International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Using Multidimensional Canonical Partitioning (MCP) as a Supply-Driven Approach for Data Warehouses Design

Apollinaire Bamana Batoure[a*], Kolyang[b], Michel Tchotsoua[c]

[a]*The University of Ngaoundéré. Department of Mathematics and Computer Science, Po Box 454, Ngaoundéré, Cameroon*

[b]*The University of Maroua, Department of Computer Science, Po Box 46, Maroua, Cameroon*

[c]*The University of Ngaoundéré, Departement of Geography, Po Box 454, Ngaoundéré, Cameroon*

[a]*Email: apollinaire.batoure@univ-ndere.cm*

[b]*Email: kolyang@univ-maroua.cm*

[c]*Email: tchotsoua@gmail.com*

Abstract

Various information systems have been developed for decision support. But, they rely essentially on transactional methods. From data and transactional databases, we proposed a supply-driven approach to design data warehouses. The approach takes as input, a universal relation, applies vertical partitioning by a greedy type heuristic algorithm. Partitions obtained are transformed into dimensions using a matching algorithm. The other elements of the multidimensional annotation are deduced by guidelines, and the data warehouse schema is generated using a multidimensional conceptual pattern. The transformation of those transactional systems into decision support ones aims at facilitating the storage, exploitation and the representation of data using new databases generation technologies.

Keywords: Data warehouse design approach; multidimensional data schema; relational database; universal relation; vertical partitioning.

1. Introduction

Design of decision support systems or data warehouses (DW) is still a challenge. The discipline does not have an established and recognise method, compare to other software engineering fields [1, 2]. Various approaches are proposed.

---

* Corresponding author.

They are classified in three main categories adopted by researchers and industrials. We distinguish *user-driven, demand-driven* or *top-down* approaches which define conceptual schema of the DW using decision maker's needs; the *supply-driven, data-driven* or *bottom-up* approaches which use existing data and information systems; and the *mixed* or *hybrid* approaches who take into account both the needs of users and existing data [3, 4, 5].

The surveys done in [1, 6] show that most of the design approaches have been developed between 1998 and 2010. The following works focused on the improvement of existing approaches or use of other types of sources such as *UML, i* framework, ontologies* or *Web* [7].The main purpose of this ongoing work is to transform legacy systems into decision support systems. Sources are from *entity-relationship* (ER) schema. We study about fifteen supply-driven approaches having ER schema as input. Most of them rely essentially on functional dependencies and cardinality of relationships between tables to produce data warehouses schemas. They are not global in terms of relational database taken as input. We propose an approach to transform relational databases [8] into decision support systems [9, 10]. The approach proposed takes into account, existing data and systems to produce data warehouses schemas. The processing needs a *universal relation* (UR) [11, 12]. A universal relation allows a view of the database as if it was composed of a single flat relation containing all characteristics which describe real word entities [13]. It's a view (external schema) on top of a relational database schema (conceptual schema) [14] which offers a data model for advanced applications [15]. The universal relation model was first introduced as a means to free the users from the need to know the logical navigation of the database [16]. Several versions of universal relation assumptions, with respect to relational systems have been introduced to satisfy different objectives. A simple illustration of the notion of universal relation assumptions is as follows:

Universal relation assumption [16]:

For the set of relations $S = \{R1 < X1; D1 >, R2 < X2; D2 >... Rn < Xn; Dn >\}$, there exist a Universal Relations $UR < T; G >$ such that:

(1) The columns of *UR* consist of all the columns of the relations in $S: T = X1 \cup X2 \cup ... \cup Xn$

(2) Each relation in S is a projection of $UR: Ri = U[Xi]$.

Our objective is to derive the multidimensional schema [17, 18] using this type of relation. On the best of our knowledge, there is no approach using universal relation to design data warehouses, as the one we proposed here. After that introduction, the rest of the paper is organised as follows: section 2 describes the Multidimensional Canonical Partitioning approach. Section 3 characterizes the approach, before the conclusion ends.

2. Multidimensional Canonical Partitioning

In this paper, we use universal relation to obtain multidimensional data schema. Let us note $R = <C, D>$ a relational schema where C is a set of constituents and D, a set of functional dependencies. Constituents represent

characteristics of fields. Functional dependencies express constraints between fields [8, 19]. Therefore, a universal relation put together all the constituents C and all the dependencies D. Each constituent or component describes one concept with a precise semantic, different from all the others. Such an attribute plays a single role, irreplaceable, in the general comprehension of the data schema [20]. We can assume that the same attributes describe the same concepts. The relation is on its first normal form [21] because attributes are defined in an atomic form, without any ambiguity. After a first processing, a so defined relation is divided in groups of attributes describing a given entity. Later, the entity or partition can be used as dimension [9]. We use the partitioning and not the decomposition [20, 22] because we are most interested in attributes' semantic and less by dependencies between them. The schema we want to get at the end of the process is a multidimensional one. This schema is a data mart (star or snowflake) [10, 17, 18]. It can also be on constellation [17] forms if the facts table shares some dimensions. The proposed schema, to be multidimensional, has to respect some canons: the identification of facts table, measures, associated dimensions and granularities. The relationships between different tables will permit us to define primaries and foreigners integrities [8]. For the rest of the paper, we propose an approach named *multidimensional canonical partitioning* (MCP) to split universal relation and set parts into multidimensional norms. The approach is supply-driven [9] because it takes, as input, existing systems and data.

Figure 1 illustrates the whole process of the transformation of universal relation into multidimensional schema. Using a universal relation (*first shape*), we apply MCP approach (*second shape*) to obtain multidimensional elements such as dimensions, granularities, facts, measures, and keys (*third shape*). Then, data warehouse can be generated (*forth shape*).

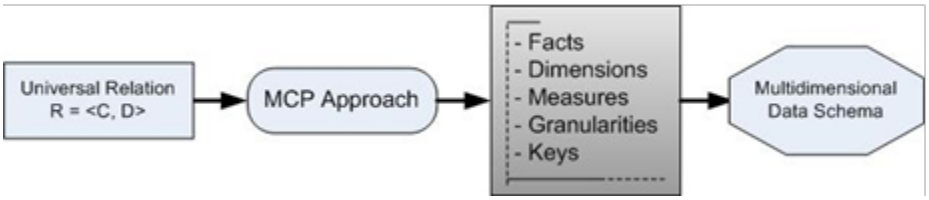

Figure 1: Process of transforming universal relation into multidimensional schema

The approach involves six steps. It begins by verifying if the schema supplied as input is on universal relation form. If not, it can be transformed so as to get one [11, 14, 15, 23]. After this stage, we split the universal relation into non-empty and disjoints sub-sets. Union of sub-sets should give the whole set.

These sub-sets or partitions will be candidates to be dimensions. Partitioning, both horizontal and vertical, aims at improving transactions and requests performances in information systems [24, 25]. Here, we use the vertical one. It is based on attributes of the original relation. The approach should create partitions and arrange each attribute under a given partition. We therefore set an attribute under a partition, *if and only if* the attribute describes that partition. We define the subsequent function, to be applied on attributes relatively to partitions. "*Describe*" refers to a property or characteristic of the entity in semantic database modeling [20, 26]. The function has as input, one attribute and one partition and returns a Boolean if the attribute describes the partition.

(1) $$f(attribute_i, partition_k) = \begin{cases} 1 \text{ if } attribute_i \ describes \ partition_k \\ 0 \text{ else} \end{cases}$$

i covers attributes and k partitions; a given attribute describes one and only one partition.

We are now going to describe each of the six steps of the approach.

2.1. Verification of input schema

We apply the partitioning on the universal relation. For this first step, if the database schema given as input is not on that form, we can restructure it. The works done in [14, 15] show how to get view of database schema as a single universal relation with no ambiguity and less prerequisites. To achieve this goal, they use special definition of projection and join. Universal relation assumption [16] can also help. If the schema is already on universal relation form, we go straight to the next step.

2.2. Partitioning of the universal relation

This step aims at determining eligible partitions and their attributes. We start with a set of sets (set of partitions). Partitions are sets of attributes. Set of partitions and partitions are initialised empty. For the first non-primary key attribute read, we create the first partition. Then, we read through the remaining attributes. For each one, we check if it describes an existing partition. If yes, we include it in that partition. If no, we create a partition and include the attribute. At the end, we have the partitions and their attributes. The output is the partitions' set. We deduce the following algorithm:

Algorithm *partitioning*;

var i, k: integer;

var n: integer; (attributes non key of the UR)

var part: set of attributes;

var partitions: set of part;

Begin

i ← 1; k ← 1; read (n);

part ← Ø; partitions ← Ø;

$part_1 \leftarrow part_1 \cup \{att_1\}$;

partitions ← partitions U {$part_1$};

for i **from** 2 **to** n

do for all $part_k$ **in** partitions;

if f(att_i, $part_k$) **then** $part_k$ ← $part_k$ U {att_i};

else {create $part_{(k+1)}$; partitions ← partitions U {$part_{(k+1)}$}};

endif

endfor;

return partitions;

End;

The algorithm is a greedy type heuristic algorithm. We construct a feasible solution of the partitioning by successive best decisions taken, in relation with a local criteria *(describe)*, without contradicting previous decisions [27]. The obtained solution is an approximate one, like all design solutions. This algorithm is on O(n*m) complexity. n represents the number of attributes of the UR and m, the maximum number of partitions we can get. At the outlet of this stage, we have partitions and properties describing them. Partitions are candidates for the role of dimension in the upcoming data warehouse schema.

2.3. Processing of partitions into dimensions

This step aims at matching partitions in order to deduce effective dimensions. For each partition, we first of all add attribute with primary key properties. For the other attributes of partition, we apply one of the followings: either we delete it, either we modify (**MOD**) it or we just keep it (**OK**). At the end, the possibility is given to add relevant attributes. We propose the bellow procedure. It takes as input, a set of partitions and returns a set of dimensions.

Procedure *dimension* partitions: dimensions;

var i, k: integer;

var m: integer; (number of partitions)

var dim: set of attributes;

var dimensions: set of dim;

Begin

i ← 1; k ← 1; read(m);

dim ← Ø; dimensions ← Ø;

for k **from** 1 **to** m **do**

for i **from** 1 **to** n **do**

if att_i OK **then** $dim_k \leftarrow dim_k \cup \{att_i\}$

else

if MOD **then** {rename att_i; $dim_k \leftarrow dim_k \cup \{att_i\}$};

endif;

endif;

endfor;

→ rename, if necessary, the dimension;

→ add necessaries attributes to dim;

→ add primary key to dim;

→ dimensions ← dimensions U $\{dim_k\}$;

Endfor;

return dimensions;

End;

OK and MOD are Boolean, applied on attributes. OK means that the attributes does not need change. MOD means that the attribute has to be modified. This procedure is equivalent to a matching algorithm between partitions and dimensions. The algorithm performs on O(p*q) complexity. p is the total number of partitions as input and q, the maximum number of attributes of the biggest partition.

2.4. Normalization of dimensions

After obtaining dimensions, we can normalise them. It's the concern of this step. This permits us to deduce eventual granularities. Granularities or hierarchies give more details on dimensions. We use the third normal form (3NF) algorithm [8, 21] because it is the level we want to reach. Actually, according to the definition of UR, it's on first normal form (1NF). As well as primary keys are not composited, we can easily get the second normal form (2NF). The following work is based on decomposition algorithm and functional dependencies.

2.5. Construction of the facts table

The facts table is the central table in a data warehouse schema. It contains measures or values for decision support. Values are computed from information in dimension by aggregates functions. The Galois lattice is used to determine view of all possible measures for the facts table. The single minimal node represents the aggregation of all quantities.

At this stage of the approach, we construct the facts table by using guidelines. We add to facts table primary keys of the linked dimensions. The set of these primary keys is its primary key. Needed measures for decision support indicator are also added.

2.6. Generating the multidimensional schema

The last step is the generation of the data mart schema. We use a multidimensional conceptual pattern, on spatio-temporal form. Conception patterns are approved solution for recurrent design problems in a given context [28]. They are defined as abstract way (using UML for example) and concrete way (using archetype implementation). This stage is out of the scope of this paper.

In this section, we have defined a design approach for data warehouses, using existing databases, especially a universal relation. We are now going to characterize the work.

3. Approach characterization

The approach proposed in this paper consist of designing multidimensional data schema using existing systems on Entity/Relationship form. Up to now, there is no approved method for data warehouses design [1, 4]. We use universal relation as input of the approach. The approach has the advantage to start constructing the multidimensional schema from a terraced relation. We don't mind on functional dependencies, cardinality of relationships, relevant or non-relevant entities as in most of other approaches. If the provided schema is not on universal relation form, there is a possibility to transform it. Another achievement is the possibility to generate multidimensional schema at the outlet of the approach. For this purpose, we use a multidimensional conceptual pattern. It gives the facility for recurrent design activities. Must of the proposed approaches does not give the way for building the multidimensional schema. The proposed approach is more global in terms of transactional data schema which can be used to be transformed in data warehouses schema.

The main drawback or difficulty of the approach is if the provided data schema is not on universal relation form. We then need to transform it. The stage cannot be easy or be fastidious if the relational schema has too many relations.

The approach has six steps. Steps 1 and 4 are works we just use. The others (steps 2, 3, 5 and 6) are the ones we propose. They consist of a greedy type heuristic algorithm for the construction of partitions. The obtained partitions are transformed, by a matching algorithm, in dimensions. After the construction of the facts table, the data warehouse schema can be generated.

Figure 2 summarizes steps of the approach. It is the explosion of the above-mentioned black box (*figure 1*).

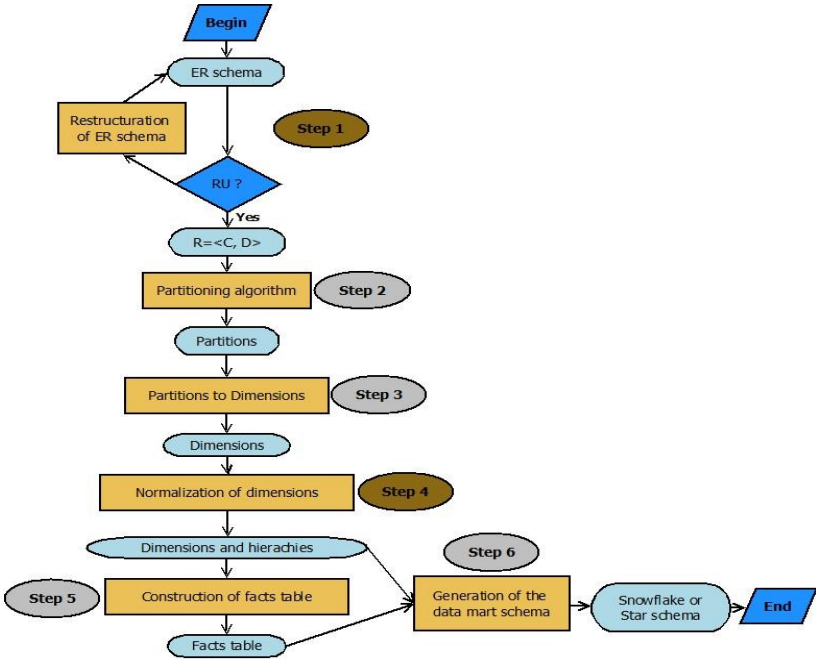

Figure 2: Steps of the design approach

Some criteria have been defined in [7] for the classification of multidimensional design approaches. They take upstream the design, general criteria defining the approach principle; and downstream, criteria describing how the approach is applied.

According to this classification, we can notice that, approach proposed in this paper is supply-driven, using guidelines and algorithms. The automation is not yet managed. Sources are existing data and systems. For the modeling process, facts are identified by guidelines according to the thematic studied. Dimensions are identified by heuristic method associated with a matching algorithm. The hierarchies (granularities) of dimensions are defined using functional dependencies by third normal form algorithm.

Conceptual modeling is effective and the formalism is ER model. Logical modeling is done by relational

implementation through star, snowflake or constellation schema. Physical modeling is out of the scope of this paper. It is done using model-driven architecture.

4. Conclusion

In this paper, we have proposed a design approach for data warehouses. It takes into account existing data and systems. No matter the type of databases schema, the approach can be applied. It is why the approach is more global in terms of transactional data schema which can be used to be transformed in data warehouses schema. Six steps are required for this approach. We use two algorithms. One of them is heuristic. The objective of transforming transactional systems into decisional one is to be able to use data warehouses technologies to realise decision support systems. After that, we can apply data mining and big data techniques.

References

[1] J. Rajni and T. Shweta. “Comparative Study of Data Warehouse Design Approaches: A Survey.” International Journal of Database Management Systems (IJDMS), vol. 4, No.1, 13p, 2012

[2] S. Shashank and K. Manoj. “Comparison of Data Warehouse Design Approaches from User Requirement to Conceptual Model: A Survey.” International Conference on Communication Systems and Network Technologies, IEEE, 5p, 2011

[3] I. Gam. "Ingénierie des exigences pour les systèmes d’information décisionnels : concepts, modèles et processus - la méthode CADWE." PhD thesis in Computer Science. Université Panthéon- Sorbonne - Paris I, 320p, 2008

[4] A. Cravero and S. Sepúlveda. "A chronological study of paradigms for data warehouse design," Ingeniería E Investigación. vol. 32, No. 2, pp. 58-62, 2012

[5] A. Vaisman and E. Zimanyi. "A Method for Data Warehouse Design," Chapter 10 In Data Warehouse Systems, Data-Centric Systems And Applications, Springer-Verlag Berlin Heidelberg, 2014, 39p;

[6] A. Cravero and S. Sepúlveda. "Methodologies, techniques and tools for OLAP design: A Systematic Mapping Study." IEEE Latin America Transactions, vol. 14, No. 2, 8p, 2016

[7] S. Khouri. "Cycle de vie sémantique de conception de systèmes de stockage et manipulation de données". PhD thesis, ISAE-ENSMA École Nationale Supérieure de Mécanique et d’Aérotechique - Poitiers, 246p, 2013

[8] J. Akoka and I. Comyn-Wattiau, "Conception des bases de données relationnelles, en pratique.”, Collection Informatique, Vuibert, Paris, 2001

[9] W. Inmon. Building the Data Warehouse. Fourth Edition. Willey Publishing, 2005, 576p

[10] G. Gardarin. Internet/intranet et bases de données: Data Web, Data Media, Data Warehouse, Data Mining, Edition Eyrolles, 2005, 246p

[11] D. Maier, J. D. Ullman and M. Y. Vardi. "On the foundations of the universal relation model," DI, A C M Trans. Database Systems 9, No. 2, 283-308, 1984

[12] F. Leymann. "A Survey of the Universal Relation Model." Data & Knowledge Engineering 4, pp. 305-320, 1989

[13] R. Fagin, A. O. Mendelzon and J. D. Ullman. "A simplified universal relation assumption and its properties," ACM Trans. Database Systems 7, No. 3, pp. 343-360, 1982

[14] J. Biskup and H. H. Brüggemann. "Universal relation views: A pragmatic approach". In Proceedings of the 9th International Conference on Very Large Data Bases, pp. 172–185, 1983

[15] H. Hyotyniemi and A. Lerhtola. "A universal relation database interface for knowledge based." In Proceedings of the Second International Symposium, Database Systems for Advenced Applications'91, Tokyo, Japan, Apr. 1991- Singapore : World Scientific, 1992, pp. 84-88

[16] S. Harihodin. "A Universal Relation Approach For Natural Query In Logic Database System," Jurnal Teknologi, vol. 20, pp. 46-64, Dec. 1992

[17] C. Imhoff. Mastering the Data Warehouse Design, Relational and Dimensional Techniques. Willey Publishing, 2003

[18] C. Adamson. Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance. Willey Publishing, 2006, 377p

[19] E. F. CODD. "Recent Investigations in Relational Database Systems," IFIP Congrès, North-Holland Ed., pp. 1017-1021, 1974

[20] R. HULL and R. KING. "Semantic Database Modelling: Survey, Applications, and Research issues." ACM Computing Surveys, vol. 19, No. 3, 60p, Sep. 1987

[21] R. Fagin. "Normal Forms and Relational Database Operators," ACM SIGMOD 1979, Boston, pp. 153-160, Jun. 1979

[22] J. Gray and A. Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufman Ed., 1993, 1070p

[23] M. Y. Vardi. "The universal-relation data model for logical independence," IEEE Software 5, pp. 80-85, 1988

[24] E. S. Abuelyaman. "An Optimized Scheme for Vertical Partitioning of a Distributed Database," International Journal of Computer Science and Network Security (IJCSNS), vol.8, No.1, pp. 310-316, Jan. 2008

[25] S. Agrawal, V. Narasayya and B. Yang. "Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design." SIGMOD 2004, Paris, France, 12p, Jun. 2004

[26] M. Bouzeghoub and E. Métais. "Semantic Modelling of Object Oriented Databases." 17th Very Large Database International Conference, Morgan Kaufman Pub., Barcelone, Espagne, 1991

[27] M. G. C. Resende and C. C. Ribeiro. "Handbook of Metaheuristics", chapter Greedy randomized adaptive search procedures, Kluwer Academic Publishers, pp. 219–249, 2003

[28] J. Feki and Y. Hachaici. "Conception assistée de MD. Une démarche un outil," Journal of Decision Systems, vol. 16 – N° 03, pp. 303-333, 2007.