IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4322

A Review of Information Modeling and its Significance for the Development of CASE Tools for Source Integration in Data Warehouse

Azene D. Zenebe
Graduate Student, University of Maryland Baltimore County
Tel: (410) 455-3550, Fax: (410) 455-1073, azenezl@gl.umbc.edu

ABSTRACT

There are different data warehouse structures or architectures for source integration and along with corresponding modeling methods for representation and inference. Broadly they are categorized as traditional or two level perspective architecture that does not consider the enterprise conceptual model, and three/four level perspective architecture that does consider and center on the enterprise conceptual model. This paper reviews the architectural structure, and the associated representation and reasoning techniques developed for quality (accurate, consistence and complete) source integration in data warehousing. Finally, tools and applications that implement this architectural structure and the associated representation and reasoning techniques are reviewed followed by the presentation of the significance and limitations of these architectures for the development of Computer Assisted Software Engineering (CASE) tools for data warehouse design and development.

INTRODUCTION

Data warehouse (DW) designing is a complex task and requires techniques different from those adopted for operational methods [1,5,7]. These authors have also indicated the lack of a complete and consistent modeling and design methodology for data warehouse. The major areas in data warehousing are source and data integration, data refreshment, multidimensional modeling, query processing and optimization, and metadata management [8].

Source integration in data warehousing refers to the process of bringing data from the different internal and external heterogeneous data sources. The ultimate goal of the process is to represent the migration of high quality data from the sources to the data warehouse for supporting the design of views that meet the dynamic user requirements. Among others, source integration is the most important aspect of a data warehouse, and its quality is highly depends on its architecture, modeling and reasoning mechanism employed [1]. Architecture is a blue print that depicts the different components of a system and their interaction. Narrowly it can be viewed as a data/information structure of the data warehouse.

Realizing the ideal architecture requires devising a warehouse specification language, rule capabilities, wrapper/monitor interfaces, and appropriate algorithms to generate the integrator and the relevant change detection mechanism automatically. It is undesirable to hard-code a wrapper/monitor for each information source participating in a warehousing systems, especially if new information sources become available frequently [10]. Architectures and sources integration in data warehouse along with methods for representing and reasoning are reviewed next.

ARCHITECTURES AND SOURCE INTEGRATION

Two broad categories of data warehouse architectures for source integration have been widely used in industries and researches. They are traditional two level perspective (figure 2) and the four level perspective architecture (figure 1). The former focuses on the integration of sources via wrappers and mediators using different logical formalisms and technical implementation techniques, without considering a conceptual domain model as a basis for integration. These architectures not only do not support a large number of quality problems but also do not cover all tasks to be accomplished in data warehousing (only two of the five steps in the four levels perspective model). Their modeling techniques are not verifiable against desired quality factors. Projects such as the Stanford-IBM Manager of Multiple Information Sources, Squirrel and Warehouse Information Prototype at Stanford are examples [1,8].

The four level perspective architecture focus on the use of a conceptual domain model as a basis for integration. It attempts to solve the limitations of the traditional architectures. The Information Mainfold (IM) project at AT&T research is the first example. In IM, a conceptual domain model (the world view) is used as a basis for integration, but the view of the different sources and the enterprise model need to be in relational schemata irrespective of the type of sources. This is a limitation, as relational schemata cannot be the perfect choice for all data sources [1,3]. An improved and robust four level perspective architecture for source integration that has been proposed and developed by [1] is reviewed in the next section.

The Data Warehouse Quality (DWQ) Architecture

The DWQ architecture supports an incremental approach for source integration. It is centered on extended conceptual modeling techniques. A formal and an integrated architecture for source integration with a logic based formalism or representation method for the models and schemas along with a reasoning technique for verifying the data warehouse quality are provided [1,2,3,8]. The four levels in the architecture are conceptual, logical, physical and meta (figure 1). The conceptual level describes the business models underling the information systems of an enterprise. It consists of the source model, client model and enterprise model. The source model is a conceptual representation of the data residing in the source. The enterprise model is a conceptual representation of the global concepts and relationships that are important to the enterprise. The client model is a conceptual representation of the users' information needs (OLAP queries) that incorporate multidimensional aggregations [8]. The logical level describes logical content (structure) of the sources, data warehouse and OLAP queries. Schema is the central point at this level. Client schema, Data warehouse schema and Source schema are the three components. The major features at these levels are described in [1,2,3,8,9].

The physical level contains a store for the materialized data, wrappers for the sources and mediators for the information needs and the materialized data store. The basic components are agents and data stores. The meta-level comprises the meta-model that represents a repository of the meta-information concerning the data warehouse system. It is a framework for representing common properties of

Figure 1: The four level perspective (DWQ) architecture (Source [1])

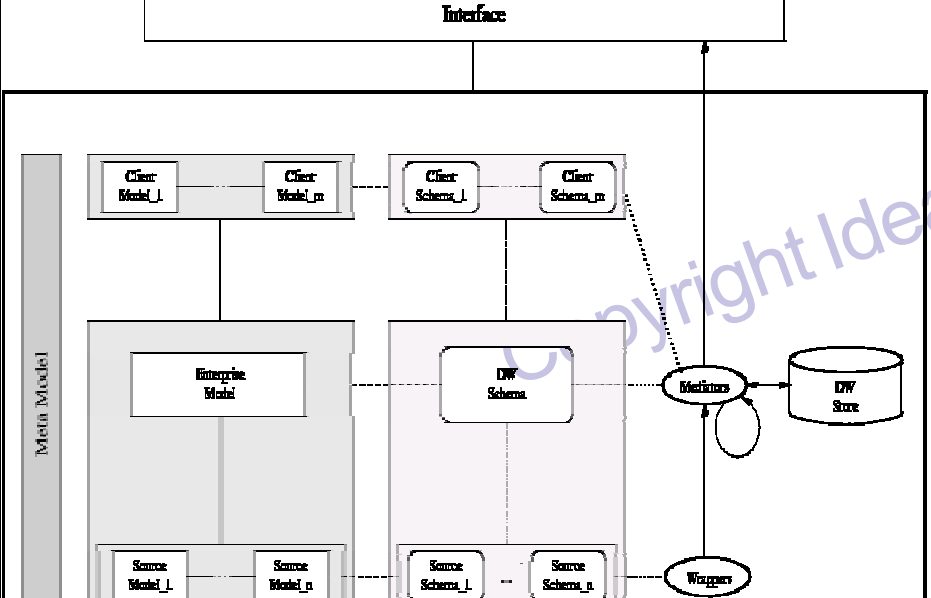
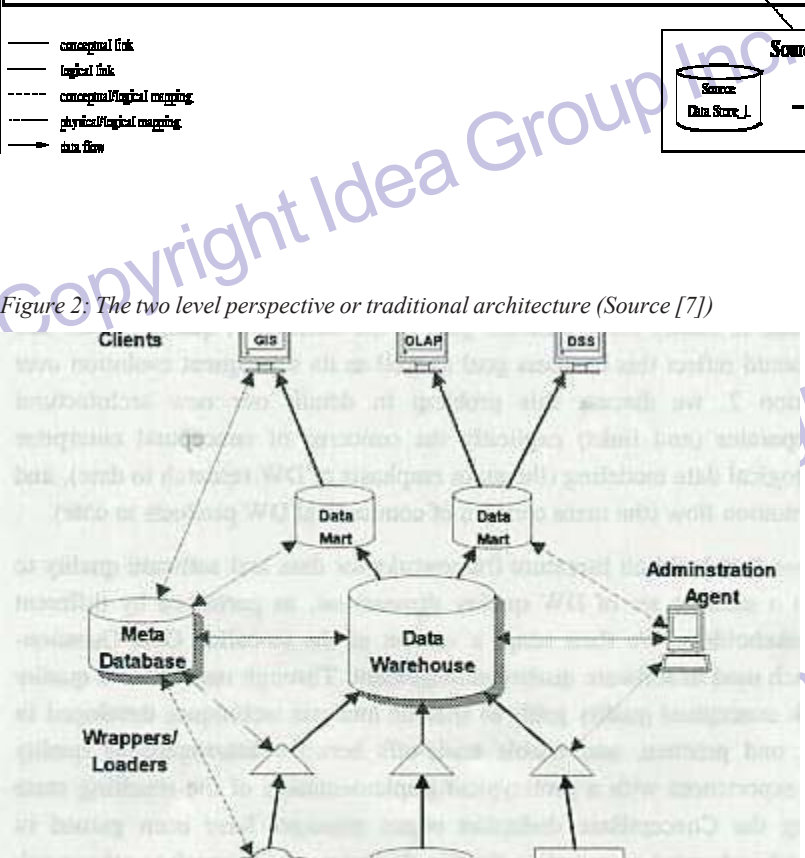


Figure 2: The two level perspective or traditional architecture (Source [7])



conceptual modeling languages for metadata- description of the components of DW and their relationships, and represents each element of the integration process [8].

In the DWQ data structure, not only an explicit model of the conceptual relationships between an enterprise model, the information sources and the OLAP clients is required, but also all other models including the source schemas have to be defined as views on this enterprise model. Hence, the wrapping and aggregation transformations performed in the logical perspective can be verified for completeness, consistency, etc., with respect to the enterprise model using the powerful representation and reasoning mechanism reviewed next.

Representation and Reasoning Techniques

Having a formalism that could represent the different components of the architecture with associated reasoning capabilities is the strength of the DWQ approach. A new Description Logic (called DLR) with a support for n-ary relations is developed for the Conceptual modeling of both the various sources and enterprise model. DLR is an extension of a special class-based logical formalism from knowledge representation field called Description Logic. It has a syntax and semantics capable of expressing the most common data models such as entity-relationship (ER) model, the relational model, and object-oriented data model. The correctness of this transformation is proved by deduction [2,3].

Since relations in the relational model can be represented in terms of DRL-relations and the constructors of DLR can be used to represent dependencies in the relational model, the transformation of relational to DRL knowledge base is plausible. Similarly mapping from object-oriented schemas into DLR knowledge bases enables the transformations. The corresponding reasoning in the two models can also be reformulated in terms of reasoning on the corresponding DLR knowledge base [1,2,3,8]. Detail examples on how to transform ER schemas into DLR knowledge base are shown in [1,6,8,9].

A logic-based formalism is used to express inter-model assertions, by specifying knowledge on the conceptual interrelationships, and the associated inference techniques provide a means to reason about the inter-dependencies among the models — between enterprise model and source models, and among the sources [1,8]. Representation at logical level is done in terms of a set of relations describing elements of the source schema or the logical definition of a materialized view in the data warehouse. A conjunctive query, for each of this relation, over the elements of the conceptual data warehouse model (represented in DLR knowledge base) establishes the connection to the enterprise model. It's predicate can denote any complex relations or concept expressible in DLR. Along with the associated reasoning techniques over conjunctive queries this formalism

is more powerful than the one used in Information Mainfold [1,8]. For the physical level, commercial solutions are well developed as in SourcePoint tool from Software AG, the data warehouse system of RedBrick and Informix's MetaCube, and hence the corresponding techniques are used [1,8]. Detailed description of these processes and examples are available in [1].

The abstract representations (knowledge-base) with associated reasoning techniques are mainly back-end feature of the data warehouse applications where users and designers cannot use them directly. A meta level is another component in the architecture that describes how to store the abstractions in a form that is accessible to designer and end users. The next section reviews the meta-model's structure and representation.

Representation of the Three Perspectives in the Meta-Database

A meta-database is a repository with all meta information about the various systems components. It is indispensable element for the end user and the designer being as front end to the data warehouse system. The DWQ approach calls for metadata structure that offer the three perspectives as a large number of quality aspects relevant for data warehousing cannot expressed with the current DW meta models such as Metadata Interchange Specification and Microsoft Repository [8].

A deductive object data model is used to capture the data structure of the different components of a data warehouse along with a quality model/measures into a meta database. The ConceptBase system, which is an implementation of Telos (an extensible metadata modeling language with a graphical support) and suitable for managing abstract representations of the data warehouse architecture and the quality, is used. The ConceptBased repository meta model of the architecture is mainly consisted of objects for conceptual models, logical schemas and data stores that are instance of the corresponding meta classes and stored in the meta database. Preloaded with these meta classes, the ConceptSystem serves as the meta database for quality oriented data warehousing. Relational schema at logical level, and extended entity-relationship and similar semantic models at conceptual level are supported. Detailed description of the meta classes, and how the representation of the three perspectives, their corresponding reasoning mechanisms and associated quality measures are captured in the meta database, is available in [8,9].

Implementations of the Architecture and the Models

Most available tools are based on the two-level data warehouse architecture. A DWQ demonstrator [9] explains how the DWQ architecture can be implemented by taking a few Telecom Italia database sources related to contracts. Enterprise and source models construction, source models integration, and sources and data warehouse schema specifications are demonstrated. An evolution result of the DWQ demo is the software tool called intelligent conceptual modeling (i•com) tool [6].

The i•com tool consists of GUI client and i•com server (background inference engine) and has a user friendly graphical interface. It adopts an extended ER conceptual data model, enriched with multidimensional aggregations and inter-schema constraints. Experiment shows that the tool is able to handle the large integrated conceptual data warehouse model of Telecom Italia [6,9].

Although formal evaluation results are not reported, the architecture is also experimented with commercial application such as the SourcePoint (by Software AG), a data mining project at Swiss Life, a DW project in Telecom Italia and administrative data warehouse of the city of Cologne, Germany. Data warehouse architectures of these applications were successfully represented in meta-database [8]. There are a few prototype tools for supporting data warehouse design. In addition to i.com, IDEA-DWCASE [11] is another recent tool that supports the data warehouse designing methodology called EINSTIEN [11]. These developments are positive steps in having a complete CASE tool for data warehouse construction.

CONCLUSION AND FURTHER RESEARCH

Among others, realizing a quality data warehouse requires a complete and sound architecture with tools, approaches and methodologies that are robust and complete. This in turn requires a complete and robust representation language, reasoning capabilities, wrapper/monitor interfaces, and appropriate algorithms for the different components of a data warehouse system. The DWQ architecture presents complete components of a data warehouse along with an automated formalism for their representation and reasoning mechanisms. Based on the architecture, a methodology for source integration - source driven and client driven integration [3] - is proposed along with tools for experimentation.

Some of areas that need further investigation are formal evaluation of the architecture along with its modeling techniques and proposed methodology, comparison of the different methodologies and development of complete CASE tools for data warehouse design and development. The use of XML as modeling languages is suggested to overcome one of the limitations of the DWQ approach - uses a very specific and complex language that is not interoperable.

REFERENCES

1. Calvanese, D., & et al. (1997) Source Integration in Data Warehousing. Project Deliverable. http://www.dis.uniroma1.it/~calvanes/publications.shtml/
2. Calvanese, D., & et al. (1998) Information integration: Conceptual modeling and reasoning support. Proceeding of the 6th International Conference on Cooperative Information Systems (CoopIS-98), pages 280-291, 1998. http://www.dis.uniroma1.it/~calvanes/publications.shtml/
3. Calvanese, D., & et al. (1999) A Principled Approach to Data Integration and Reconciliation in Data Warehousing. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99), Heidelberg, Germany, 14 - 15.6.1999 URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19/
4. Cavero, J. M., & et al.(2001). A Methodology for Datawarehouse Design: Conceptual Modeling. In Managing Information Technology in a Global Environment, Mehdi Khosrowpour (ed.), 2001 Information Resources Management Association International Conference, Toronto, Ontario Canada, May 20-23, 2001:Idea Group Publishing.
5. Franconi E. & Sattler U. (1999) A Data Warehouse Conceptual Data Model for Multidimensional Aggregation. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99), Heidelberg, Germany, 14 - 15.6.1999 URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19/
6. Franconi, E. & Ng, G. (1999) The i.com Tool for Intelligent Conceptual Modeling. Heidelberg, Germany, 14 - 15.6.1999 URL: http://www.cs.man.ac.uk/~frnconi/
7. Golfarelli, M. and Rizzi, S. (1998) A Methodological Framework for Data Warehouse Design. Proceedings ACM First International Workshop on Data Warehousing and OLAP (DOLAP 98), Nov. 7, 1998, Washington, D.C., USA.
8. Jarke M. & et al. (1998) Architecture and quality in data warehouses Proc. of the 10th Conference on Advanced Information Systems Engineering (CAiSE '98), Pisa, Italy, June, 8-12, 1998. http://www.dbnet.ece.ntua.gr/~dwq/p49.ps
9. Jarke M. & et al. (2000) Concept Based Design of Data Warehouses: The DWQ Demonstrators. http://www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/dwq/dwqdemo/index.htm
10. Widom, J. (1995) Research Problems in Data Warehousing. Proc. of 4th Int'l Conference on Information and Knowledge Management (CIKM), Nov. 1995. URL: http://www.dbnet.ece.ntua.gr/dwq/p49.ps
11. Miguel, A. D. and et al. (2000) IDEA-DWCASE: Modeling Multidimensional Databases. Conference on Extending Database Technology, March 27-31 2000, Konstanz – Germany. URL: http://www.edbt2000.uni-konstanz.de/demos/

A Fuzzy Knowledge Based Fault Tolerance Mechanism for Wireless Sensor Networks

Sasmita Acharya and C. R. Tripathy (2018). *International Journal of Rough Sets and Data Analysis (pp. 99-116).*

www.irma-international.org/article/a-fuzzy-knowledge-based-fault-tolerance-mechanism-for-wireless-sensor-networks/190893