An Approach for Developing Platform of OLAP

Zhuo CHEN

Xinyang Agriculture and Forestry University, Xinyang, China

Keywords: DSS, OLAP, DETL.

Abstract. Decision support systems (DSS) have evolved since 1970s. The functions of the system include database access, data preprocessing, data mining, and data visualization. In this paper, a developing platform model Architecture for OLAP is introduced. The system provides a complete solution for decision making. A data mining task may integrate with several mining algorithms. Users cannot easily obtain results from only interesting portion of data. This may prevent the usage of mining from online decision support for multidimensional data. A task manager helps users select data sources and mining algorithms, and build corresponding task models by providing a task wizard. The task processing engine’s function is to schedule and execute tasks.

Introduction

Data warehouses (DW) play a central role in current DSS because they provide crucial business information to improve strategic decision-making processes [1]. With imploring data warehousing and data mining technologies, the goal of the paper to implement a decision support tool. In this system, the integration of an entire decision-making process on an interactive basis is emphasized. The system is expected to accommodate the easy addition of new data mining algorithms, as well as new data preprocessing functions and new data sources. Unfortunately, none of them has been accepted as a standard for the conceptual modeling of Data Warehouses.

A large part of the literature on schema versioning in relational databases is surveyed in [2]. Data cubes return the values of attributes that affect the decision making process in the organization such as sales amount, inventory, etc. [3].

The DW design is based on the use of the star schema and its different variations by using a relational approach. Although we consider this work as a fundamental reference in the DW field, authors are only concerned about a relational representation of the DW, and no other kind of technology is considered. From our point of view, this proposal is only oriented to the conceptual and logical design of the DWs repository, because it does not consider important aspects such as the design of ETL processes. [4] Furthermore, authors assume the existence of all the schemas of the data sources, which unfortunately is not always true. Finally, we consider that the use of a particular notation makes the application of this proposal difficult.

The Platform System Architecture

The platform consists of five parts: ETL subsystem, metadata management subsystem, data warehouse management subsystem, OLAP subsystem, and data mining subsystem. Cooperating with the ETL subsystem, the data warehouse subsystem creates a data warehouse from relational data sources, which is managed and maintained by metadata management subsystem. Based on this data warehouse, it supports OLAP and data mining tasks. The data mining subsystem includes several algorithms, and provides a task manager and a task processing engine for data mining. It provides the function of data.

Access control usually requires authentication as a prerequisite. The Audit process gathers data about activities in the system and analyzes it in order to discover security violations or to diagnose their cause.

One cube must have one or more measures. The measures selected depend upon the types of information requested by the users. Some common measures are sales, cost, expenditures, and production count.
In order to solve data conflicts in information fusion we follow the approach presented in, which is based upon a conceptual representation of the data warehouse application domain. The main idea is to declaratively specify suitable matching and reconciliation operations to be used in order to solve possible conflicts among data in different sources.

A fact is composed of measures or fact attributes. By default, all measures in the fact class are considered to be additive. For non-additive measures, additive rules are defined as constraints and are included in the fact class.

**Authorization Rules**

Sensitivity Information Assignment Rules allow us to specify multilevel security policies, but unfortunately, these rules are not enough to specify other situations in which the access class of objects or subjects is not relevant. Nevertheless, access control and audit have an important component of design. In fact, access control and audit considerations should be taken into account throughout the design process, beginning at its early stages, and not just when the system is completely developed.

**Data Mining**

The data mining subsystem organizes and executes the data mining task in the object-oriented form and its data sources are obtained from the data warehouse. It includes many data mining algorithms and possesses flexible expandability. Data mining subsystems generally include an algorithm manager, task manager, and task processing engine for data mining. An algorithm manager is in charge of registering new or unregistered existing data mining algorithms, those of which can be called by data mining tasks. The Task manager helps users select data sources and mining algorithms, and build corresponding task models by providing a task wizard. The task processing engine’s function is to schedule and execute tasks. It achieves high efficiency by using multithread technology. After the results are explained and evaluated, it is stored in a data warehouse and can be visualized or exported to files.

Create and run a data mining task using a Miner Creating or running a data mining task is very convenient by using the Miner. A straightforward wizard is provided to help the user create a complicated data mining task easily. The mining task wizard is based on dialogs. Input the general information: Users can establish some general information for this task, such as the task name, author, description of the task, etc.

**Data Preprocessing**

Giving a text document, which is represented by strings or characters, we should change it into the format that is suitable for automatic text categorization learning algorithm. In order to extract content-carrying terms and category labels, raw data must be preprocessed and normalized. It includes a series of operations,

1. change upper-case letters into low-case letters.
2. remove punctuation and digits.
3. Remove stopwords. Stopword is the high frequency terms in top of term list in a document are useless to represent a document. The word “this”, “is” and so on are typical examples. So we need to remove them from the files.
4. remove ies, oes, ees, er at the end of noun.
5. remove prefix and suffix.

By performing data cleaning, a set of meaningful stemming \( ti = \{ w_1, w_2, \ldots, w_n \} \) were then achieved from original text documents. After this step, documents can be represented by this set of words and the category \( C_j \) to which they belong. Documents in the training set therefore belonging to different categories can form different transaction tables for association rules mining. Transaction can be represented as \( \{ w_1, w_2, \ldots, w_n, C \} \).
Subsystems in OLAP Platform

System Model

The main features of DS modeling considered here are the relationships many-to-many between facts and one specific dimension, degenerated dimensions, multiple classification and alternative path hierarchies, and non-strict and complete hierarchies.

We treat both measures and dimension attributes as just attributes; since the preceding class name will tell us if we refer to a measure or a dimension attribute. Moreover, treating both measures and dimension attributes in the same way will allow us to execute security measures on attributes correctly, if in a further analysis carried out by OLAP tools, users interchange measures and dimension attributes.

The OLAP modeling used to design the data warehouse is the same one that the final user will use to query the data warehouse by front-end tools such as OLAP tools. However, this does not mean that a final user may have access to all the MD elements. Therefore, we believe that the data warehouse designer, in addition to design its structure by means of a DS modeling, should also specify access control and audit rules together with the conceptual OLAP modeling at the same in order to protect the data warehouse information against further improper disclosure.

Data Warehouse Environment

The function of the data warehouse is to provide a general data warehouse environment, by which users can create and maintain their data warehouse in accordance with different needs to finish data analysis and processing, and provide preparation for data mining tasks.

In addition, these operations finally execute the read operation on the database structure that hosts the data, and therefore, from the conceptual point of view, the most relevant operation at this point is the read one. However, specific navigating and OLAP operations are beyond the scope of this work, as in this paper we present the basis for the conceptual modeling of secure business data warehouses, and then more specific security measures for final users or specific OLAP tools may be defined from our secure enterprise data warehouse.

This function is used to set the parameter of the algorithm when the user is interested in using it. Parameters include the data source, the position to save result, and other necessary parameter to run the algorithm.

Transform Layer

This standard defines a hybrid specification for transformations. On the one hand, there is a declarative part, which provides mechanisms to define transformations as a set of relations that must hold between the model elements of a set of candidate models. This declarative part can be split into two layers according to the level of abstraction: the relational layer that provides graphical and textual notation for a declarative specification of relations, and the core layer.

That provides a simpler, but verbose, way of defining transformations. On the other hand, each DW layer is designed according to the three different kinds of OLAP viewpoints. Obviously, for each layer and viewpoint, a different kind of model is required. Moreover, a set of transformations is defined in order to obtain and integrate every model. Therefore, we consider that the whole development of a DW can be structured into an integrated framework with five layers and three viewpoints for each layer.

As we have hitherto remarked, in terms of basic database operations, all OLAP operations execute the read action.

It specifies the fact, dimension or base classes that have to be involved in the query in order for this to be applicable.

A bridge dimension is used to capture and traverse the relationships between leaves (or nodes) in a hierarchical tree. The hierarchy relationship is implemented via a bridge table that defines the relationship of each node in the hierarchy to the other nodes in the hierarchy. Kimball recommends this approach to represent organizational hierarchies and manufacturing parts explosion hierarchies.
A bridge table is also used to solve the problem of multi-valued dimensions. For example, in the financial domain an account may be associated with two or more people thus requiring the linking of those people to a single account. Creating a bridge table between the account and customer dimension can mitigate the multivalued attribute.

**DETL Architecture**

To obtain the clean data for a data warehouse, the data from source databases must be cleaned and transformed before being integrated into a data warehouse. It is a key and complex step for building data warehouses. Generally speaking, the D ETL subsystem is expected to finish the following tasks: (1) Due to data redundancy and inconsistency in the source data from diverse sources, the subsystem is expected to eliminate data inconsistency. (2) To obtain required data for the data warehouse, the subsystem is expected to convert the original data structures from application-oriented to subject-oriented and perform certain computations. (3) User friendly interface: Users can request any DETL operations expediently by using the interface including designing DETL tasks, registering new DETL DLL functions, scheduling and executing D ETL tasks, and evaluating the results of DETL tasks. (4) Integrated DETL function management and DETL task management: This module helps to register new DETL DLL functions, build new DETL tasks, and schedule and process DETL tasks. (5) Consistent and integrated metadata management: The whole subsystem is developed in a metadata-oriented way. Namely, all information of this subsystem, including data sources, algorithms and results, is managed by metadata.

DETL (Decentralize Extract-Transform-Load) integrates data from different and heterogeneous sources, applies transformation, and loads data in DW database. Moreover, DETL is an active ETL; it interacts with meta-data to generate the transformations and to specify the loading with target schema. Hence, It optimize the flow of data and reduces the update of warehouse process by making automatic the creation of valid transformation.

**DETL Function Module**

It is known that the data in a data warehouse are extracted and integrated from diverse database platforms such as Oracle, SQL Server, DB2, etc., and there are many differences between the operational data in the source database and the analytical data in a data warehouse; therefore, it is not practical to load the data from various data sources into a data warehouse.

Here we define several tagged values for the model, classes, attributes, instances and constraints. The main relevant advantage of this approach is that it gives us the possibility of taking security aspects into account from the early stages of a DW project, i.e., at the conceptual level. considered in this paper, so as to permit us to specify security aspects in the crucial ETL processes for DWs, thereby considering other operations such as delete, insert and update.

Therefore, our access control and audit model allows us to specify sensitivity information assignment rules for all elements of system models, which define static and dynamic object classification. Moreover, our model allows us to define authorization rules that represent exceptions to the general multilevel rules, where the designer can specify different situations in which the multilevel rules are not sufficient. Finally, a set of audit rules, which represent the corresponding audit requirements, can be included in the model. Once the data warehouse designer has developed the conceptual model, he or she should discover all the security issues associated to this model, and specify them through our model.

They propose a set of multidimensional normal forms in order to obtain the correct conceptual model of the DW repository from the operational data sources. Unfortunately, authors informally derive a relational representation from the conceptual model, but no formal transformations are defined to obtain this logical model. Concerning the development of ETL processes, a conceptual model is proposed in [5].
Data Node Process in DETL System

The order of the process is not conserved in DETL, for that it loads data inside DBMS in a temporary database, which has the same schema as the source data. After execution of the transformations procedures and creation of DW database, the temporary database will be deleted. The different components of DETL tool are described below:

The role of this process consists in collecting needed data from different sources. These sources could be traditional database inner digital documents that are produced by applications in enterprise, legacy data or even web documents published by other partners. The meta-data repository supplies the extraction process with the needed parameters. If the data source is a traditional database, the extraction process will consist of queries stored in the meta-data repository. Other data sources need extra programs to extract data. Therefore, the main role of the extraction process consists of supplying the system of the needed data.

The second step consists of loading data to a temporary database. This database will be used later to execute transformation queries. The temporary database has the same structure of the source database. This implies the creation of a temporary database for each source. In the case of integration data flat files or XML documents, we create a temporary database having a simple structure or we use XML schema description for XML documents. The loader reads the description of the data sources from the meta-data repository and creates the temporary database. The second goal of the loader consists on filling the temporary database with the data. In order to not fill the temporary database with unused data, which could have a dramatic influence on the DBMS, the extraction process should optimize the flow of data and not supply the loader with unneeded data.

Conclusions and Future Work

The main relevant advantage of this platform is that it gives us the possibility of taking security aspects into account from the early stages of a ETL project. At the conceptual level, the extension we used for the conceptual modeling of the model in conjunction allow us to represent all main relevant DW properties at the conceptual level. Our work for the immediate future is to improve the model, extending the set of privileges considered in this paper, so as to permit us to specify security aspects in the crucial ETL processes for DWs, thereby considering other operations such as delete, insert and update.

References