, 420, which is the same result we obtain by computing the sum of the costs in Table 6.

AdverseReaction	**TotalCost**	**MaxDailyDosage**
Hepatitis	257,5	50
Urticaria	162,5	40

TAB. 6 – *Result of the query that shows the total cost and the max daily dosage of treatments per adverse reaction*

The same query may be expressed by means of the SQL on the relational schema depicted in Figure 2. The usual way to combine fact tables is through the drill-across operation which performs the join through their shared dimensions, but as we have seen this is not appropriate. Therefore, the new Drill-Across-Link operator should be used, since it takes into account the link between the fact tables. The relationship between the two facts is many-to-many, possibly leading to the double counting problem. The following SQL query expresses the above query *Show the total cost and the max daily dosage of treatments per adverse reaction*, which result is depicted in Table 6.

```
CREATE VIEW Drill-Across-Link AS (
        SELECT S.TreatmentKey, S.AdvReactionKey, T.StartDateKey AS TreatStartDateKey,
                T.EndDateKey AS TreatEndDateKey, T.DrugKey, T.PatientKey,
                T.DailyDosage, T.Cost, A.StartDateKey AS AdvReactionStartDateKey,
                A.EndDateKey AS AdvReactionEndDateKey, A.LLTKey, A.Severity
        FROM    Treatment T JOIN Suspected S ON T.TreatmentKey = S.TreatmentKey
                JOIN AdverseReaction A ON S.AdvReactionKey = A.AdvReactionKey AND
                T.PatientKey = A.PatientKey )
SELECT      A.AdvReactionKey, SUM(Cost/COUNT(DISTINCT M.TreatmentKey))
            AS TotalCost, MAX(DailyDosage) AS MaxDailyDosage
FROM        Drill-Across-Link D JOIN LowestLevelTerm L ON D.LLTKey = L.LLTKey
GROUP BY L.LLTName
ORDER BY L.LLTName
```

8 Conclusions

Data analysts often need to combine related data from heterogeneous multidimensional cubes. However, they are restricted in using only shared dimensions for navigating related multidimensional cubes. In this paper, we show that the inclusion of non-shared dimensions in the navigation of data from multiple cubes provides the analyst with the ability to view and analyze data that would be otherwise not available. To overcome this limitation, we introduced an explicit link that relates two multidimensional cubes, indicating that they represent different aspects of the same reality, and hence they may be connected. We argued that the standard drill-across operator is not suited to perform such operation, and we extended it by proposing a new operator called drill-across-link. Finally, we also addressed the double-counting problem that arises when merging the two cubes.

References

Abelló, A., J. Samos, and F. Saltor (2002). On relationships offering new drill-across possibilities. In *Fifth International Workshop on Data Warehousing and OLAP, Proceedings*, pp. 7–13.

Jensen, C. S., A. Kligys, T. B. Pedersen, and I. Timko (2004). Multidimensional data modeling for location-based services. *VLDB 13*(1), 1–21.

Kimball, R. and M. Ross (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley Publishing.

Lechtenbörger, J. and G. Vossen (2003). Multidimensional normal forms for data warehouse design. *Inf. Syst. 28*(5), 415–434.

Lehner, W., J. Albrecht, and H. Wedekind (1998). Normal forms for multidimensional databases. In *10th International Conference on Scientific and Statistical Database Management, Proceedings*, pp. 63–72.

Lenz, H. and A. Shoshani (1997). Summarizability in OLAP and statistical data bases. In *Ninth International Conference on Scientific and Statistical Database Management, Proceedings*, pp. 132–143.

Malinowski, E. and E. Zimányi (2006). Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data Knowl. Eng. 59*(2), 348–377.

Mansmann, S. and M. H. Scholl (2007). Empowering the OLAP technology to support complex dimension hierarchies. *IJDWM 3*(4), 31–50.

Pedersen, T. B., C. S. Jensen, and C. E. Dyreson (2001). A foundation for capturing and querying complex multidimensional data. *Inf. Syst. 26*(5), 383–423.

Rafanelli, M. and A. Shoshani (1990). STORM: A statistical object representation model. In *5th International Conference of Statistical and Scientific Database Management, Proceedings*, pp. 14–29.

Riazati, D., J. A. Thom, and X. Zhang (2008). Drill across and visualization of cubes with non-conformed dimensions. In *Database Technologies 2008. Proceedings of the Nineteenth Australasian Database Conference, Proceedings*, pp. 85–93.

Song, I., W. Rowen, C. Medsker, and E. F. Ewen (2001). An analysis of many-to-many relationships between fact and dimension tables in dimensional modeling. In *3rd Intl. Workshop on Design and Management of Data Warehouses, Proceedings*, pp. 6.

Torlone, R. (2008). Two approaches to the integration of heterogeneous data warehouses. *Distributed and Parallel Databases 23*(1), 69–97.

Vaisman, A. A. and E. Zimányi (2014). *Data Warehouse Systems - Design and Implementation*. Springer.

Résumé

Les entrepôts de données structurent les données dans cubes multidimensionnels, où les dimensions précisent différentes façons dont les mesures des faits peuvent être consultées, agrégées et triées. Il est essentiel pour les analystes de combiner les données de cubes multidimensionnels hétérogènes afin d'améliorer leurs capacités d'analyse. Pour cela, les utilisateurs sont limités à n'utiliser que les dimensions communes pour naviguer des cubes multidimensionnels connexes. Dans cet article, nous montrons que cela limite les possibilités d'analyse. Nous introduisons un lien explicite qui lie deux cubes multidimensionnels, ce qui indique qu'ils représentent différents aspects d'une même réalité, et donc ils peuvent être connectés. Nous soutenons que l'opérateur drill-across traditionnel n'est pas adapté pour effectuer cette opération, et nous l'étendons en proposant un nouvel opérateur appelé drill-across-link.