International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

DATA WAREHOUSE: CONCEPTUAL AND LOGICAL SCHEMA -SURVEY-

Nouha Arfaoui [a], *Jalel Akaichi* [b]

[a,b]*WDW-SOIE, Institut Superieur de Gestion,41, Avenue de la liberté,Cité Bouchoucha,*
Le Bardo 2000,Tunisia

Abstract

The Data Warehouse (DW) is considered as a repository that contains data collected from different sources. Its design is one of many issues treated in the literature. It is considered as the most important since it influences the quality of DW projects. Despite the number of works that have done, the design still suffers from many problems such as the lack of a consistent methodology that assists the user.

In this work, we present a survey. It contains a presentation of the conceptual design models and the different logical schemas that exist.

Keywords: Conceptual Modeling, Conceptual Schema, Logical Schema.

1. Introduction

The DW is considered as a “collection or repository of integrated, detailed, historical data to support strategic decision making” [11], and according to the same authors, it serves “as a data repository that stores data from disparate sources, making it accessible to another set of data stores”.

The design of the DW is important since it ensures its creation according to the users' needs and it is different compared to OLTP design as presented in [10].

First concerning the models, usually the operational models are typically ER, and for the DW are dimensional.

Concerning the utility of the model, in the operational word the user uses the models as a tool to capture requirement not to have an access to the data. In the DW, to the user the data must look like DW model.

Concerning the purpose behind the creation of each one, the design in an operational is concerned with creating a database that will perform well based on a well-defined set of access paths. Data warehouse design is concerned with creating a process that will retrieve and transform operational data into useful and timely warehouse data.

And finally, in terms of performance, its considerations cannot be handled in a data warehouse in the same way they are handled in operational systems. In fact, the unpredictable characteristic of the DW's queries limits how much further you can design for performance.

And because of this difference, many elements are not considered as the same way. We can present as example the redundancy of data [4], it is important in the case of the DW since it is admitted for improving the performance of the complex queries, also it must take into account not only the DW requirements but also the features and existing instances of the source databases.

The design has made the subject of many works. Some of them present personal experiences in building the DWs, and as result, many approaches have been proposed and they are different according to the target (DW or DM). This variety can cause confusing for designers even experienced ones [21]. Despite all the existing works, it presents some problems such as the lack of methodological framework that helps the designer during the DW development process [21] also there are no efforts have been so far to develop a complete and consistent design methodology [32].

To ensure the design of DW, three approaches are proposed top-down, bottom-up and middle-out.

The top-down takes into consideration the needs of all users. It constructs then one schema corresponding to the entire DW [21].

The bottom-up focuses on building the schema of each data mart taking into consideration the requirements of the decision-making users. Then all the schemas are merged to form one global schema corresponding to the DW [21].

The middle-out takes advantages of the two previous approaches. It has the speed and the user-orientation of the top-down and the integration enforced by a DW in top-down [47].

The design is composed by set of steps. The number is different from one work to another. Indeed, the authors in [40] and [46] propose three steps: conceptual level, logical level and physical level. In [21], the methodology is based on four steps including requirements specification, conceptual design, logical design, and physical design. Those phases are independent of the used approach (top-down or bottom-up). In [33], the authors propose a generic methodology. It is composed by 8 steps: requirements analysis, analysis and reconciliation, conceptual design, workload refinement, logical design, data staging design, physical design and implementation.

Let's give more detail about the last methodology, indeed, during the requirements analysis, the user defines his/her requirements which are represented informally using propos glossaries or formally. The analysis and reconciliation serve to construct schema from the inspected, normalized and integrated data sources. The user requirements and the available data are used as input to constitute the fact schemata in the conceptual design level. The workload refinement refines the preliminary workload as expressed by the users. The logical design translates the conceptual schema. For the data staging design, the ETL procedures are designed considering the source schemata and the reconciled schema. The physical design includes index selection, schema fragmentation and all other issues related to physical allocation. And finally the implementation, it includes the implementation of ETL procedure and the creation of front-end reports.

In this work we present a survey that describes the conceptual as well as the logical level. To respond to our objective, this paper is organized as follow:
In section 2, we present the different models used to ensure the conceptual design of the DW.
In section 3, we describe the logical level.
In section 4, we finish with the conclusion.

2. The conceptual design of DW

The conceptual design allows having closer ideas about the ways that a user can perceive an application domain [20]. In fact, it is considered as a key step that ensures the successful of the DW projects since it defines the expressivity of the multidimensional schemata [34], and the result of this step is a graphical notation which facilitates to the designer and the user different tasks such as writing, understanding and managing the conceptual schemata [15].
Despite the diversity of models proposed in this level (ER extension, OO, ad-hoc, etc), none of them is considered as a standard because of the absence of agreement between the researches and the industrial communities about the multidimensional properties to be modeled, the problems related to the translation of some modeled properties existing in the conceptual level to logical level, and the vendors who everyone propose his/her design method.
In the literature there are many proposed models, we can categorize them into four different groups: extension-ER models, object-oriented models, ontology models and ad-hoc models. In the following, we present a thorough study of each type.

2.1. The extension-ER models

According to many works such [42][12], they are widely agree that ER is not appropriate to deal with multidimensional concepts as well as the multidimensional and aggregative nature of OLAP applications, it does not provide a suitable means to describe the DW design that needs to represent explicitly certain important aspects that are related to the abstract representation of real world concepts and to realize the final goal of the DW that supports the data analysis oriented to the decision making [42]. In fact, for each conceptual model, two specific notions must be recognized and introduced: the fact and the dimension.

In the following, we present panoply of models based on ER model.

In [12], the authors present **Multidimensional Entity Relationship** (**MER**) that deals with the conceptual modeling of OLAP applications. To represent the multidimensional semantic, MER is constructed by three main keys the specification of the ER model, the minimal extension of the ER model, and the representation of the multidimensional semantic.

Concerning the specialization of the E/R model, all elements that are introduced should be special cases of native E/R constructs. Thus, the flexibility and expressiveness of the E/R model is not reduced. For the minimal extension of the E/R model, the specialized model should be easy to learn and use for an experienced E/R modeler. Thus, the number of additional elements needed should be as small as possible, and for the representation of the multidimensional semantics, despite the minimality, the specialization should be powerful enough to express the basic multidimensional semantics, namely the separation of qualifying and quantifying data and the hierarchical structure of the qualifying data.

Concerning the hierarchical classification structure of the dimensions, it is expressed by dimension level and roll-up relationships that defines a directed acyclic graph on the dimensional level.

The fact relationship set contains measures considered as multiple attributes. The MER has as common with ER the static structure related to the application domain, and for the measure, its calculation uses a functional information that is not included in the static model.

In [18], [19] and [20] the authors propose **MultiDimER** which is a conceptual multidimensional model based on the ER model. The MultiDimER is constructed with ER semantic including: entity types, attributes, and relationship types. Its schema is presented as a finite set of dimensions and fact relationships. This conceptual model focuses on the spatial data which requires considering different features such as spatial dimensions, spatial hierarchies, spatial facts, relationships and spatial measures. Concerning the spatial dimensions, it includes the spatial level and the spatial hierarchy. The spatial level is a level for which the application needs to keep its spatial characteristics (including the geometry: point, line, area, or collection of these data).The spatial hierarchy includes at least one spatial level. A spatial dimension it includes at least one spatial hierarchy. For the fact relationships, they link leaf members from all dimensions participating in the relationship, and it requires a spatial join between two or more spatial dimensions, and finally the spatial measure that can be associated to a fact relationship independently of whether the relationship is spatial or not.

The authors introduce, in [38], the **starER** as a conceptual model. It combines the star structure with the ER model, and this combination has as results the addition of special types of relationships to support hierarchies. The star schema is chosen because it is dominant in the DW and it captures its structure, in addition, the star-structure data has the facts about the companies in the center and data unfolds around them. The ER is used because of the ease of the use and the small set of supported constructs. It is composed by the entities that present the real world objects, the relationships that capture the associations among objects and the attributes that represent the properties of entity or relationship.

In [6], the authors propose a conceptual data model based on ER model and called **CGMD**. The proposed model captures database schemata expressed using ER diagram and describes multidimensional structure including dimensions with their hierarchically organized levels and the structure of aggregations. The ER is extended by the construction of the aggregated entities together with their interrelationships with the other parts of the schema.

The authors propose in this work [29] the **Structured Entity Relationship Model** (**SERM**) which is not only useful for the development of big operational systems but can also help with the derivation of data warehouse structures. The SERM is an extension of the conventional Entity Relationship Model (ERM). The SERM presents a set of advantages:

- Designing extensive data models: in the SERM the nodes are arranged in such a way as to indicate their interdependencies in a hierarchical way. This leads, to a quasi-hierarchical (acyclic and directed) graph as opposed to the bipartite graph of the ERM.
- Visualization of the order of dependencies between data object types: in contrast to the ERM, where relations between E-types are modeled, modeling in SERM means constructing the data model based on the principle of dependency.
- Avoiding inconsistencies: the hierarchical structure of a SER-diagram prevents the modeling of a cycle, a special kind of closed loop. These cycles can be syntactically correct but lead semantically to inconsistent data models.
- Avoiding unnecessary relationships: in contrast to ERM, the creation of a relational database from a conceptual data model in SERM can be done with very little structural transformations.

Besides the E- and the R-type, the SERM also includes an entity relationship-type (ER-type). This is a combination of an E-type and a R-type with a (1, 1)-relationship. Different kinds of edges between the data object types correspond with the specification in (min, max)- notation.

2.2. The Object-Oriented models

According to [1], the power of the Object-Oriented paradigm (OO) is because of its ability to support six dimensions (i.e. Classification/Instantiation, Generalizing/Specialization, Aggregation/Decomposition, Derivability, Caller/Called, and Specialization). Each one provides the data model with a little of semantic power. This paradigm is more expressive and better represents static and dynamic properties of information systems [34]. It is the current dominant trend in the field of data modeling. It is presented through the UML (Unified Modeling Language), this is because the UML is a standard and naturally extensible also it provides powerful mechanism such as the Object Constraint Language and Object Query Language for embedding data warehouse constraints and initial requirements in the conceptual model [27].

The UML is proposed in [27] to ensure the conceptual design of the DW. This choice is because, according to the authors, the UML considers an information system's structural and dynamic properties at the conceptual level; also, it provides powerful mechanisms such as the object constraint language and the object query language for embedding DW constraints and initial user requirements in the conceptual model.

In the following, we present some models existing in the literature and used to construct the conceptual schema.

The authors present a multidimensional conceptual object oriented model **Yet Another Multidimensional Model** (**YAM^2**). It has been developed as an extension of UML core meta-classes [3], [2]. To be relevant, YAM^2 must satisfy two main characteristics: the expresssiveness or Semantic Power, and the Semantic Relativism.

The expressiveness or Semantic Power corresponds to the degree to which a model can express or represent a conception of the real world. It is important for the YAM^2 since they are used to represent user ideas, through the nodes and arcs to improve the expressiveness.

The Semantic Relativism is the degree to which the model can accommodate not only on, but many different conceptions. YAM^2 provides mechanisms to model the same data from different points of views.

In those papers [44] and [45], the authors present an extension of UML using the **UML profile** which is defined by a set of stereotypes, constraints and tagged values to elegantly represent main MultiDimensional (MD) properties at the conceptual level.

The properties of the MD aspects are specified by means of a UML class diagram that serves to separate the facts and the dimensions. This work takes into consideration the following main features: many-to-many relationships between facts and dimensions, degeneration facts and dimensions, multiple and alternative path classification hierarchies, and non-strict and complete hierarchies.

The UML profile focuses on different levels of details that show how one package can be further exploded by defining their corresponding elements into the next level as following:

- Level 1: Model definition. It describes the dependency between two packages. In case of a star schema presented in a package, this dependency indicates that the star schemas share at least one dimension.
- Level 2: Star schema definition: it describes the dependency between two dimension packages. In the case where the package represents a fact or a dimension belongs to the star schema, this dependency indicates that the packages share at least one level of a dimension hierarchy.
- Level 3: Dimension/Fact definition: the package in this level is exploded into a set of classes that represent the hierarchy levels defined in a dimension package or in the case of a fact package, it represent the whole star schema.

The authors propose the model **GOLD** ([26]) as extension of **OOMD** ([25]). The two models define the multidimensional database using: dimension classes (DC) (they contain dimension objects that provide the characteristics of the actual data. They are specified as components classes in an aggregation relation), fact classes (FC) (they contain fact objects that represent the factual data itself. They are defined as composite classes in an aggregation relation), cube classes (they are defined from DC and FC. They allow accomplishing a subsequent data analysis), and views.

The OOMD and GOLD introduce the aggregation patterns of fact attributes to take into consideration the additivity. So, if the aggregation operations can be applied along all dimensions, the fact attributes can be additive. If the aggregation operations are not additive, the fact attributes can be semi-additive and if the aggregation operations are not additive along any of the dimensions, the fact attributes are non-additive.

The schema, in this case, is presented as a directed, acyclic and weakly connected graph. The edges present to-one relationship between attributes. It distinguishes between roll-up relation paths and attributes classification paths.

2.3. The ontology models

The ontology is used to solve the problem of the semantic heterogeneities that exist between different databases [15], it is used, also, to analyze the knowledge related to a specific field by modeling the relevant concepts [37]. It facilitates, then, the distinction of the different domain concept.

In the following, we present some works that integrate the ontology as a way to solve the problem of the heterogeneity data in the field of DW.

In [39], the authors present a new approach to automate the multidimensional design of DWs. This approach is based on ontology, because it serves to overcome the heterogeneity of the data source, in addition, the data sources have nothing in common, but they are described by the some domain language. The solution ensures the multidimensionality through the placement of data in a multidimensional space and correct summarizability of data.

The authors propose a set of criteria to ensure the identification of multidimensional concepts:

- The multidimensional model: it is based on the notion of fact and dimension. They will be identified along the process.
- The multidimensional space arrangement constraint: the fact must be related to each analysis dimension by many-to-many relationship. Every instance of data is related to one instance of an analysis dimension, and every dimension instance may be related to many instances of data.
- The base integrity constraint: the base implies the minimal set of levels functionally determining a fact; it corresponds to the primary key.
- The summarization integrity constraint: three necessary conditions allow performing correctly the data summarization: Disjointness, Completeness and Compatibility.

The proposed method is composed by three tasks. At the end of each step, the end-user multidimensional requirements are taken into consideration:

- The first task: it looks for the good candidates including the subject of analysis (the Fact), the potential Dimensions and Measures. At the end of this task, the users choose their subject of interest among those concepts proposed.
- The second task: it presents the set of concepts that are used as Base for each Fact identified. Bases are composed by concepts corresponding to the potential Dimensions.
- The third task: it gives rise to Dimension hierarchies. Indeed, concerning each Dimension, they conform its hierarchy of levels.

Concerning the multidimensional aspect, it includes the determination of facts, and measures. The determination of facts is done manually and it is considered as the most difficult step in the design process. In this approach, the potential subject of analysis must be related to many potential dimensions and measures and a data is related to one and just one of its instances. The measures are numeric attributes allowing data aggregation. The to-one multiplicity in the Measure side forces each Fact instance to be related to just one Measure value. The to-one multiplicity in the Fact side preserves disjointness. Bases must contain orthogonal Dimensions, and a set of potential Dimensions will be considered a feasible Base if they are able to identify all instances of a Fact, and once the Dimensions are pointed out, it is important to share the hierarchies to allow summarizability of data. The Dimension hierarchies must guarantee a correct summarizability of data. it is necessary to take into consideration to-one relationship

The authors in [5] propose the use of decision ontology to solve the problem of the integration of heterogeneous data during the design task, the automatic interpretation of the semantic of the heterogeneous and autonomous data. The use of ontology permits avoiding semantic and structural ambiguities. The specification of the decision ontology compared to the others ontology is that is dedicated to the decision systems. It assists the designer during the DW life cycle to solve the problems of data sources heterogeneity.

The proposed approach standardizes the multidimensional terminology extracted from several heterogeneous data sources.

The construction of the ontology is done in an incremental and progressive ways:

- Extraction: it consists of extracting the multidimensional concepts from heterogeneous data sources manually to build the initial version of the decision ontology. It is composed by three sub strep (extraction of the Multidimensional Concepts (MC), confirmation of the extracted MC by the designer, and extraction of multidimensional relations between the concepts).
- Comparison: it consists on a semantic comparison of MC with the ontology content. The goal is to deduce the adequate relation between two compatible MC to resolve the syntactic and semantic ambiguities between multidimensional concepts.

- Upgrade: it consists on upgrading the ontology using concepts and relations extracted in “Extraction” step. It inserts the MC and their deduced semantic relations as well as the insertion of their multidimensional relations.
- Optimization: it is related to the relations of the ontology. It uses the inference rules to discover the impact of the insertion of semantic deduced between MC and the existing ones.

In this work, the authors propose to standardize the multidimensional concepts in order to determine the semantic relations and a mapping of these concepts; it goes through set of steps: determining the semantic relations through the comparison of the name of these concepts, and factorizing the semantic relations to ensure the mapping between the concepts. The mapping implies grouping the concepts having the same type and relations of equivalence, identity or synonymy and then assigning them a significant name.

According to this work, there are five types of relations can exist between concepts (facts, measures, parameters, dimensions):

- Synonymy: it expresses that the concepts converge on the same meaning.
- Equivalence: it expresses that the concepts may converge on the same meaning.
- Identity: it expresses that the concepts have the same name and meaning.
- Homonymy: it expresses that the same concept can have two different meanings.
- Antonymy: it expresses that the concepts have no implication relationships between them.

In [16], the authors propose the use of ontology-based approach to facilitate the conceptual design of the back stage of a DW. The proposed approach supports both structured and semi-structured data and it handles them in a uniform way. It is based on the use of semantic web technology to semantically annotate the data sources and DW.

The graph used in this work is a directed graph where the nodes correspond to the elements and the edges represent containment or reference of one element by another.

The contributions of this work are:

- The use of a graph-based representation (datastore graph). It serves to represent several types of schemas such as relational and XML schemas to deal with structured and semi-structured sources in a unified way.
- The graph representation (ontology graph) it presents the different classes and properties using different symbols. It facilitates the creation task, the verification, the maintenance, and the communication between the parties involved in the project.
- Defining the mapping between nodes of the datastore graph and the ontology graph. The mapping corresponds to labels assigned to the nodes of the data store graph.
- The use of automated reasoning techniques to infer correspondences and conflicts among the datastores. By this way, we can identify the sources and propose conceptual operations allowing the integration of data into DW.

Concerning the Datastore graph is composed by two types of elements: elements that contain the actual data and elements that contain or refer to other elements

According to the authors the relation schema and the XML are considered as the most typical models used for structured and semi-structured data, they propose then how is it possible to construct the graph from the previous models:

- From Relational schema to Graph: a relational graph can be presented by a graph where the nodes correspond to the relations and non foreign key, and the edges correspond to the containment of attributes in relations and the references between the relations (the foreign keys).

- From XML schema to graph: the XML schema can be presented by directed edge-labeled graph where the nodes present: elements, attributes, complexType, and simpleTypes. The edges represent nesting or referencing of elements. And the labels denote the min and max cardinality allowed for an element.

2.4. The ad-hoc models

According to [34] the use of ad-hoc models serves to compensate for the designers' the lack of familiarity. In fact, they achieve better notational economy; also, they give proper emphasis to the multidimensional model, and finally, they facilitates to the non-expert user the intuitively and the readability.

In the following, we present some works use the ad-hoc models to ensure the conceptual DW design.

The authors [8] present a conceptual DW design method that is in-line with traditional DataBase design. In fact, the conceptual design is considered as the important phase; it serves to sort out dimensions, corresponding dimension hierarchies, and measures and it has to determine which attribute from underlying databases.

The aim of this phase is to produce a graphical multidimensional schema which for each measure expresses its multidimensional context in terms of relevant dimensions and their hierarchies. So, in the output, we get tables such as the extract concerning account information (which contains an informal description for each relevant attribute and indicates whether the attribute may be used as measure or dimensional attribute and whether the attribute is optional or not) and standard multidimensional queries.

The process phase of conceptual DW design is subdivided into three sequential phases:

- Context definition of measures: it starts by determining functional dependencies (FDS) from dimensional level to measure.

- Dimensional hierarchy design: we gradually develop the dimension hierarchies for each dimension. To this end, we determine all FDs between dimension levels belonging to a dimension dim with terminal dimension level.
- Definition of summarizability constraints: the conceptual model should provide means to distinguish meaningful aggregations of measures from meaningless ones, as this information helps analysts in formulating their queries. In particular, the warehouse schema should express explicitly which measure may be aggregated along what dimension hierarchy according to what aggregation function.

In [7], the authors propose the **Multidimensional Aggregation Cube data model** (**MAC**). This model covers the requirement description to provide a highly expressive and intuitive modeling methodology for the information used in multidimensional analysis. The proposed model uses concepts closely to the way that OLAP users perceive the information.

The **Dimensional Fact Model** (**DFM**) as presented in [30] and [31] is a graphical conceptual model for the DWs. It ensures the constitution of the reality basing on the dimensional scheme that is consists of a set of fact schemes. Those latter have as basic elements the facts, dimensions and hierarchies. The DFM is presented as a directed, acyclic and weakly connected graph. It is a quasi-tree (two or more directed path may converge on the same vertex, the root is connected to each other vertex through exactly one path degenerated into a directed tree).

They propose a way to derive the conceptual model of the DW from the existing ER schemas. This methodology is composed by the following steps:

Defining facts: a fact can be represented on the ER scheme either by an entity or by an n-ary relationship between entities, and for each fact:

- Building the attribute tree: each vertex corresponds to an attribute of the scheme, the root corresponds to the identifier of fact, and for each vertex v the corresponding attribute functionally determines all the attributes corresponding to the descendants of v.

- Pruning and grafting the attribute tree: they are important in case where there are attributes not interesting for the DW. They serve to eliminate the unnecessary levels of detail. Concerning the pruning, it carries out by dropping any sub-tree from the tree, and for grafting, it is used when, though a vertex of the tree expresses uninteresting information, its descendants must be preserved.
- Defining dimensions: dimensions determine how fact instances may be aggregated significantly for the decision-making process. They must be chosen in the attribute tree among the children vertices of the root. They may correspond either to discrete attributes, or to ranges of discrete or continuous attributes.
- Defining fact attributes: the fact attributes are typically either counts of the number of instances of a scheme, or the sum/average/maximum/minimum of expressions involving numerical attributes of the attribute tree with the exclusion of the attributes chosen as dimension. Their way of calculation must be indicated in the logical design phase.
- Defining hierarchies: it is the last step, it arranges, along each hierarchy, the attributes into a tree such that to-one relationships holds between each node and its descendants. In this stage, the pruning and the grafting are possible to eliminate irrelevant details. It is also possible to add new levels of aggregation by defining ranges for numerical attributes.

3. The logical design of DW

According to [29] and [28], the main emphasis of DW modeling is the logical and physical phases. The logical design is considered as a process that starts with a source schema and ends with a final schema that corresponds to the DW schema. This latter is constructed by the application of primitives to the source schema relations [4].

The logical design of the DW serves to define the structures to ensure an efficient access to information. It can be presented as relational or multidimensional structure that takes as input the conceptual schema representation, the information requirements, the source databases, and non functional requirements [46]. Different works has focused on the logical design issues such as data models, data structures specifically designed for DWs, and criteria for defining table partitions and indexes [46].

In the next, we present in first part the different relational models that can be used as logical schema, and in the second part, we present the multidimensional model.

3.1. Dimensional modeling concepts

The dimensional model has two objects namely the production of database structures that make easier writing queries, and the maximization of the efficiency of queries. So, to ensure the achievement of those two objects, [13] propose the use of minimal number of tables and relationships that exist between them, by this way, we can reduce the complexity of the database and minimize the number of joins required in user queries.

The data is presented using ER model, and we find then the following schemas: flat, terraced, star, snowflake, Starflake, and star cluster.

3.1.1. Flat schema

The flat schema is considered according to [13] as the simplest schema that keep all the information. It is formed by collapsing all the entities existing in the data model down into the minimal entities. It serves then to minimize the number of the used tables and then minimizes the joins that will be needed in user queries. In this kind of schema, we get at the end one table for each minimal entity in the original data model. Such structure serves to keep all the information existing in the original data model so it contains redundancy as transitive and partial dependencies but it does not involve the aggregation.

3.1.2. Terraced schema

The terraced schema as presented in [13] is formed by collapsing entities down maximal hierarchies. This process stops when reach a transaction entity. The terraced schema contains at the end a single table for each transaction entity, so it separates explicitly between levels of transactions entities, so it reduces the causes of problems especially for the inexperienced user

3.1.3. Star schema

The star schema is generally credited to Ralph Kimball. It was developed in the early 1980s. This type of schema serves to eliminate the large number of paths so then reducing the number of indexes that are needed to support the DW [22]. It is the most common modeling paradigm since it provides reasonable approach for the ad-hoc and user data access [24].

The star schema is composed by a large center table called fact table that contains the data (without redundancy) and a set of smaller attendant tables called dimension tables. Each one of the dimensions is represented by one table that contains a set of attributes [24].

According to [13] the derivation of the star schema from ER model is easy, and formed by the following steps:

- For the fact table, it corresponds to each transaction entity. It contains a key that corresponds to the combination of the associated components entities.
- For the dimension table, it corresponds to each component entity by collapsing hierarchically related classification entities into it.
- For the hierarchies that connected transactions entities, the child in this case inherits all dimensions, including key attributes from the parent entity. This relationship provides the ability to “drill down” between the transaction levels.
- For the numerical attributes that exit in the transaction entities, they should be aggregated by key attributes.

3.1.3.1. Constellation schema

The constellation schema is used in the case where the sophisticated application required fact tables to share dimension tables. This schema is considered then as a collection of star schemas with hierarchical linked fact tables. This links provide the ability to drill down between levels of detail [13] [24].

3.1.3.2. Galaxy schema

The galaxy schema is considered as a combination of star schemas or even constellations. So, it has a collection of star schemas having as common the dimensions. Concerning the fact tables existing in the galaxy schema, they are not need to be directly related (as it is the case in the constellation schema) [13].

3.1.4. Snowflake schema

According to [13] and [24], the snowflake schema is a variant of the star schema with the normalization of some dimensions. It contains multiple independent hierarchies. This kind of schema is easy to maintain, it saves also the space of the storage because a large dimension table can become enormous when the dimensional structure is included as columns.

The snowflake schema is not popular as the star schema and this is because:

- The saving of space is negligible compared to the typical magnitude of the fact table.
- It needs more joins to execute the queries, which causes the reduction of the effectiveness of browsing (the performance may be adversely impacted).

As the star schema, the snowflake can be derived from the ER model as follow:

- For the fact table, it corresponds to the transaction entity. The key is a combination of the keys of the associated component entities.
- For the dimension table, each one corresponds to the component entity.
- For the hierarchical relationships, they exist between transaction entities so the child entity inherits all relationships to component entities also the key attributes from the parent entity.
- For the numerical attributes, they exist in the transaction entities. They should be aggregated by the key attributes.

3.1.5. Starflake schema

In [14], the starflake schema presents a compromise between the star schema and the snowflake schema. It presents a balanced between two extremes since the star schema is a dimensional model with fully denormalized hierarchies and the snowflake is a dimensional model with fully normalized hierarchies.

Concerning the dimensional models, the overlap between dimensions is undesirable since it increases the complexity of load process and in case of the inconsistent hierarchies it can lead to inconsistent query result (the potential overlap between dimensions corresponds in the ER model to the branch entity which is a classification entity with multiple one-to-many relationships). The starflake schema removes the overlapping since it is selectively normalized.

The hierarchical segments in this kind of schema are separated out into sub-dimension tables. These represent “highest common factors” between dimensions.

The construction of this schema is made as follow:

- Collapsing classification entities from the top of each hierarchy until they reach either a branch entity or a component entity.
- If a branch entity is reached, a sub-dimension table is formed
- Collapsing begins after the branch entity
- If the component entity is reached, a dimension table is formed.
- The design of the fact table is done as for the star schema.

3.1.6. Star cluster schema

The authors in [13] suggest identifying the overlapping dimensions using “forks” in hierarchies. The fork is presented when an entity acts as a parent in two different dimensional hierarchies, so the entity and all of its ancestors are collapsed into two separate dimension table.

The star cluster schema is defined as a schema having the minimal number of tables while avoiding overlap between dimensions. It can be produced from the ER model as follow:

- For a fact table, it corresponds to each transaction entity. The key is the combination f the keys of the associated component entities.

- For the classification entities, they must be collapsed down their hierarchies until they reach either a fork entity or a component entity. If the fork is reached, a sub-dimension table should be formed. The sub-dimension table, in this case, consists of the fork entity and its entire ancestor. The collapsing should begin again after the fork entity. If the component entity is reach, a dimension table must be formed.
- For the hierarchical relationships that exist between transaction entities, the child inherits all dimensions and the key attributes also.
- For the numerical attributes that exist in the transaction entities, they should be aggregate by the key attributes.

Set of star cluster schema can be combined together to form constellations or galaxies.

3.2. Comparative study

We present in this section comparative study of different relational models, presenting their advantages and drawbacks in the table.1.

	Advantages	Drawbacks
Flat schema	- It is the simplest schema [34] - It minimizes the number of tables [34] - It minimizes the joins in the queries[34]	- It contains redundancy (as transitive partial dependencies)[34] - It may lead to aggregation errors [34] - It contains large number of attributes [34
Star schema	- It is the simplest structure [14]. - It reduces the number of tables[9] - It reduces the number of relationships between the tables [9]. - It reduces the number of joins required in user queries [9]. - It speed up query performance [9]	- It can be very inflexible [22] - For every gigabyte of raw data, a S schema will require at least an additiona gigabytes for aggregations [22]. - The amount of development maintenance effort needed to manage a S oriented data warehouse [22].

		- The difficulty of doing cross-functio analysis [22]. - It has the highest level of data redunda [14].
Constellation schema	- It reuses the dimension tables to save storage space [36].	- It may not be useful for small organizati because of its complexity [36].
Snowflake schema	- It shows explicitly the hierarchical structures of each dimension [22]. - It is intuitive and easy to understand [35] - It can accommodate for aggregate data [35] - It is easily extensible by adding new attributes without interfering with existing database programs [35].	- It adds unnecessary complexity [22]. - It reduces query performance [22].
StarFlake schema	- It eliminates the redundancy between dimensions [22]. - It reduces the inconsistency between dimensions [22].	- It has a slightly more complex structure t star schema [22]. - It has redundancy within each table [22].

Table 1. The comparison between DW schemas in term of advantages and drawbacks

3.3. The multidimensional model

The traditional relational data models are not powerful enough to deal with DW applications. Many authors propose as solution the use of data cubes that provide the functionality needed for summarizing, viewing and consolidating the data existing in DW [41]. This kind of structure offers various benefits namely:

- It is close to the way of thinking of data analyzers; therefore, it helps users to understand data [41].

- It supports performance improvement as its simple structure allows designers to predict the user intentions [43]
- It facilitates understanding and writing queries through producing database structures [13].
- It maximizes the efficiency of queries, it reduces the number of tables and the relationships between them, it reduces the complexity of databases, and it minimizes the number of joins required in user queries [13].

The cube (or hypercube) corresponds to events existing in the business domain [43], it represents the data in a multidimensional space. The cube is composed by dimensions; each one has an associated hierarchy of levels of consolidated data. The measures correspond to columns in a relational database table whose values functionally depend on the values of other columns. A value in a single cell may represent an aggregation measure computed from more specific data at some lower level of the same dimension that comprises a set of aggregation level [17].

4. Conclusion

This work is a survey that focuses on two main points, the conceptual DW schema, and the logical DW schema.

The choice of presenting the conceptual is because it is considered as a key step that ensures the successful of the DW projects, since it gives closer ideas about the application domain and its result is a graphical notation that facilitates the task for the designer and the user to write, understand and manage the conceptual schemata

In its side, the logical design is an emphasis of DW modeling, it takes as input a schema, the information requirements, the source databases and non-functional requirements to give as output a final schema that corresponds to the DW schema.

5. References

[1] A. Abelló, J. Samos, F. Saltor, "Benefits of an Object-Oriented Multidimensional Data Model", In Proceedings of the 14th European Conference on Object-Oriented Programming (ECOOP'00), pp.141-152, 2000

[2] A. Abellò, J. Samos, F. Saltor, "YAM2 (Yet another multidimensional model): An extension of UML", International Database Engineering & Applications Symposium (IDEAS 2002). Edmonton, Canada, IEEE Computer Society, pp.172-18, 2002.

[3] A. Abellò, J. Samos, F. Saltor, "YAM2 : a multidimensional conceptual model extending UML", Information Systems (IS), Elsevier, pp.541–567, 2006.

[4] A. Gutiérrez, A. Marotta, "An Overview of Data Warehouse Design Approaches and Techniques", VLDB, Reporte Técnico INCO-01-09. InCo - Pedeciba, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 2000.

[5] A. Nabli, J. Feki, F. Gargouri, "An Ontology Based Method for Normalisation of Multidimensional Terminology", Advanced Internet Based Systems and Applications, Second International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2006, Tunisia, pp.235-246, 2006.

[6] A.S. Kamble, "A conceptual model for multidimensional data", In Proceedings of the fifth on Asia-Pacific conference on conceptual modeling, Wollongong, NSW, Australia, pp.29-38, 2008.

[7] A. Tsois, N. Karayannidis, T. Sellis, "MAC: Conceptual data modelling for OLAP", In Proceedings of the InternationalWorkshop on Design and Management of Data Warehouses (DMDW-2001)', pp. 5–1, 5–13, 2001.

[8] B. Hüsemann, J. Lechtenbörger, G. Vossen, "Conceptual data warehouse design", In Proceedings International Workshop on Design

and Management of Data Warehouses, Stockholm, Sweden, pp.3-9, 2000.

[9] B. P. Başaran, “A Comparison Of Data Warehouse Design Models”, A MASTER'S THESIS in. Computer Engineering, 2005.

[10] C. Ballard, D. Herreman, D. Schau, R. Bell, E. Kim, A. Valencic, “Data Modeling Techniques for Data
Warehousing”, IBM Redbooks publication, California 1998.

[11] C. Imhoff, N. Galemmo, J.G. Geiger, “Mastering Data Warehouse Design”, Wiley Publishing .Inc, Indianapolis, Indiana, 2003.

[12] C. Sapia, M. H. Blaschka, G. Fling, B. Dinter, “Extending the E/R Model for the Multidimensional Paradigm”, In Proceeding of the Intrenational Workshop on Data Warehousing and Data Mining, Singapore, pp.105-116, 1998.

[13] D. L. Moody, M. A. R. Kortink, “From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design”, In Proceedings of International Workshop on Design and Management of Data Warehouses (DMDW'2000), Sweden, pp: 5-1,5-12, 2000.

[14] D. L. Moody, M. A. R. Kortink, “From ER Models to Dimensional Models Part II: Advanced Design Issues”, Journal of Business Intelligence, pp.1-12, 2008.

[15] D. N. Xuan, L. Bellatreche, G. Pierra, “A Versioning Management Model for Ontology-Based Data Warehouses”, In Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery DaWaK 2006, Krakow, Poland, pp.195-206, 2006.

[16] D. Skoutas, A. Simitsis, “Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data”, International Journal of Semantic Web and Information Systems, 3(4): 1-24, 2007.

[17] E. Franconi, U. Sattler, “A data warehouse conceptual data model for multidimensional aggregation”, In Proceedings of International Workshop Design and Management of Data Warehouses (DMDW’ 99), Heidelberg, germany, 1999.

[18] E. Malinowski, E. Zimányi, “Representing spatiality in a conceptual multidimensional model”, In Proceedings of ACM international workshop on Geographic information systems, ACM Press, New York, pp. 12-22, 2004.

[19] E. Malinowski, E. Zimányi, “Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER model”, In Proceedings of the 22nd British National Conference on Databases, BNCOD22, number 3567 in Lecture Notes in Computer Science, Springer-Verlag, Sunderland, UK, pp:17-28, 2005.

[20] E. Malinowski, E. Zimányi, “Hierarchies in a Multidimensional Model: From Conceptual Modeling to Logical Representation”, Data & Knowledge Engineering, 59(2):348-377, 2006.

[21] E. Malinowski, E. Zimany, “Advanced Data Warehouse Design, From Conventional to Spatial and Temporal Applications”, Springer Verlag Berlin Heidelberg, 2008.

[22] F. Teklitz, “The Simplification of Data Warehouse Design”, Sybase, 2000.

[23] H.J. Lenz, A. Shoshani, “Summarizability in olap and statistical data bases”, In Proceedings of the Ninth International Conference on Scientic and Statistical Database Management (SSDBM '97), pp.132-143, 1997.

[24] J. Han, M. Kamber, “Data Mining: Concepts and Techniques, Chapter2: Data Warehouse and OLAP Technology for Data Mining”, Barnes & Nobles, 2000

[25] J.Trujillo, M. Palomar, “An Object Oriented Approach to Multidimensional Database Conceptual Modeling (OOMD), Conference on Information and Knowledge Management”, ACM Press, New York, pp.16-21, 1998.

[26] J. Trujillo, “The GOLD Model: An Object Oriented Multidimensional Data Model For Multidimensional Databases”, In Proceedings of the 9th ECOOP International Workshop For PhD Students In Objects Oriented Systems, Lisboa, Portugal, 1999.

[27] J. Trujillo, M. Palomar, J. Gomez, I.Y. Song, “Designing Data Warehouses with OO Conceptual Models”, IEEE Computer, 34(12): 66-75, 2001.

[28] L. Cabibbo, R. Torlone, “A Logical Approach to Multidimensional Databases”, In Proceedings of the 6th International Conference on Extending Database Technology, (EDBT'98), Spain, pp.183-197, 1998.

[29] M. Boehnlein, A.U.Ende, “Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems”, In Proceedings of the ACM Second International Workshop on Data Warehousing and OLAP (DOLAP'1999, Kansas City, 6. November), pp.15-21, 1999.

[30] M. Golfarelli, D. Maio, S. Rizzi, The Dimensional Fact Model: a Conceptual Model for Data Warehouses”, International Journal of Cooperative Information Systems, pp. 215-247, 1998.

[31] M. Golfarelli, D. Maio, S. Rizzi, “Conceptual design of data warehouses from E/R schemes”, In Proceedings of the Thirty-First Hawaii International Conference on System Sciences, pp.334 -343, 1998.

[32] M. Golfarelli, S. Rizzi, “A Methodological Framework for Data Warehouse Design”, In Proceedings ACM First International Workshop on Data Warehousing and OLAP (DOLAP 98), Washington, D.C., USA, pp.3-9, 1998.

[33] M. Golfarelli, S. Rizzi, “A comprehensive approach to data warehouse testing”, In Proceedings of 12th International Workshop on Data Warehousing and OLAP (DOLAP 2009), Hong Kong, pp.17-24, 2009.

[34] M. Golfarelli, “From User Requirements to Conceptual Design in Data Warehouse Design”, In Data Warehousing Design and Advanced Engineering Applications Methods for Complex Construction, 2009.

[35] M. Hahne, “Logische Datenmodellierung für das Data Warehouse”, In Chamoni, P.;Gluchowski, P. (editors): Analytische Informationssysteme, Springer, Berlin, pp.104-122, 1998.

[36] M. Levene, G. Loizou, “Why is the Snowflake Schema a Good Data Warehouse Design?”, In Source, Information Systems, pp. 225-240, 2003.

[37] M. Mhiri, “Méthodologie de construction des ontologies pour la résolution de conflits de Systèmes d’Information”, Revue Technique et Science Informatiques, Lavoisier, Paris, France, 2009.

[38] N. Tryfona, F. Busborg, J.G.B. Christiansen, “starER: A Conceptual Model for Data Warehouse Design”, In Proceedings of the ACM 2nd International Workshop Data Warehousing and OLAP (DOLAP99), Kansas City, USA, 1999, pp.3-8.

[39] O. Romero, A. Abelló, “Automating Multidimensional Design from Ontologies”, In Proceedings of the 10th International Workshop on Data Warehousing and OLAP, pp.1-8, 2007.

[40] P. Bizarro, H. Madeira, “Adding a Performance-Oriented Perspective to Data Warehouse Design”, In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery DaWaK, pp.232-244, 2002.

[41] P. Vassiliadis, T. Sellis, “A Survey on Logical Models for OLAP Databases”, In Proceedings of SIGMOD Record, pp. 64-69, 1999.

[42] R. Torlone, “Conceptual Multidimensional Models”, Multidimensional Databases, pp.69-90, 2003.

[43] S. Rizzi, “Conceptual Modeling Solutions for the Data Warehouse”, Database Technologies: Concepts, Methodologies, Tools, and Applications, pp.86-104, 2009.

[44] S.L. Mora, J.Trujillo, “A comprehensive method for data warehouse design”, In Proceedings of the 5th International Workshop on Design and Management of Data Warehouses, Germany, (1).1-14, 2003.

[45] S.L. Mora, J. Trujillo, I.Y. Song, “A UML profile for multidimensional modeling in data warehouses”, Data & Knowledge Engineering, pp.725-769, 2006.

[46] V. Peralta, A. Illarze, R. Ruggia, “On the Applicability of Rules to Automate Data Warehouse Logical Design”, In Proceedings of the 15th Conference on Advanced Information Systems Engineering Klagenfurt, Velden, Austria, pp.329-340, 2003.

[47] W. Eckerson, “Four Ways to Build a Data Warehouse”, http://www.tdan.com/view-articles/4770, 2007.