Integrations of Data Warehousing, Data Mining and Database Technologies:

Innovative Approaches

David Taniar
Monash University, Australia

Li Chen
LaTrobe University, Australia

Information Science
REFERENCE

Senior Editorial Director: Kristin Klinger
Director of Book Publications: Julia Mosemann
Editorial Director: Lindsay Johnston
Acquisitions Editor: Erika Carter
Development Editor: Michael Killian
Production Coordinator: Jamie Snavely
Typesetters: Milan Vracarich Jr., Jennifer Romanchak & Michael Brehm
Cover Design: Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

Library of Congress Cataloging-in-Publication Data

Integrations of data warehousing, data mining and database technologies : innovative approaches / David Taniar and Li Chen, editors.
p. cm.
Includes bibliographical references and index.
Summary: “This book provides a comprehensive compilation of knowledge covering state-of-the-art developments and research, as well as current innovative activities in data warehousing and mining, focusing on the integration between the fields of data warehousing and data mining, with emphasis on the applicability to real world problems”--Provided by publisher.
ISBN 978-1-60960-537-7 (hardcover) -- ISBN 978-1-60960-538-4 (ebook) 1. Data warehousing. 2. Data mining. I. Taniar, David. II. Chen, Li, 1982-
QA76.9.D37I4577 2011
006.3’12--dc22

2011009996

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 5
Multidimensional Design Methods for Data Warehousing

Oscar Romero
Universitat Politècnica de Catalunya, Spain

Alberto Abelló
Universitat Politècnica de Catalunya, Spain

ABSTRACT

In the last years, data warehousing systems have gained relevance to support decision making within organizations. The core component of these systems is the data warehouse and nowadays it is widely assumed that the data warehouse design must follow the multidimensional paradigm. Thus, many methods have been presented to support the multidimensional design of the data warehouse.The first methods introduced were requirement-driven but the semantics of the data warehouse (since the data warehouse is the result of homogenizing and integrating relevant data of the organization in a single, detailed view of the organization business) require to also consider the data sources during the design process. Considering the data sources gave rise to several data-driven methods that automate the data warehouse design process, mainly, from relational data sources. Currently, research on multidimensional modeling is still a hot topic and we have two main research lines. On the one hand, new hybrid automatic methods have been introduced proposing to combine data-driven and requirement-driven approaches. These methods focus on automating the whole process and improving the feedback retrieved by each approach to produce better results. On the other hand, some new approaches focus on considering alternative scenarios than relational sources. These methods also consider (semi)-structured data sources, such as ontologies or XML, that have gained relevance in the last years. Thus, they introduce innovative solutions for overcoming the heterogeneity of the data sources. All in all, we discuss the current scenario of multidimensional modeling by carrying out a survey of multidimensional design methods. We present the most relevant methods introduced in the literature and a detailed comparison showing the main features of each approach.

DOI: 10.4018/978-1-60960-537-7.ch005

INTRODUCTION

Data warehousing systems were conceived to support decision making within organizations. These systems homogenize and integrate data of organizations in a huge repository of data (the data warehouse) in order to exploit this single and detailed representation of the organization and extract relevant knowledge for the organization decision making. The data warehouse is a huge repository of data that does not tell us much by itself; like in the operational databases, we need auxiliary tools to query and analyze data stored. Without the appropriate exploitation tools, we will not be able to extract valuable knowledge of the organization from the data warehouse, and the whole system will fail in its aim of providing information for giving support to decision making. OLAP (On-line Analytical Processing) tools were introduced to ease information analysis and navigation all through the data warehouse in order to extract relevant knowledge of the organization. This term was coined by E.F. Codd in (Codd, 1993), but it was more precisely defined by means of the FASMI test that stands for *fast analysis* of *shared* business *information* from a *multidimensional* point of view. This last feature is the most important one since OLAP tools are conceived to exploit the data warehouse for analysis tasks based on *multidimensionality*.

The multidimensional conceptual view of data is distinguished by the *fact* / *dimension* dichotomy, and it is characterized by representing data as if placed in an n-dimensional space, allowing us to easily understand and analyze data in terms of facts (the subjects of analysis) and dimensions showing the different points of view where a subject can be analyzed from. One fact and several dimensions to analyze it produce what is known as *data cube*. Multidimensionality provides a friendly, easy-to-understand and intuitive visualization of data for non-expert end-users. These characteristics are desirable since OLAP tools are aimed to enable analysts, managers, executives, and in general those people involved in decision making, to gain insight into data through fast queries and analytical tasks, allowing them to make better decisions.

Developing a data warehousing system is never an easy job, and raises up some interesting challenges. One of these challenges focus on modeling multidimensionality. Nowadays, despite we still lack a standard multidimensional model, it is widely assumed that the data warehouse design must follow the multidimensional paradigm and it must be derived from the data sources, since a data warehouse is the result of homogenizing and integrating relevant data of the organization in a single and detailed view.

Terminology and Notation

Lots of efforts have been devoted to multidimensional modeling, and several models and methods have been developed and presented in the literature to support the multidimensional design of the data warehouse. However, since we lack a standard multidimensional terminology, terms used to describe the multidimensional concepts may vary among current design methods. To avoid misunderstandings, in this section we detail a specific terminology to establish a common framework where to map and compare current multidimensional design methods.

Multidimensionality is based on the fact/dimension dichotomy. Dimensional concepts produce the multidimensional space in which the fact is placed. Dimensional concepts are those concepts likely to be used as an analytical perspective, which have traditionally been classified as dimensions, levels and descriptors. Thus, we consider that a dimension consists of a hierarchy of levels representing different granularities (or levels of detail) for studying data, and a level containing descriptors (i.e., level attributes). In contrast, a fact contains measures of analysis. One fact and several dimensions for its analysis produce a multidimensional schema. Finally, we denote by base a *minimal* set of levels function-

ally determining the fact. Thus, the base concept guarantees that two different instances of data cannot be placed at the same point of the multidimensional space.

For example, consider Figure 1. There, one fact (sales) containing two measures (price and discount) is depicted. This fact has four different dimensions of analysis (buyer, seller, time and item sold). Two of these dimensions have just one level of detail, whereas the other two have an aggregation hierarchy with more than one level. For example, the time dimension has three levels of detail that, in turn, contain some descriptors (for example, the holiday attribute). Finally, if we consider {item X day X buyer X seller} to be the multidimensional base of sales it would mean that a value of each of these levels identify one, and just one, instance of factual data (i.e., a specific sale and its price and discount).

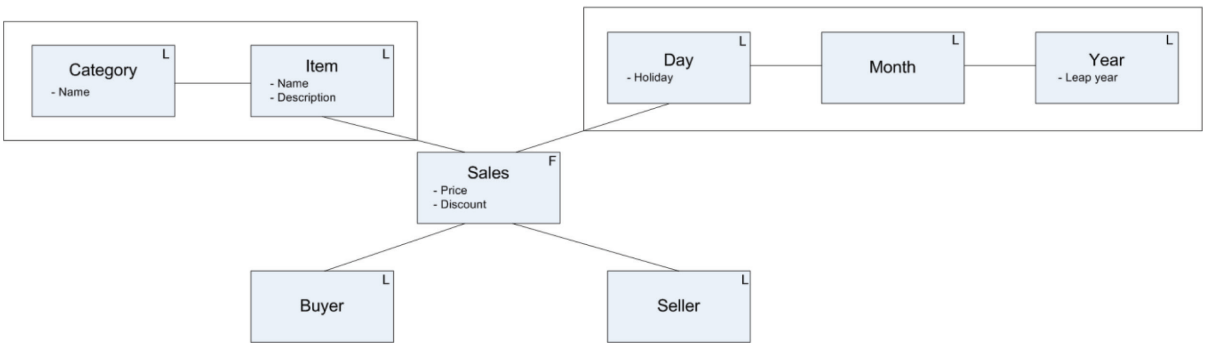
A Piece of History

In this section we introduce the background of multidimensional modeling. Our objective here is to provide an insightful view of how this area evolved with time.

Multidimensional modeling as it is known today was first introduced by Kimball in (Kimball, 1996). Kimball's approach was well received by the community and a deeper and advanced view of multidimensional modeling was presented in (Kimball et al., 1998). In these books the authors introduced the first method to derive the data warehouse logical schema. Like in most information systems, this method is requirement-driven: it starts eliciting business requirements of an organization and through a step-by-step guide we are able to derive the multidimensional schema from them. Only at the end of the process data sources are considered to map data from sources to target.

Shortly after Kimball introduced his ad hoc modeling method for data warehouses, some other methods were presented in the literature. Like Kimball's method, these methods are step-by-step guides to be followed by a data warehouse expert that start gathering the end-user requirements. However, these approaches gave more relevance to the data sources. According to the data warehouse definition, the data warehouse is the result of homogenizing and integrating relevant data of the organization (stored in the organization data sources) in a single and detailed view and consequently, data sources must be considered somehow during the design process. Involving the data sources in these approaches means that it is compulsory to have well-documented data sources (for example, with up-to-date conceptual schemas) at the expert's disposal but it also entailed two main benefits. On the one hand, the user may not know all the potential analysis contained in the data sources and analyzing them we may find unexpected potential analysis of interest for the

Figure 1. Multidimensional concepts



user. On the other hand, we should guarantee that the data warehouse will be able to be populated with data available within the organization.

To carry out the design task manually it is compulsory to have well-documented data sources. However, in a real organization the data sources documentation may be incomplete, incorrect or may not even exist and, in any case, it would be rather difficult for a non-expert designer to follow these guidelines. Indeed, when automating this process is essential not to depend on the expert's ability to properly apply the method chosen and to avoid the tedious and time-consuming task (even unfeasible when working over large databases) of analyzing the data sources.

In order to solve these problems several methods automating the data warehouse design were introduced in the literature. These approaches work directly over relational (i.e., logical) database schemas. Thus, despite they are restricted to relational data sources, they get up-to-date data that can be queried and managed by computers. Furthermore, they argued that restricting to relational technology made sense, since it was / is de facto standard for operational databases. About the process carried out, these methods follow a data-driven process focusing on a thorough analysis of the data sources to derive the data warehouse schema in a reengineering process. This reengineering process consists of techniques and design patterns that must be applied over the relational schema of the data sources to identify data likely to be analyzed from a multidimensional perspective.

Nevertheless, a requirement analysis phase is crucial to meet the user needs and expectations. Otherwise, the user may find himself frustrated since he / she would not be able to analyze data of his / her interest, entailing the failure of the whole system. Today, it is assumed that the ideal scenario to derive the data warehouse conceptual schema would entail a hybrid approach (i.e., a combined data-driven and requirement-driven approach). Therefore, the resulting multidimensional schema would satisfy the end-user requirements and it would have been conciliated with the data sources simultaneously (i.e., capturing the analysis potential depicted in the data sources and being able to be populated with data within the organization).

Another interesting trend worth to remark, is the level of abstraction used for the methods' output. The first methods introduced (such as Kimball's method) produced multidimensional star schemas (i.e., logical schemas), but soon the community realized it was as important as in any other system to differentiate the conceptual and logical layer in the data warehouse design task (for example, it was obvious when MOLAP tools gained relevance regarding ROLAP ones). As result, newest approaches generate conceptual schemas and it is up to the user to implement them with any of the logical design (either relational or ad hoc multidimensional) alternatives. The fact that logical design was first addressed in data warehousing gave rise to a spread language abuse when referring to multidimensional conceptual schemas, which are also denoted as star schemas. Originally, star schemas were logical design models introduced in (Kimball et al., 1998). The reason is that multidimensional conceptual schemas also are *star-shaped* (with the fact in the center and the dimensions around it), and the star schema nomenclature was reused for conceptual design.

About the last research efforts in this area, we can identify two main addressed issues. On one hand, most recent works aim to automate the process considering both the organization data sources and the user multidimensional requirements. However, automating the requirement management is not an easy job, as it is compulsory to formalize the end-user requirements (i.e., translate them to a language understandable by computers) and nowadays, most of the current methods handle requirements stated in languages (such as natural language) lacking any kind of formalization. On the other hand, some new approaches have been introduced to automate the multidimensional design of data warehouses from other sources that have gained relevance in the last years, such as the

semantic web (Berners-Lee, Hendler & Lassila, 2001). In the recent past, the multidimensional analysis of data has been restricted to the well structured information sources within the company (which mainly were relational). Nevertheless, (Inmon, Strauss & Neushloss, 2008) outlines the opportunity and importance of using unstructured and semi-structured data (either textual or not) in the decision making process. These data could still come from the sources in the company, but also from the web. Accordingly, some new approaches consider data sources based on web-related technologies such as ontologies or XML.

A COMPREHENSIVE SURVEY

This section presents an insight into current multidimensional design methods. Methods here discussed were selected according to three factors: reference papers with a high number of citations according to Google Scholar (Google, 2008) and Publish or Perish (Harzing, 2010), papers with novelty contributions and in case of papers of the same authors, we have included the latest version of their works. As general rule, each method is described using the terminology presented in the *Terminology and Notation* section. Finally, we follow a chronological order when introducing the design methods surveyed. As an exception, when a method publications span over several different papers, we place them at the chronological point occupied by their first paper but we quote them by means of the most relevant paper.

All in all, this section provides a comprehensive framework of the evolution of multidimensional design methods.

(Kimball et al., 1998) introduced multidimensional modeling as known today. In addition, they also introduced the first method to produce the multidimensional schema. Being the first approach, it does not introduce a formal design procedure, but a detailed guide of tips to identify the multidimensional concepts and then, give rise to the multidimensional schema. The presentation is quite informal and it relies on examples rather than on formal rules. Kimball's approach follows a demand-driven framework to derive the data warehouse relational schema (i.e., logical), as follows.

First, the designer must identify all the data marts we could possibly build. Data marts are essentially defined as pragmatic collections of related facts. Although data sources are not considered, they already suggest to take a look at the data sources to find which data marts may be of our interest.

Next step aims to list all conceivable dimensions for each data mart. At this point it is suggested to build an ad hoc matrix to capture our multidimensional requirements. Rows represent the data marts, whereas columns represent the dimensions. A given cell is marked whether that dimension must be considered for a data mart.

This matrix is also used to show the associations between data marts by looking at dimensions shared. This process is supposed to be incremental. First, it is suggested to focus on single-source data marts, since it will facilitate our work and later, in a second iteration, look for multiple-sources data marts combining the single-source designs.

The method's third step designs the fact tables of each data mart:

- First, we must declare the *grain* of detail (i.e., the data granularity of interest). It is suggested to be defined by the design team at the beginning, although it can be reconsidered during the process. Normally, it must be determined by primary dimensions.
- Next, we choose the analysis dimensions for each fact table. Dimensions selected must be tested against the grain selected. This must be a creative step. We need to look for the dimension *pieces* (i.e., levels and descriptors) in different (and potentially heterogeneous) models and through different documents, which, in the end, results

in a time-consuming task. At this point, it is also suggested to choose a large number of descriptors to populate dimensions.
- Finally, the last stage adds as many measures as possible within the context of the declared grain.

(Cabibbo & Torlone, 1998) present one of the most cited multidimensional design methods. This approach generates a logical schema from *Entity-Relationship* (ER) diagrams, and it might produce multidimensional schemas in terms of relational databases or multidimensional arrays.

At first sight, this method may be thought to follow a supply-driven paradigm, as it performs an in-depth analysis of the data sources. However, no formal rules to identify the multidimensional concepts from the data sources are given. In fact, multidimensional concepts must be manually identified by the user (i.e., from requirements).

For this reason, we consider it to follow a hybrid framework. In general, like Kimball's approach, this approach is rather informal but they set up the foundations that were later used by the rest of methods.

This method consists of four steps. First and second steps aim to identify facts and dimensions and restructure the ER diagram. Both steps may be performed simultaneously and benefit from the feedback retrieved by each step. Indeed the authors suggest to perform them in an iterative way to refine results obtained. However, no clue about how to identify facts, measures and dimensions are given and they must be identified from the end-user requirements. Once identified, each fact is represented as an entity. Next, we add dimensions of interest that may be missing in the schema but could be derived from external sources or metadata associated to our data sources. At this point, it is also compulsory to refine the levels of each dimension by means of the following transformations: (i) replacing many-to-many relationships, (ii) adding new concepts to represent new levels of interest, (iii) selecting a simple identifier for each level entity and (iv) removing irrelevant concepts. Finally, two last steps aim to derive the multidimensional schema. Some clues are given to derive a multidimensional graph that will be directly mapped into the multidimensional schema.

(Golfarelli & Rizzi, 2009) present one of the reference methods in this area. This work present a detailed view of the multidimensional design process proposed, which subsumes their previous works such as (Golfarelli, Maio & Rizzi, 1998a; Golfarelli & Rizzi, 1998b). This approach presents a formal and structured method (partially automatable) that consists of six well-defined steps. However, the fourth step aims to estimate the data warehouse workload which goes beyond the scope of this study:

- First step analyzes the underlying information system and produces a conceptual schema (i.e., a ER diagram) or a logical schema (i.e., a relational schema).
- Second step collects and filters requirements. In this step it is important to identify facts. The authors give some tips to identify them from ER diagrams (entities or n-ary relationships) or relational schemas (tables frequently updated are good candidates).
- Next step derives the multidimensional conceptual schema from requirements and facts identified in previous steps. This step may be carried out semi-automatically as follows:
  - Building the attribute tree: From the primary key of the fact we create a tree by means of functional dependencies. Thus, a given node (i.e., an attribute) of the tree functionally determines its descendants.
  - Pruning and grafting the attribute tree: The tree attribute must be pruned and grafted in order to eliminate unnecessary levels of detail.

  - Defining dimensions: Dimensions must be chosen in the attribute tree among the root vertices.
  - Defining measures: Measures are defined by applying aggregation functions, at root level, to numerical attributes of the tree.
  - Defining hierarchies: The attribute tree shows a plausible organization for hierarchies. Hierarchies must be derived from to-one relationships that hold between each node and its descendants.
- Finally, the last two steps derive the logical (by translating each fact and dimension into one relational table) and physical schemas (the authors give some tips regarding indexes to implement the logical schema in a ROLAP tool).

The fourth step of this method aims to estimate the workload of the data warehouse. The authors argue that this process may be used to validate the conceptual schema produced in the third step, as queries could only be expressed if measures and hierarchies have been properly defined. However, no further information is provided.

(Böehnlein & Ulbrich-vom Ende, 1999) present a hybrid approach to derive logical schemas from SER (Structured Entity Relationship) diagrams. SER is an extension of ER that visualizes existency dependencies between objects. For this reason, the authors argue that SER is a better alternative to identify multidimensional structures. This approach has three main stages:

- Pre-process: First, we must transform the ER diagram into a SER diagram. A detailed explanation is provided.
- Step 1: Business measures must be identified from goals. For example, the authors suggest to look for business events to discover interesting measures. Once business measures have been identified, they are mapped to one or more objects in the SER diagram. Eventually, these measures will give rise to facts.
- Step 2: The hierarchical structure of the SER diagrams is helpful to identify potential aggregation hierarchies. Dimensions and aggregation hierarchies are identified by means of direct and transitive functional dependencies. The authors argue that discovering dimensions is a creative task that must be complemented with a good knowledge of the application domain.
- Step 3: Finally, a star or snowflake schema is derived as follows: each fact table is created by using the set of primary keys of their analysis dimensions as its compound primary key, and denormalizing or normalizing the aggregation hierarchies accordingly.

(Hüsemann, Lechtenbörger & Vossen, 2000) present a requirement-driven method to derive multidimensional schemas in *multidimensional normal form* (MNF). This work introduces a set of restrictions that any multidimensional schema produced by this method will satisfy. Furthermore, although this approach produces conceptual schemas, they also argue that the design process must comprise four sequential phases (requirements elicitation and conceptual, logical and physical design) like any classical database design process:

- Requirement analysis and specification: Despite it is argued that the operational ER schema should deliver basic information to determine the multidimensional analysis potential, no clue about how to identify the multidimensional concepts from the the data sources is given. Business domain experts must select strategically relevant operational database attributes and specify the purpose to use them as dimensions or measures. The resulting requirements specification contains a tabular list of attri-

butes together with their multidimensional purpose, similar to Kimball's proposal. Supplementary informal information may be added such as standard multidimensional queries that the user would like to pose.

- Conceptual design: This step transforms the semi-formal business requirements into a formalized conceptual schema. This process is divided in three sequential stages:
  - Context definition of measures: This approach requires to determine a base for each measure (i.e., a minimal set of dimension levels functionally determining the measure values). Furthermore, measures sharing bases are grouped into the same fact, as they share the same dimensional context.
  - Dimensional hierarchy design: From each atomic level identified, this step gradually develops the dimension hierarchies by means of functional dependencies. Descriptors and levels are distinguished from requirements. In this approach, the authors distinguish between simple and multiple (containing, at least, two different aggregation path) hierarchies. Moreover, specialization of dimensions must be considered to avoid structural NULL values when aggregating data.
  - Definition of summarizability constraints: The authors argue that some aggregations of measures over certain dimensions do not make sense. Therefore, they propose to distinguish meaningful aggregations from meaningless ones and include this information in an appendix of the conceptual schema.

Finally, the authors argue that a multidimensional schema derived by means of this method is in *dimensional normal form* (MNF) (Lehner, Albrecht & Wedekind, 1998) and therefore it fully makes multidimensional sense. Consequently, we can form a data cube (i.e., a multidimensional space) free of summarizability problems. In short, it is achieved by means of five constraints: measures must be fully functionally identified by the multidimensional base, each dimension hierarchy must have an atomic level, each dimension level must be represented by identifier attribute(s), every descriptor must be associated to a dimension level and dimensions generated must be orthogonal. By following their method, all these constraints are guaranteed.

(Moody & Kortink, 2000) present a method to develop multidimensional schemas from ER models. It was one of the first supply-driven approaches introduced in the literature, and one of the most cited papers in this area. Although it is not the first approach working over ER schemas, they present a structured and formal method to derive multidimensional logical schemas. Their method is divided into four steps:

- Pre-process: This step develops the enterprise data model if it does not exist yet.
- First step: This step classifies the ER entities into three main groups:
  - Transactional entities: These entities record details about particular events that occur in the business (orders, sales, etc). They argue that these are the most important entities in a data warehouse and form the basis of fact tables in star schemas, as these are the events that decision makers want to analyze. Although the authors do not consider requirements, they underline the relevance of requirements to identify facts, because not all the transactional entities will be of interest to the user. Moreover, they provide the key features to look for this kind of entities: the entity must describe an event that happens at a point in time, and

it must contain measures or quantities summarizable.
    - Component entities: These entities are directly related to a transaction entity via a one-to-many relationship and they define details or components of each business event. These entities will give rise to dimension tables in star schemas.
    - Classification entities: These entities are related to component entities by a chain of one-to-many relationships. Roughly speaking, they are functionally dependent on a component entity directly or by transitivity. They will represent dimension hierarchies in the multidimensional schema.

The authors assume that a given entity may fit into multiple categories. Therefore, they define a precedence hierarchy for resolving ambiguities: *Transaction* > *Classification* > *Component*. Thus, if an entity may play a transaction entity role, it is not considered neither as a classification nor a component entity. The rest of entities in the ER schema will not be included in the multidimensional schema.

- Second step: Next step aims to shape dimension hierarchies. The authors provide some formal rules to identify them. Specifically, a dimension hierarchy is defined as a sequence of entities joined together by one-to-many relationships all aligned in the same direction. Moreover, they introduce the concept of minimal entity (i.e., atomic level) and maximal entity (i.e., that with a coarser granularity data). Some formal rules to identify minimal and maximal entities are given. For example, minimal entities are those without one-to-many relationships, and maximal are those without many-to-one relationships.
- Third step: Transactional entities will give rise to facts, whereas dimension hierarchies will give rise to their analysis perspectives. The authors introduce two different operators to produce logical schemas:
    - Collapse hierarchy: Higher levels in hierarchies can be collapsed into lower levels. Indeed, the authors propose to denormalize the hierarchies according to our needs, as typically performed in data warehousing to improve query performance.
    - Aggregation: Can be applied to a transaction entity to create a new entity containing summarized data. To do so, some attributes are chosen to be aggregated (i.e., measures) and others to aggregate by (i.e., dimensional concepts).

By these operators, this approach introduces five different dimensional design alternatives. According to the resulting schema level of denormalization and the granularity of data, they introduce rules to derive *flat schemas, terraced schemas, star schemas, snowflake schemas* or *star cluster schemas*. They also introduce the notion of constellation schema that denotes a set of star schemas with hierarchically linked fact tables.

(Bonifati et al., 2001) present a hybrid semi-automatic approach consisting of three basic steps: a demand-driven, a supply-driven and a third stage to conciliate the two first steps (i.e., it introduces a sequential hybrid approach). The final step aims to integrate and conciliate both paradigms and generate a feasible solution that best reflects the user's necessities. This method generates a logical multidimensional schema and it was the first to introduce a formal hybrid approach with a final step conciliating both paradigms. Moreover, this method has been applied and validated in a real case study:

- We start collecting the end-user requirements through interviews and expressing user expectations through the *Goal / Question / Metrics* (GQM) paradigm. GQM is composed of a set of forms and guidelines developed in four stages: (i) a first vague approach to formulate the goals in abstract terms, (ii) a second approach using forms and a detailed guide to identify goals by means of interviews, (iii) a stage to integrate and reduce the number of goals identified by collapsing those with similarities and finally, (iv) a deeper analysis and a detailed description of each goal. Next, the authors present an informal guideline to derive a logical multidimensional schema from requirements. Some clues and tips to identify facts dimensions and measures from the forms and sheets used in this process are given.
- Second step aims to carry out a supply-driven approach from ER diagrams capturing the operational sources. This step may be automated, and it performs an exhaustive analysis of the data-sources. From the ER diagram, a set of graphs that will eventually produce star schemas are created as follows:
  - Potential fact entities are labeled according to the number of additive attributes they have. Each identified fact is taken as the center node of a graph.
  - Dimensions are identified by means of many-to-one and one-to-one relationships from the center node. Moreover, many-to-many relationships are transformed into one-to-many relationships. Finally, each generalization / specialization taxonomy is also included in the graphs.

Next, they introduce an algorithm to derive snowflake schemas from each graph. This transformation is immediate and once done, they transform the snowflake schemas into star schemas by flattening the dimension hierarchies (i.e., denormalizing dimensions).

- Third step aims to integrate star schemas derived from the first step with those identified from the second step. In short, they try to map demand-driven schemas into supply-driven schemas by means of three steps:
  - Terminology analysis: Before integration, demand-driven and supply-driven schemas must be converted to a common terminological idiom. A mapping between GQM and ER concepts must be provided.
  - Schema matching: Supply-driven schemas are compared, one-by-one, to demand-driven schemas. A match occurs if both have the same fact. Some metrics, with regard to the number of measures and dimensions, are calculated.
  - Ranking and selection: Supply-driven schemas are ranked according to the metrics calculated in the previous step and presented to the user.

As final remark, this method does not introduce the concept of descriptor in any moment. However, since they map relational entities into levels, we may consider attributes contained in the entities as the multidimensional descriptors.

(Phipps & Davis, 2002) introduced one of the first methods automating part of the design process. This approach proposes a supply-driven stage to be validated, a posteriori, by a demand-driven stage. It is assumed to work over relational schemas (i.e., at the logical level) and a conceptual multidimensional schema is produced. In this approach, their main objective is the automation of the supply-driven process with two basics premises: numerical fields represent measures and the

more numerical fields a relational table has, the more likely it is to play a fact role. Furthermore, any table related with a to-many relationship is likely to play a relevant dimensional role. In general, they go one step beyond in the formalization of their approach since a detailed pseudo-algorithm is presented in this paper (and therefore, automation is immediate). However, this approach generates too many results and a demand-driven stage is needed to filter results according to the end-user requirements. Thus, the demand-driven stage in this approach is rather different from the rest of demand-driven approaches, because they do not derive the multidimensional schema from requirements but they use requirements to filter results. This method consists of five steps:

- First step finds tables with numerical fields and create a fact node for each table identified. Tables with numerical fields are sorted in descending order of number of numeric fields. Tables will be processed in this order.
- Second step creates measures based on numerical fields within fact tables.
- Third step creates date and / or time dimension levels with any date / time fields per fact node.
- Fourth step creates dimensions (consisting of just one level) for each remaining table attribute that is non-numerical, non-key and non date field. Although this may be considered as a controversial decision (any other attribute would give rise to a dimension of analysis), it was the first method handling partially denormalized data sources.
- Fifth step recursively examines the relationships of the tables to add additional levels in a hierarchical manner. To do so, it looks for many-to-one relationships (according to foreign keys and candidate keys) all over the schema.

The heuristics used to find facts and determine dimensional concepts within a fact table are rather generic, and they generate results containing too much noise. Consequently, the authors propose a final requirement-driven step to filter results obtained. This step presents a step-by-step guide to analyze the end-user requirements expressed as MDX queries and guide the selection of candidate schemas most likely to meet user needs. This last step must be manually performed.

(Winter & Strauch, 2003) present a detailed demand-driven approach. This is a reference paper because it presents a detailed discussion between different multidimensional design paradigms. Furthermore, they present a method developed from the analysis of several data warehouse projects in participating companies. However, their approach is rather different from the rest of methods. They do not assume the multidimensional modeling introduced by Kimball like the rest of methods do, and they present a high-level step-by-step guideline.

In short, they identify the best practices that a data warehouse design project must consider, according to their analysis task. The design process must be iterative and it is divided into four stages:

- First step embraces the analysis of the information supply (i.e., from the sources) and the analysis of the information needed.
- Next, we must match requirements demanded with current information supply and order requirements accordingly.
- In a third step, information supply and information demand must be synchronized on a full level of detail (i.e., considering data granularity selected).
- Finally, we must develop the multidimensional schema. This schema must be evaluated and if needed, reformulate the process from the first step to develop the multidimensional schema in an iterative way.

Although this approach gives relevance to the data sources and demands to synchronize data demanded with the sources, we consider it to be a demand-driven approach since no clue about how to analyze the data sources is given.

(Vrdoljak, Banek & Rizzi, 2003) present a semi-automatic supply-driven approach to derive logical schemas from XML schemas. This approach considers XML schemas as data sources. Therefore, the authors propose to integrate XML data in the data warehouse, as XML is now a *de facto* standard for the exchange of semi-structured data. Their approach works as follows:

- Preprocessing the XML schema: The schema is simplified to avoid complex and redundant specifications of relationships.
- Creating and transforming the schema graph: Every XML schema can be represented as a graph. Two transformations are carried out at this point; functional dependencies are explicitly stated (by means of key attributes) and nodes not storing any value are discarded.
- Choosing facts: Facts must be chosen among all *vertexes* (i.e., nodes) and *arcs* (i.e., edges) of the graph. An arc can be chosen only if it represents a many-to-many relationship.
- Building the dependency graph: For each fact, a dependency graph is built. The graphical representation of the XML schema facilitates finding the functional dependencies. The graph must be examined in the direction expressed by arcs and according to cardinalities included in the dependency graph. It may happen that no cardinality is provided. In this case, XML documents are queried by means of XQueries to look for to-one relationships. The authors also consider many-to-many relationships to be of interest in some cases. However, these cases must be manually identified by the user. Finally, the dependency graph will give rise to aggregation hierarchies.
- Creating the logical schema: Facts and measures are directly depicted from vertexes and arcs chosen, whereas dimensions are derived from the aggregation hierarchies identified.

(Jensen, Holmgren & Pedersen, 2004) present a supply-driven method from relational databases. They present data-mining techniques to be applied over the intensional data to discover functional and inclusion dependencies and, eventually, derive snowflake schemas.

Their method starts collecting metadata such as table and attribute names, cardinality of attributes, frequency, etc. Later, data is divided into three groups according to its potential multidimensional role: measure, keys and descriptive data. Next, integrity constraints such as functional and inclusion dependencies are identified between attributes and finally, the snowflake schema is produced.

First two steps are performed consulting the database catalog. The role of each attribute is derived with a bayesian network that takes as input metadata collected for each attribute. The third step discovers the database structure by identifying functional and inclusion dependencies that represent many-to-one relationships that will give rise to dimensions. Candidate keys and foreign keys are identified assuming that there are no composite keys in the database. Furthermore, inclusion dependencies among foreign keys and candidate keys are identified in this step. These dependencies will be mainly used to identify dimensions. This step is critical, since all permutations of candidate keys and foreign keys are constructed with the consequent computational cost. To pair two keys, both must have the same attribute type and the candidate key must have, at least, as many distinct values for the attribute as the table containing the foreign key. If these constraints hold, a SQL statement is issued to check if the join of both tables (by means of these

attributes) has the same cardinality as the table containing the candidate foreign key. If so, an inclusion dependency is identified between both keys. Next, they propose an algorithm to derive snowflake schema from this metadata:

- Fact tables are identified in a semi-automatic process involving the user. First, facts are proposed by means of the table cardinality and the number of measures identified by a bayesian network. Then, the user chooses those of his / her interest.
- Inclusion dependencies discovered form different connected graphs. A connected graph is considered to be a dimension if exists a inclusion dependency between a fact table and a graph node. In this case, that node will play the atomic level role of the dimension. The authors propose an algorithm to break potential cycles and give rise to the aggregation hierarchy from the graph. When shaping the aggregation hierarchy, two consecutive levels are analyzed to avoid aggregation problems (i.e., duplicated or lost values).

(Giorgini, Rizzi & Garzetti, 2005) present a hybrid approach to derive the conceptual multidimensional schema. They propose to gather multidimensional requirements and later map them onto the data sources in a conciliation process. However, they also suggest that their approach could be considered a pure demand-driven if the user do not want to consider the data sources. The authors introduce an agent-oriented method based on the *i** framework. They argue that it is important to model the organization setting in which the data warehouse will operate (organization modeling) and capture the functional and non-functional requirements of the data warehouse (what authors call the decisional modeling). If we consider their hybrid approach, the next step is to match requirements with the schema of the operational sources. In this approach both ER diagrams and relational schemas are allowed as inputs describing the data sources. This matching stage consists of three steps:

- Requirement mapping: Facts, dimensions and measures identified during the requirement analysis are now mapped over the data sources. According to the kind of data sources considered, the authors introduce a set of hints to map each concept. For example, facts are mapped onto entities or n-ary associations in ER diagrams and onto relations in relational schemas.
- Hierarchy construction: For each fact identified, the data sources are analyzed looking for functional dependencies based on the algorithm already discussed in (Golfarelli & Rizzi, 2009).
- Refinement: This step aims to rearrange the fact schema in order to better fit the user's needs. In this process, we may distinguish among concepts available (mapped from requirements), unavailable (demanded in the requirements but not mappable to the data sources) and what is available and not needed. The authors propose to use this information to reorder dimensions (grafting and pruning the aggregation hierarchies) and / or try to find new directions of analysis.

(Annoni et al., 2006) present a demand-driven approach to derive a multidimensional conceptual schema that meets the end-user requirements. In order to provide a comprehensive framework for the end-user, they propose to distinguish between strategic and tactical requirements. The formers correspond to key performance indicators used for decision making, whereas the lasts represent functional objectives expressed by the end-user. In this paper, both how to collect and formalize each kind of requirements is detailed:

- Tactical and strategic requirements are suggested to be collected in a tabular representation, which in turn, must produce a *decisional dictionary* (i.e., a general view of requirements regarding the information and processes –aimed at modeling the ETL process- involved). At this level, strategic requirements differ from tactical requirements as they are expressed as measures whose only analysis perspective is the time dimension. On the contrary, tactical requirements correspond to the traditional multidimensional queries. To facilitate its collection and the creation of the decisional dictionary, the authors propose a pseudo-language, derived from natural language, to express requirements in a higher abstraction level and easily derive the tabular shape demanded a posteriori.
- Finally, requirements gathered are formalized using *decisional diagrams* by means of transformation rules. These diagrams are closer to how people involved in decision making think and reason and, simultaneously, captures all the information and processes involved to meet them.

Finally, the multidimensional schema must be derived from the decisional diagrams defined.

(Prat, Akoka & Comyn-Wattiau, 2006) present a method to derive the conceptual, logical and physical schema of the data warehouse according to the three abstraction levels recommended by ANSI / X3 / SPARC. Starting from end-user requirements, the conceptual phase leads to a *Unified Modeling Language* (UML) model. To this end, UML is enriched with concepts relevant to multidimensionality that will facilitate the generation of the logical schema. The logical phase maps the enriched UML model into a multidimensional schema and finally, the physical phase maps the multidimensional schema into a physical database schema depending on the target implementation tool (in this case Oracle MOLAP).

At each phase, they introduce a metamodel and a set of transformations to perform the mapping between metamodels. In this study, we will focus on the method to produce the conceptual and logical schemas and we will avoid to discuss the transformations to be performed to derive the physical schema.

- Conceptual phase: In this first step, the authors embrace requirements elicitation and the conceptual representation of requirements. First, requirements should be captured by means of a UML-compliant system analysis method. Requirements engineering techniques used in transactional design processes may be considered, and for example, they mention interviews, joint sessions, study of existing reports and prototyping of future reports as potential techniques to be used. Next, requirements are represented in a UML class diagram that needs to be enriched to capture multidimensional semantics. To do so, they present an extension of the UML metamodel.
  - Classes which are not association classes are denoted as ordinary classes. Similarly, associations which are not association classes are denoted as ordinary associations.
  - Each attribute of an ordinary class must be identified as an attribute or not. According to authors, it must be decided by the end-user and designers jointly.
  - Each attribute belonging to one-to-one or many-to-one relationships is transferred to the to-many side.
  - Generalizations are transformed to facilitate their mapping to the logical level. Each specialization is mapped to a new class that is related to the superclass by means of an aggregation relationship.

- Logical phase: Creating the logical schema from the enriched conceptual model produced in the first phase is immediate and a set of transformations expressed in *Object Contraint Language* (OCL) are presented. They also introduce an ad hoc multidimensional metamodel to represent the logical schema as follows:
  - Every many-to-many association of the conceptual model is identified as a fact of interest and their attributes (if any) are mapped into measures of the fact. This fact would be dimensioned by mapping the ordinary classes directly or indirectly involved in the association. Similarly, every ordinary class containing numerical values of interest is also identified as a fact. In this case, the fact is dimensioned by one dimension level defined by mapping the class (similar to the approach presented in (Phipps & Davis, 2002)).
  - Next, following many-to-one relationships between ordinary classes we give rise to aggregation hierarchies for each dimension level identified in the previous step.
  - Descriptors are defined from those non-identifier attributes from the classes involved in the dimension hierarchy that have not been chosen as measures of interest.
  - Finally, for each measure and for each dimension related to the fact where the measure is defined, it is compulsory to define which aggregation functions preserve a meaningful aggregation.

(Romero & Abelló, 2010a) present a method to derive conceptual multidimensional schemas from requirements expressed in SQL queries. Thus, it assumes relational data sources. This approach is fully automatic and follows a hybrid paradigm, which was firstly introduced in (Romero & Abelló, 2006). On the one hand, unlike other hybrid approaches, it does not carry out two well-differentiated phases (i.e., data-driven and requirement-driven) that need to be conciliated a posteriori, but carry out both phases simultaneously. In this way, both paradigms benefit from feedback returned by each other and eventually, it is able to derive more valuable information than carrying out both phases sequentially. On the other hand, this is the first method automating its demand-driven stage. In other words, automating the analysis of the end-user requirements. This method produces constellation schemas from the requirements (i.e., the SQL queries) and the data sources logical schema (i.e., relational schemas). Moreover, it is able to cope with denormalization in the input relational schemas and get equivalent outputs when applied over normalized (up-to third normal form) or denormalized relational sources. The multidimensional schema is derived along two different stages:

- For each input query, first stage extracts the multidimensional knowledge contained in the query (i.e., the multidimensional role played by each concept in the query as well as the conceptual relationships among concepts), that is properly stored in a graph. In this stage, the role played by the data sources logical schema will be crucial to infer the conceptual relationships among concepts.
- Second stage validates each multidimensional graph according to multidimensionality. To do so, this method defines a set of constraints that must be preserved in order to place data in a multidimensional space and produce a data cube free of summarizability problems. This step main objective is to guarantee that concepts and relationships captured in the graph give rise, as a whole, to a data cube. If the validation process fails, the method ends since data de-

manded could not be analyzed from a multidimensional point of view. Otherwise, the resulting multidimensional schema is directly derived form the multidimensional graph.

Unlike data-driven methods, this approach focuses on data of interest for the end-user (by considering the end-user requirements by means of the SQL queries). However, the user may not know all the potential analysis contained in the data sources and, unlike requirement-driven approaches, it is able to propose new interesting multidimensional knowledge related to concepts already queried by the user. To do so, it does not analyze the whole data sources but those concepts closely related to the end-user requirements. Finally, multidimensional schemas are derived from a validation process are proposed. Therefore, like in (Hüsemann, Lechtenbörger & Vossen, 2000) and (Mazón, Trujillo & Lechtenbörger, 2007), schemas proposed are sound and meaningful.

(Mazón, Trujillo & Lechtenbörger, 2007) present a semi-automatic hybrid approach that obtains the conceptual schema from user requirements and then, verifies and enforces its correctness against the data sources by means of *Query / View / Transformation* (QVT) relations. Their approach work over relational sources and requirements expressed in the *i** framework. The modus operandi of this approach shares many common points with (Bonifati et al., 2001), but in this case, they also provide mechanisms for validating the output schema.

This approach starts with a requirement analysis phase. They introduce a detailed demand-driven stage in which the user should state his / her requirements at high level by means of business goals. Then, the information requirements are derived from the information business goals. Both, goals and information requirements must be modeled by an adaptation of the *i** framework and eventually, the multidimensional conceptual schema must be derived from this formalization. Finally, the authors propose to express the resulting multidimensional schema by using an ad hoc UML extension (i.e., their own data structure) provided in the paper. Recently, the authors improved their deman-driven stage initially presented. In (Pardillo, Mazón & Trujillo, 2008) and (Carmè, Mazón & Rizzi, 2010) they propose two new approaches to detect facts and multidimensional metadata by exploiting the data source schemas. Yet, the demand-driven stage within this approach must be manually performed.

Next, they propose a final step to check the conceptual multidimensional model correctness. The objective of this step is twofold: they present a set of QVT relations based on the *multidimensional normal forms* (MNF) to align the conceptual schema derived from requirements with the relational schema of the data sources. Thus, output schemas will capture the analysis potential of the sources and at the same time, they will be validated according to the MNF. The MNF used in this paper are an evolution of those used in (Hüsemann, Lechtenbörger & Vossen, 2000), and they share the same objective. By means of five QVT relations, in a semi-automatic way, this paper describes how the conceptual multidimensional schema should be aligned to the underlying relational schema:

- 1MNF (a): A functional dependency in the conceptual schema must have a corresponding functional dependency in the relational schema.
- 1MNF (b): Functional dependencies among dimension levels contained in the source databases must be represented as aggregation relationships in the conceptual schema. Therefore, they complement the conceptual schema with additional aggregation hierarchies contained in the sources.
- 1MNF (c): Summarized measures that can be derived from regular measures must be identified in the conceptual schema. Therefore, they support derived measures.

- 1MNF (d): Measures must be assigned to facts in such a way that the atomic levels of the fact form a key. In other words, they demand to place the measure in a fact with the correct base (and thus, preserve the proper data granularity).
- 2MNF and 3MNF: These constraints demand to use specializations of concepts when structural NULLs in the data sources do not guarantee completeness.

(Song, Khare & Dai, 2007) present an automatic supply-driven method that derives logical schemas from ER models. This novel approach automatically identifies facts from ER diagrams by means of the *connection topology value* (CTV). The main idea underlying this approach is that facts and dimensions are usually related by means of many-to-one relationships. Concepts at the many-side are fact candidates and concepts in the one-side are dimension candidates. Moreover, it distinguishes between direct and transitive many-to-one relationships:

- First, the authors demand a preprocess to transform ER diagrams into binary (i.e., without ternary nor many-to-many relationships) ER diagrams.
- The CTV value of an entity is a composite function of the topology value of direct and indirect many-to-one relationships. In this formula, direct relationships have a higher weighting factor with regard to transitive ones. Thus, all those entities with a CTV value higher than a threshold are proposed as facts. Note that facts are identified by their CTV and therefore, it would be possible to consider factless facts.
- For each fact entity, its analysis dimensions are identified by means of many-to-one relationships. Moreover, the authors propose to use *Wordnet* and annotated dimensions (which represent commonly used dimensions in business processes) to enrich aggregation hierarchies depicted.

This approach does not introduce any clue to identify measures, levels and descriptors. However, working over ER diagrams, it would be rather easy to assume that measures are identified by means of numerical attributes once a concept has been identified as a fact, whereas descriptors can be identified from those entities identified as dimensions. Furthermore, no clue about how to identify levels is given and indeed, in the examplification provided in the paper, every dimension identified contains just one level (i.e., they do not identify aggregation hierarchies).

(Romero & Abelló, 2010b) present a semi-automated supply-driven approach. This approach derives conceptual schemas from OWL ontologies that may represent different and potentially heterogeneous data sources. Thus, this method will derive multidimensional schemas from data sources of our domain that do not have anything in common but that they are all described by the same domain ontology. This approach consists of three well-differentiated tasks. In each step it automatically looks for a given multidimensional concept (facts, bases and aggregation hierarchies) by means of a fully supply-driven stage. A formal pattern expressed in Description Logics (DL) is presented at each step. Finally, at the end of each step the user selects results of his / her interest and this will trigger next steps:

- The first task looks for potential facts. Those concepts related to most potential dimensional concepts and measures are good candidates. At the end of this task, the user chooses his / her subjects of interest among those concepts proposed by the method. The rest of the tasks will be carried out once for each fact identified in this step (i.e., each fact will give rise to a multidimensional schema).

- The second task points out sets of concepts likely to be used as bases for each fact identified. Candidate bases giving rise to denser data cubes will be presented first to the user. Finally, it would be up to the user to select those bases making more sense to him / her. Although this step is not discussed in (Romero & Abelló, 2010b), it was introduced in the original paper (Romero & Abelló, 2007). Later, it has been studied in depth in (Abelló & Romero, 2010).
- The third task gives rise to dimension hierarchies. For every concept identified as a dimension its hierarchy of levels is conformed from those concepts related to it by typical part-whole relationships. In this step, this approach builds up graphs giving shape to each dimension hierarchy and again, it will be up to the user to modify them to fit his / her needs.

Finally, this approach uses the same criteria as (Romero & Abelló, 2010a) to validate the multidimensional schema.

Nebot et al. (2009) present an innovative approach based on the semantic web technologies. This work has its origin in the biomedicine field, where data warehouses play a relevant role as analytical tools. Nevertheless, as the authors shown in some of their publications, this approach can be generalized and used in alternative scenarios.

First of all, it is important to remark that the nature of this approach differs from the other works introduced in this section in the sense that it provides a solution for a wider scenario: the data warehouse modeling task (discussed in here) and the ETL process design.

In this work they distinguish between domain ontologies (containing the agreed terminology about the domain) and application ontologies (containing the detailed knowledge needed for a specific application). This method aims at developing *Multidimensional Integrated Ontologies* (MIOs), which gather only the relevant knowledge from the application ontologies aligned together with the domain ontologies. Next, they aim at extracting the ontological instances from the applications and populate the data warehouse, which is based on the MIOs defined. According to their definitions, we can see the MIOs as the data warehouse schema from where to extract multidimensional cubes (i.e., perform OLAP analysis) or any other kind of analysis tasks. Mappings between the domain ontologies and the application ontologies, as well as between the overlapped part of the application ontologies are needed in advance. From here on, we will focus on the modeling task proposed within this approach and thus, in defining and validating the MIOs and the eventual multidimensional cubes of interest. Relevantly, the authors argue that a MIO is a filtered, multidimensional compliant (i.e., with orthogonal dimensions and free of summarizability problems) ontology, derived from the available domain and application ontologies, from where to extract the multidimensional cubes. As this framework was thought for biomedicine scenarios (i.e., very large, distributed ontologies), the authors propose to carry out the multidimensional design task (i.e., what previous approaches surveyed do) from the MIOs. According to this, this work can be thought as a complementary approach rather than an alternative to previous works introduced.

The definition and generation of the MIOs is done as follows:

- By analyzing the available ontologies we must first select those concepts being the focus of analysis (i.e., the facts). Then, dimensions and measures of interest must be specified, as well as roll-up relationships. All this process, however, must be done manually.
- Concepts demanded in the previous step are expressed as Description Logics axioms and hence, the MIO generation is largely automatic.

- Finally, the authors introduce a step to validate the MIO generated. The aim is to guarantee that the eventual multidimensional cubes produced from it are sound. To do so, they check desirable properties such as if the MIO is free of summarizability problems or the orthogonality of dimensions. In this step, they distinguish between properties that can already be checked at MIO level (e.g., concept satisfiability) and those that can only be considered once the multidimensional cube is demanded (e.g., compatibility of the dimension, measure and aggregation function).

COMPARISON CRITERIA

In order to provide a comprehensive framework of the multidimensional design methods, we aim to provide a detailed comparison of the methods discussed in the previous section. Setting a basis for discussion will facilitate the mapping of the surveyed methods to a common framework from which compare each approach, detect trends such as features in common or analyze the evolution of assumptions made by the modeling methods.

These criteria were defined in an incremental analysis of the methods surveyed. For each method we captured its main features that were mapped onto different criteria. If a method introduced a new criterion, the rest of works were analyzed to know their assumptions with regard to this criterion. Therefore, criteria presented below were defined in an iterative process during the analysis of the multidimensional design methods.

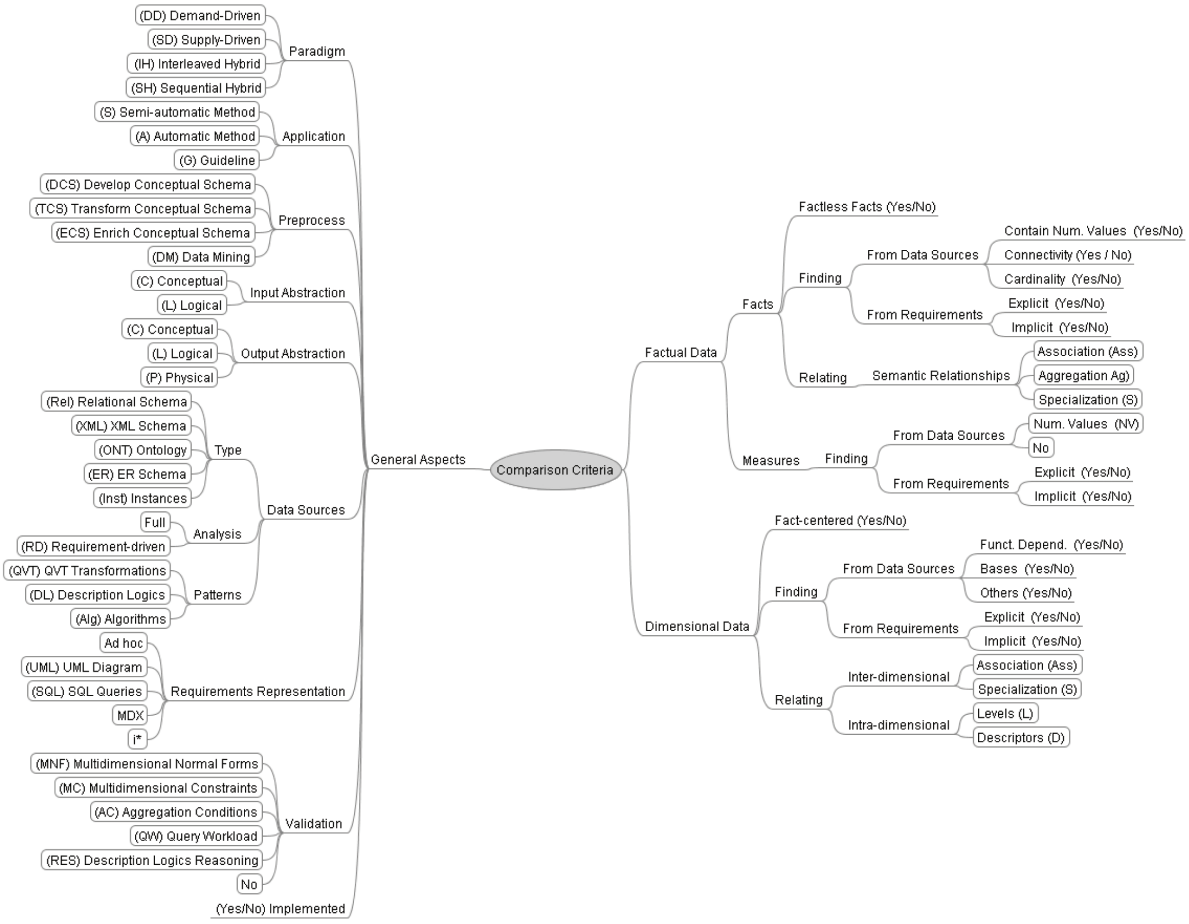
We have summarized these criteria in three main categories: general aspects, dimensional data and factual data. A graphical representation of these features is found in Figure 2. Next to each criterion, the values it may take are provided (in brackets, the acronyms). For example, the values that we may assign for the *paradigm* criterion are *demand-driven* (DD), *supply-driven* (SD), *interleaved hybrid* (IH) or *sequential hybrid* (SH). General aspects refer to those criteria regarding general assumptions made in the method, whereas dimensional and factual data criteria refer to how dimensional data and factual data are identified and mapped onto multidimensional concepts.

General Aspects

The general criteria are summarized into nine different items:

- Paradigm: According to (Winter & Strauch, 2003), multidimensional modeling methods may be classified as *supply-driven*, *demand-driven* or *hybrid* approaches. The reader may found a slightly different classification in (List et al., 2002). Furthermore, we distinguish between sequential and interleaved hybrid approaches (depending if their supply-driven and demand-driven approaches are performed either sequentially or simultaneously or sequentially).
- Application: Most methods are semi-automatic. Thus, some stages of these methods must be performed manually by an expert (normally those stages aimed to identify factual data) and some others may be performed automatically (normally those aimed to identify dimensional data). In general, only a few methods fully automate the whole process. On the contrary, some others present a detailed step-by-step guide that is assumed to be manually carried out by an expert.
- Pre-process: Some methods demand to adapt the input data into a specific format that facilitates their work. For example, these processes may ask to enrich a conceptual model with additional semantics or perform data mining over data instances to discover hidden relationships.
- Input abstraction level: Most methods (mainly those automatable) work with in-

Figure 2. Graphical view of the criteria used in the survey



puts expressed at the logical level (e.g., relational schemas), whereas some others work with inputs at the conceptual level (e.g., from conceptual formalizations such as ER diagrams or from requirements in natural language).

- Output abstraction level: Several methods choose to directly generate a star or snowflake schema, whereas some others produce multidimensional conceptual schemas. Although many approaches argue that the data warehouse method should span the three abstraction levels, only a few of them produce the conceptual, logical and physical schema of the data warehouse.
- Data sources: There are three items summarizing the main features about how data sources are considered in the method.
  - Type of data sources: The input abstraction item informs about the abstraction level of the input, whereas this item specifies the kind of technology of the data sources supported by the method. For example, if the method works at the conceptual level it may work from UML, ER conceptual schemas or ontologies, and if it works at the logical level it may work from relational schemas or XML schemas.

  - Data sources analysis: Most methods perform a fully supply-driven analysis of the data sources. However, some of them also perform a requirement-driven analysis of the data sources. Clearly, this item is tightly related to the paradigm item. Nevertheless, note that a method may follow a hybrid approach but do not consider at all requirements when analyzing the data sources.
  - Pattern formalization: Supply-driven stages usually define design patterns to identify the potential multidimensional role that concepts depicted in the data sources may play. Some methods present these patterns in an informal way, but most of them use some kind of structured language. For example, ad hoc algorithms are the most common representation but some other methods use description logic formulas or QVT Transformations.
- Requirements representation: If requirements are considered, this item summarizes how they are represented. For example, most methods use ad hoc representations (like forms, sheets, tables or matrixes), whereas some others use UML diagrams or the *i** framework. Finally, some of them lower the level of abstraction of requirements to a logical level by means of SQL queries or MDX queries.
- Validation: Some methods integrate a validation process to derive meaningful multidimensional schemas. For example, restricting summarization of data to those dimensions and functions that preserve data semantics or forming multidimensional spaces by means of orthogonal dimensions.
- Implementation: Some methods have been implemented in CASE tools or prototypes.

Factual Data

These criteria summarize how a given method identifies and handles factual data (i.e., facts and measures). First, criteria used to identify measures are summarized as follows:

- Data sources: Up to now, looking for numerical concepts is the only heuristic introduced to identify measures from the data sources.
- Requirements: Most approaches consider requirements to identify measures. We distinguish if the method only considers explicit measures or also implicit ones. Implicit measures are those explicitly stated in the requirements but implicit in the data sources (i.e., there is not a concept in the data sources that would correspond to it, but they can be derived from an already existing concept(s) in the data sources). For example, derived measures. Therefore, some kind of reasoning over the data sources is needed.

Next, we introduce criteria used to identify facts. These criteria refer to how facts are identified from the data sources or from requirements, and how they may be semantically related in the resulting schema:

- Factless facts: This kind of facts were introduced by Kimball in (Kimball et al., 1998). they are also known as empty facts and they are very useful to describe events and coverage, and a lot of interesting questions may be asked from them.
- Data sources: Most of the methods demand to explicitly identify facts by means of the requirements, but some others use heuristics to identify them from the data sources. For example, in case of relational sources, most use heuristics such as table cardinalities and the number of numerical attributes

that a table contains. Furthermore, some works also look for concepts with high to-one connectivity (i.e., with many potential dimensional concepts).

- Requirements: Similar to measures, if requirements are considered, we distinguish among explicit and implicit facts. We denote by implicit facts those that have not been explicitly stated in the requirements but can be identified from a requirement-driven analysis of the sources.
- Semantic relationships: In case of producing a conceptual schema, some methods are able to identify semantic relationships between facts. We distinguish among associations, aggregations (also called roll-up / drill-down relationships) and generalizations. In the multidimensional model, it means that we may perform multidimensional operators such as drill-across or drill-down over them.

Dimensional Data

These criteria analyze how the method identifies and handles dimensional data (i.e., dimensions, levels and descriptors). We have two main groups of items. Those referring to how dimensional data is identified (either from the data sources or from requirements), and how they are semantically related in the resulting schema. The process to identify dimensions, levels and descriptors must be understood as a whole and, unlike criteria used to identify factual data, we do not distinguish among criteria to look for different dimensional concepts. Roughly speaking, most approaches start looking for concepts representing interesting perspectives of analysis and from these concepts they look for aggregation hierarchies (i.e., levels). The whole hierarchy is then identified as a dimension and level attributes are considered to play a descriptor role:

- Fact-centered: Most methods look for dimensional data once they have identified facts. From each fact, dimensional concepts are identified using a wide variety of techniques according to the method inputs, but always looking for functional dependencies starting from the fact.
- Data sources: There are several techniques to identify dimensional concepts from data sources. We classify these techniques in three main groups: discovering functional dependencies, discovering bases and others. At the conceptual level, functional dependencies are modeled as to-one relationships, and at the logical level it depends on the technology. For example, in the relational model, dimensional concepts are identified by means of foreign keys and candidate keys. Bases (see Section *Terminology and Notation* for further information) are used to identify dimensional concepts as well. In this case, the method looks for candidate multidimensional bases in order to identify interesting perspectives of analysis (i.e., levels).
- Requirements: Dimensional concepts are mostly identified from the data sources once facts and measures have been identified. However, demand-driven approaches rely on requirements to identify dimensional concepts and some hybrid approaches also enrich their supply-driven stages with requirements. Like facts, we distinguish between explicit dimensional concepts and implicit ones.
- Intra-dimensional: Most of the methods distinguish between descriptors and levels, but some others do not.
- Inter-dimensional: Some approaches are able to identify semantic relationships between dimensions. In this case, we consider associations and generalizations as potential relationships.

METHODS COMPARISON

In this section we present a detailed summarization of the main features of each method regarding the criteria introduced in previous section, which provides a common framework to compare and discuss methods surveyed. Results are shown in Figures 3 and 4. Methods surveyed are distributed in these tables according to the chronological order. There, rows correspond to criteria introduced in the previous section and columns correspond to each method. A given cell contains information for a method and a specific criterion (we address the reader to Figure 2 to remind the meaning of each acronym). Some criteria are evaluated as *yes / no*, but most of them have alternative values. Two general values can be found for any criterion: - means that this criterion does not make sense for the method (for example, if it does not consider the data sources then, any of the criteria related to them cannot be evaluated for this method), whereas *none* means that, despite this criterion could be considered for this method, none of the alternatives are considered (i.e., it is overlooked). Therefore, *none* is the equivalent to the *no* value but for criteria having several values.

Analyzing these tables we can find some interesting trends as well as assumptions that have been considered in most of the methods surveyed. First approaches tried to contextualize the multidimensional modeling task by providing tips and informal rules about how to proceed. In other words, they presented the first guidelines to support multidimensional design. Later, when main features with regard to multidimensional modeling were set up, new formal and powerful methods were developed. These new methods focused on formalizing and automating the process. Automation is an important feature along the whole data warehouse lifecycle and multidimensional design has not been an exception. Indeed, first methods were step-by-step guidelines, but in the course of time many semi-automatic and automatic approaches have been presented. This evolution also conditioned the type of inputs used, and logical schemas were considered instead of conceptual schemas. Nowadays, last methods introduced present a high degree of automation. Moreover, we may say that this trend also motivated a change of paradigm. At the beginning, most methods where demand-driven or, in case of being hybrid approaches, they gave much more weight to requirements than to data sources. However, eventually, data sources gained relevance. This makes sense because automation has been tightly related to focusing on data sources instead of requirements. Consequently, first methods introduced gave way to others largely automatable and mostly following a supply-driven framework.

Nevertheless, today, it is assumed that the ideal approach to design multidimensional data warehouses must be a hybrid approach. In this line, last works introduced are mainly hybrid approaches.

In these tables we can also note the evolution of how the multidimensional model has been considered. First approaches used to produce logical multidimensional schemas but later, most of them generate conceptual schemas. One reason for this situation could be that Kimball introduced multidimensional modeling at the logical level (i.e., as a specific relational implementation). With the course of time, it has been argued that it is necessary to generate schemas at a platform-independent level and in fact, the multidimensional design should span the three abstraction levels (conceptual, logical and physical) like in the relational databases field.

About the kind of data sources handled, most of the first approaches choose conceptual entity-relationships diagrams describing the data sources. ER diagrams were the most spread way to represent operational databases (the most common type of data source to populate the data warehouse) but the necessity to automate this process and the need to provide up-to-date conceptual schemas to the data warehouse designer motivated that many methods worked over relational schemas instead

Figure 3. Summary of the comparison of multidimensional design methods

	[KRTR98]	[CT98]	[GR09]	[BvE99]	[HLV00]	[MK00]	[BCC+01]	[PD02]	[WS03]
General Aspects									
Paradigm	DD	IH	SH	SH	DD	SD	SH	SH	DD
Application	G	G	S	G	G	G	S	S	G
Pre-process	-	DCS	-	ECS	-	DCS	-	-	-
Input Abstr.	C	C	C/L	C	C	C	C/L	L	C
Output Abstr.	L	L	C/L/P	L	C	L	L	C	C
Data Sources									
↪ Type	-	ER	ER/Rel	SER	-	ER	ER	Rel	-
↪ Analysis	-	RD	Full	Full	-	Full	Full	Full	-
↪ Patterns F.	-	None	Alg	None	-	None	Alg	Alg	-
Req. Expr.	ad hoc	ad hoc	ad hoc	ad hoc	ad hoc	-	ad hoc	MDX	ad hoc
Validation	No	No	No	No	MNF	No	No	No	No
Tool	No	No	Yes	Yes	No	No	No	No	No
Factual Data									
Facts									
Factless Facts	Yes	No	No	No	No	No	No	No	No
Requirements	Expl	Expl	Expl	Expl	Expl	-	Expl	No	Expl
Data Sources									
↪ C.Num.Val.	-	No	No	No	-	Yes	Yes	Yes	-
↪ Connectivity	-	No	No	No	-	No	No	No	-
↪ Cardinality	-	No	No	No	-	No	No	Yes	-
Semantic Rels.	-	-	Ass	-	Ag	-	-	None	None
Measures									
Requirements	Impl	Expl	Expl	Impl	Expl	-	Expl	No	Expl
Data Sources	-	No	NV	No	-	NV	NV	NV	-
Dimensional Data									
Fact-centered	No	No	Yes	Yes	No	No	Yes	Yes	No
Requirements	Expl	Expl	Expl	Expl	Expl	-	Expl	No	Expl
Data Sources									
↪ FDs	-	No	Yes	Yes	-	Yes	Yes	Yes	-
↪ Bases	-	No	No	No	-	No	No	No	-
↪ Others	-	No	No	No	-	No	No	Yes	-
Related									
Interdim.	None	-	None	-	None	-	-	None	None
Intradim.	L/D	L/D	L/D	L	L/D	L/D	L/D	L	-

of conceptual schemas. Almost every method either considers ER diagrams or relational schemas to describe the data sources. Lately, with the relevance gained by the semantic web area, some other works automating the process from XML schemas or OWL ontologies have been presented. About requirements, their representation have varied considerably. At the beginning, ad hoc representations such as forms, tables, sheets or matrixes were proposed but lately, many methods propose to formalize requirements representation with frameworks such as UML diagrams or *i**. Moreover, some works have also proposed to lower the level of abstraction of requirements to the logical level by means of SQL or MDX queries, which opens new possibilities for automating the process.

Finally, we can also identify a trend to validate the resulting multidimensional schema as well as the importance to provide a tool supporting the method.

About how to identify factual data, there are some trends that most approaches follow. Looking at the data sources, numerical concepts are likely to play a measure role, whereas concepts containing numerical attributes or those with a high table cardinality are likely to play a fact role. First methods were mainly demand-driven but later, most of them used these heuristics to identify factual concepts within supply-driven stages. However, these heuristics do not identify facts or measures but concepts likely to play that role. Thus, requirements must be considered to filter the (vast) amount of results obtained, and in the last

Figure 4. Summary of the comparison of multidimensional design methods

	[VBR03]	[JHP04]	[GRG05]	[ARTZ06]	[PACW06]	[RA10a]	[MTL07]	[SKD07]	[RA10b]	[NBPM+09]
General Aspects										
Paradigm	SH	SD	SH	DD	DD	IH	SH	SD	SH	IH
Application	S	A	S	G	G	A	S	A	S	S
Pre-process	TCS	DM	-	-	ECS	-	-	TCS	-	-
Input Abstr.	L	L	C	C	C	L	C/L	C	C	C/L
Output Abstr.	L	L	C	C	C/L/P	C	C	L	C	C/L
Data Sources										
↪ Type	XML	Inst	ER/Rel	-	-	Rel	Rel	Rel	Ont	Ont
↪ Analysis	Full	Full	RD	-	-	RD	RD	Full	Full	Full
↪ Patterns F.	None	Alg	None	-	-	Alg	QVT	None	DL	DL
Req. Expr.	ad hoc	-	*i**	ad hoc	UML	SQL	*i**	-	-	ad hoc
Validation	No	AC	No	No	AC	MC	MNF	No	MC	Res
Tool	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Factual Data										
Facts										
Factless Facts	No	No	No	No	Yes	Yes	No	Yes	No	No
Requirements	Expl	-	Expl	Expl	Expl	Impl	Expl	-	-	Expl
Data Sources										
↪ C.Num.Val.	No	Yes	No	-	-	No	No	No	Yes	No
↪ Connectivity	No	No	No	-	-	No	No	Yes	Yes	No
↪ Cardinality	No	Yes	No	-	-	No	No	No	No	No
Semantic Rels.	-	-	None	None	None	Ass/S	Ass/S	None	Ass/Ag	None
Measures										
Requirements	Expl	-	Expl	Expl	Expl	Impl	Impl	-	-	Expl
Data Sources	No	NV	No	-	-	No	No	No	NV	No
Dimensional Data										
Fact-centered	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No
Requirements	No	-	Expl	Expl	Expl	Impl	Impl	-	-	Expl
Data Sources										
↪ FDs	Yes	Yes	Yes	-	-	Yes	Yes	Yes	Yes	No
↪ Bases	No	No	No	-	-	No	No	No	Yes	No
↪ Others	No	No	No	-	-	No	No	No	No	No
Related										
Interdim.	-	-	None	None	None	Ass/S	S	None	Ass	None
Intradim.	L/D	L/D	L/D	L	L/D	L/D	L/D	L	L/D	L/D

years requirements have gained relevance again. Capturing inter-relationships between schemas (i.e., facts) have also gained relevance lately, as they open new analysis perspectives when considering multidimensional algebras. Finally, the reader may note that although Kimball introduced the concept of factless facts from the very beginning, it has been traditionally overlooked. Lately, some methods considered them again. One of the reasons could be that it is difficult to automate the identification of facts that do not have measures.

According to our study, dimensional concepts have been traditionally identified by means of functional dependencies. From the very beginning, some methods proposed to automate the identification of aggregation hierarchies. In fact, many methods use requirements to identify factual data and later they analyze the data sources looking for functional dependencies to identify dimensional data. Maybe for this reason, the use of requirements to identify dimensional concepts has not been that relevant as to identify factual data. Another clear trend with regard to dimensional concepts is that, in general, the more automatable a method is, the more fact-centered it is. About relationships among dimensional concepts, inter-dimensional relationships (like relationships between facts) open new perspectives of analysis when considering multidimensional algebras. However, in this case they have been traditionally overlooked; even more than this kind of relationships between facts. On the contrary, intra-dimensional relationships gained more and more relevance from the very beginning. Most methods agree that distinguishing among dimensions, levels and descriptors is relevant for analysis purposes.

CONCLUSION

In this paper we provide an insight to the most relevant multidimensional design methods. Specifically, we have surveyed 19 works that have been selected according to three factors: reference papers with a high number of citations, papers with novelty contributions and in case of papers of the same authors we have discussed the latest version of their works.

Since we still lack a standard multidimensional terminology and terms used among methods to describe the multidimensional concepts may vary, we have introduced a common multidimensional notation to avoid misunderstandings and facilitate the mapping of the surveyed methods to a common framework where to compare each approach.

We have also introduced a set of criteria to set a basis for discussion and detect trends such as features in common or the evolution of assumptions made along the way. These criteria were defined in an incremental analysis of the methods surveyed in this paper. For each method we captured its main features that were mapped onto different criteria. If a method introduced a new criterion, the rest of works were analyzed to know their assumptions with regard to this criterion. Therefore, criteria presented were defined along an iterative process during the analysis of the multidimensional design methods. We have summarized these criteria in three main categories: general aspects, dimensional data and factual data. General aspects refer to those criteria regarding general assumptions made in the method and dimensional and factual data criteria refer to how dimensional data and factual data are identified and mapped onto multidimensional concepts.

All in all, we have provided a comprehensive framework to better understand the current state of the area as well as its evolution.

ACKNOWLEDGMENT

This work has been partly supported by the Ministerio de Ciencia e Innovación under project TIN 2008-03863.

REFERENCES

Abelló, A., & Romero, O. (2010). Using Ontologies to Discover Fact IDs. In I. Song, C. Ordoñez (Eds.), *Proceedings of ACM 13th International Workshop on Data Warehousing and OLAP;* pp 1-8, Toronto, Canada: ACM Press.

Annoni, E., Ravat, F., Teste, O., & Zurfluh, G. (2006). Towards Multidimensional Requirements Design. *Proceedings of 8th International Conference on Data Warehousing and Knowledge Discovery; Vol. 4081, Lecture Notes of Computer Science* (pp, 75-84). Krakow, Poland: Springer.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. *Scientific American*.

Böehnlein, M., & Ulbrich-vom Ende, A. (1999). Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems. In I. Song, T. J. Teorey (Eds.), *Proceedings of 2nd International Workshop on Data Warehousing and OLAP;* pp, 15-21. Kansas City, USA: ACM Press.

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing Data Marts for Data Warehouses. *ACM Transactions on Software Engineering and Methodology*, *10*(4), 452–483. doi:10.1145/384189.384190

Cabibbo, L., & Torlone, R. (1998). A Logical Approach to Multidimensional Databases. In H. Schek, F. Saltor, I. Ramos, G. Alonso (Eds.), *Proceedings of 6th International Conference on Extending Database Technology; Vol. 1377, Lecture Notes of Computer Science* (pp, 183-197). Valencia, Spain: Springer.

Carmè, A., Mazón, J. N., & Rizzi, S. (2010). A Model-Driven Heuristic Approach for Detecting Multidimensional Facts in Relational Data Sources. *Proceedings of 12th International Conference on Data Warehousing and Knowledge Discovery; Vol. 6263, Lecture Notes of Computer Science* (pp, 13-24). Bilbao, Spain: Springer.

Codd, E. F., Codd, S. B., & Salley, C. T. (1993). *Providing OLAP (On Line Analytical Processing) to Users-Analysts: an IT Mandate*. E. F. Codd and Associates.

Giorgini, P., Rizzi, S., & Garzetti, M. (2005). Goal-oriented Requirement Analysis for Data Warehouse Design. In I. Song, J. Trujillo (Eds.), *Proceedings of 8th International Workshop on Data Warehousing and OLAP*; pp, 47-56. Bremen, Germany: ACM Press.

Golfarelli, M., Maio, D., & Rizzi, S. (1998a). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. *International Journal of Cooperative Information Systems*, *7*(2-3), 215–247. doi:10.1142/S0218843098000118

Golfarelli, M., & Rizzi, S. (1998b). Methodological Framework for Data Warehouse Design. In In I. Song, T. J. Teorey (Eds.), *Proceedings of 1st ACM International Workshop on Data Warehousing and OLAP;* pp, 3-9. Bethesda, USA: ACM Press.

Golfarelli, M., & Rizzi, S. (2009). *Data Warehouse Design. Modern Principles and Methodologies*. McGraw Hill.

Google. (2010). Google Scholar. Retrieved October, 15th, 2010, from http://scholar.google.com/.

Harzing (2010). Publish or Perish. Retrieved October, 15th, 2010, from http://www.harzing.com/pop.htm

Hüsemann, B., Lechtenbörger, J., & Vossen, G. (2000). Conceptual Data Warehouse Modeling. In M. A. Jeusfeld, H. Shu, M. Staudt, G. Vossen (Eds.), *Proceedings of 2nd International Workshop on Design and Management of Data Warehouses*; pp 6. Stockholm, Sweden: CEUR-WS.org.

Inmon, W. H., Strauss, D., & Neushloss, G. (2008). *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Morgan Kauffman.

Jensen, M. R., Holmgren, T., & Pedersen, T. B. (2004). Discovering Multidimensional Structure in Relational Data. In Y. Kambayashi, M. K. Mohania, W. Wöß (Eds.), Proceedings of 6th International Conference on Data Warehousing and Knowledge Discovery; Vol. 3181, Lecture Notes of Computer Science (pp 138-148). Zaragoza, Spain: Springer.

Kimball, R. (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. John Wiley & Sons, Inc.

Kimball, R., Reeves, L., Thornthwaite, W., & Ross, M. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses*. John Wiley & Sons, Inc.

Lehner, W., Albrecht, J., & Wedekind, H. (1998). Normal Forms for Multidimensional Databases. In M. Rafanelli, M. Jarke (Eds.), *Proceedings of 10th International Conference on Statistical and Scientific Database Management*; pp 63-72, Capri, Italy: IEEE.

List, B., Bruckner, R. M., Machaczek, K., & Schiefer, J. (2002). A Comparison of Data Warehouse Development Methods Case Study of the Process Warehouse. In A. Hameurlain, R. Cicchetti, R. Traunmüller (Eds.) *Proceedings of 13th International Conference on Database and Expert Systems Applications; Vol. 2453, Lecture Notes in Computer Science (pp* 203-215). Aix-en-Provence, France: Springer.

Mazón, J. N., Trujillo, J., & Lechtenborger, J. (2007). Reconciling Requirement-Driven Data Warehouses with Data Sources Via Multidimensional Normal Forms. *Data & Knowledge Engineering*, *23*(3), 725–751. doi:10.1016/j.datak.2007.04.004

Moody, D. L., & Kortink, M. A. (2000). From Enterprise Models to Dimensional Models: A Method for Data Warehouse and Data Mart Design. In M. A. Jeusfeld, H. Shu, M. Staudt, G. Vossen (Eds.), *Proceedings of 2nd International Workshop on Design and Management of Data Warehouses*; pp 6. Stockholm, Sweden: CEUR-WS.org.

Nebot, V., Berlanga, R., Pérez-Martínez, J.M., Aramburu, M.J. & Pedersen, T.B. (2009). Multidimensional Integrated Ontologies: A Framework for Designing Semantic Data Warehouses. *Journal of Data Semantics XIII, Vol. 5530, Lecture Notes of Computer Science* (pp, 1-36). Springer.

Pardillo, J., Mazón, J. N., & Trujillo, J. (2008). Model-Driven Metadata for OLAP Cubes from the Conceptual Modelling of Data Warehouses. *Proceedings of 10th International Conference on Data Warehousing and Knowledge Discovery; Vol. 5182, Lecture Notes of Computer Science* (pp, 13-22). Turin, Italy: Springer.

Phipps, C., & Davis, K. C. (2002). Automating Data Warehouse Conceptual Schema Design and Evaluation. In L. V. S. Lakshmanan (Ed.), *Proceedings of 4th International Workshop on Design and Management of Data Warehouses*; pp 23-32, Toronto, Canada: CEUR-WS.org.

Prat, N., Akoka, J., & Comyn-Wattiau, I. (2006). A UML-based Data Warehouse Design Method. *Decision Support Systems*, *42*(3), 1449–1473. doi:10.1016/j.dss.2005.12.001

Romero, O., & Abelló, A. (2006). Multidimensional Design by Examples. In A. M. Tjoa, J. Trujillo (Eds.), *Proceedings of 8th International Conference on Data Warehousing and Knowledge Discovery; Vol. Lecture Notes of Computer Science (pp* 85-94). Krakow, Poland: Springer.

Romero, O., & Abelló, A. (2007). Automating Multidimensional Design from Ontologies. In I. Song, T. B. Pedersen (Eds.), *Proceedings of ACM 10th International Workshop on Data Warehousing and OLAP*; pp 1-8, Lisbon, Portugal: ACM Press.

Romero, O., & Abelló, A. (2010a). Automatic Validation of Requirements to Support Multidimensional Design. *Data & Knowledge Engineering*, *69*(9), 917–942. doi:10.1016/j.datak.2010.03.006

Romero, O., & Abelló, A. (2010b). A Framework for Multidimensional Design of Data Warehouses from Ontologies. *Data & Knowledge Engineering*, *69*(11), 1138–1157. doi:10.1016/j.datak.2010.07.007

Song, I., Khare, R., & Dai, B. (2007). SAMSTAR: A Semi-Automated Lexical Method for Generating STAR Schemas from an ER Diagram In I. Song, T. B. Pedersen (Eds.), *Proceedings of ACM 10th International Workshop on Data Warehousing and OLAP;* pp 9-16, Lisbon, Portugal: ACM Press.

Vrdoljak, B., Banek, M., & Rizzi, S. (2003). Designing Web Warehouses from XML Schemas. In In Y. Kambayashi, M. K. Mohania, W. Wöß (Eds.), *Proceedings of 5th International Conference on Data Warehousing and Knowledge Discovery; Vol. 2737, Lecture Notes of Computer Science* (pp 89-98). Prague, Czech Republic: Springer.

Winter, R., & Strauch, B. (2003). A Method for Demand-Driven Information Requirements Analysis in DW Projects. *In Proceedings of 36th Annual Hawaii International Conference on System Sciences*; pp 231-239. Hawaii, USA: IEEE.