Preferences on OLAP and Generation of OLAP Schemata form Conceptual Graphical Model

K.S.Charumathi[1] , I.B.Rajeswari[2]
Department of Computer Science,Pillai's Institute of Information Technology,
University of Mumbai, New Panvel-410206

ABSTRACT
This paper discuss about the OLAP preferences and generating OLAP schema automatically using conceptual graphical models. OLAP is expressed in preferences to avoid flooding of information and empty results. Main research issues are outlined when user preferences are used in OLAP multidimensional cubes. OLAP schema is generated automatically using the tool called Computer Aided data Warehouse Engineering(CAWE). BabelFish data warehouse design tool is developed for implementing such a tool and it generates the OLAP schema automatically. This paper also lists the differences between Conceptual Graphical Model notation and the data model of commercial OLAP tools. Further, it describes the graph grammars for MD schema descriptions and the generation process for OLAP schema.

Keywords----Conceptual design, Data warehouse, Graphical Modeling notation, OLAP, Schema Genaration, User preferences

1.INTRODUCTION
. This section gives the introduction on user preferences and OLAP schema generation automatically.

1.1 Preferences

Users express the preferences by using personalizing e-services is becoming more common practice. While querying using preferences is one way avoid information and null results. Many researches are going on during last few years on preferences on multidimensional OLAP cubes. In this domain, expressing preferences would be a valuable one. Some issues are discussed upon handling user preferences on OLAP cubes.

1.2. OLAP schema generation

Design and Implementation Methodologies of data warehouse is of great interest in practice. Designing and Implementation of warehouse using a specific CASE tools which specifies the set of graphical notations. Our main aim is to build a CAWE system called BabelFish tool environment. The following objectives are used in designing this tool.

- Warehouse system is specified by set of graphical notations.
- On a conceptual level, the design of the system is performed.
- All aspects of warehouse design including data model design, data transformation design, analysis and security design etc are covered by the specification.
- The whole lifecycle of a data warehouse project is supported by providing the BabelFish environment.
- This system ensures that the consistency between static and dynamic system view and between the specification and the implementation of the system design.

The generation of OLAP schema perform a mapping between the semantics of the tool specific configuration and the tool independent graphical notation. This paper outline the implementation and integration into our tool environment.

2. BASICS ON PREFERENCES

Two major classification on preferences are quatitative and qualitative preferences. Quantitative preferences expressed by means of a scoring function. This scoring function associates a numerical score to each tuple which is returned by a query. Qualitative preferences are expressed directly as binary relations on the space of tuples. In the following section we will discuss about qualitative preferences that gives higher expressiveness than quantitative preferences.

The leading approaches for dealing with qualitative preferences are due to Kiebling [1] and Chomicki [2]. According to both, preferences are defined as a strict partial order(s.p.o) over the set of possible tuples. In [5], by applying composition operators to a set of predefined base preference constructors which gives a preference algebra and complex preferences are constructed. In [3], first-order preference formulas are used to define complex preferences. Proposals has been made for extending the preference formulation to SQL like in [3].

3. A Research Agenda

An adhoc approach has devised for dealing with preferences on OLAP multidimensional cubes. A number of issues focused while implementing preference models. The following are the main issues over preferences.

- **Preference Model**. The first problem is how to take the aggregation into account. Aggregation has impact on size of the result returned to the user and it may be inappropriate by setting improper aggregation. Improper aggregation setting may results in empty data results and flooded by detailed data. For this reason, user must express their preferences on the query aggregate level, for instance, by stating proper query yearly and daily data can be retrieved from monthly data.
- **Context-awareness.** User preference has been coupled with Context-awareness in order to introduce dependency between the particular operation and user's wish. The context is determined by a set of dimensions such as the type device he is operating on, type of data he wants to analyze and the user role.
- **Query optimization.** Query optimization techniques are introduced in preferences by using new operators in aggregation level. Rewriting expressions in better way by applying set of equivalence rules on complex preferences.
- **Query processing.** It is necessary to develop original processing techniques that is capable of efficiently copying with preferences using aggregation levels on adhoc algorithms and new types of indexes.

User interface. The problem here is how to enable the user to express preferences through an OLAP front-end. The second problem is visualization of results. Preferences can also be expressed with different granularities which may be returned together as the result of query and gives the diagrammatic forms for viewing results fall short.

4. Basic concepts on BabelFish approach

BabelFish data warehouse contains all the details to specify a data warehouse. The meta model of the data warehouse is the object oriented schema of the warehouse model. Such a model is very complexto use it for graphical representations. So we describe the view based approach which has been already deployed for object oriented software engineering tools i.e., certain subsets of the warehouse model are defined as a part of the BabelFish method design. Graphical visualization of these views can be manipulated indirectly by the designer of the warehouse model. The warehouse design represents each view on certain aspects. For e.g.

1. Static data model view [5] : It describes the conceptual schema of the warehouse.
2. Dynamic view [6] : It specifies the typical analysis tasks of the end user who performs using the warehouse data.
3. Functional view : It specifies the functional interrelationship between data.
4. Data source view : It describes the static structure of the operational data and their interrelationship to the static data warehouse model.

A graphical view definition has two components: One is query which is defined to select the part of the warehouse model and second component is the specification of displaying the selected data.

For BabelFish, views are restricted to the typed graphs because the syntax of these structures can be defined through layered graph grammers. Mapping of classes, relationships, and attributes of the warehouse can be done easily. Furthermore, from the graph structure warehouse model can be easily manipulated.

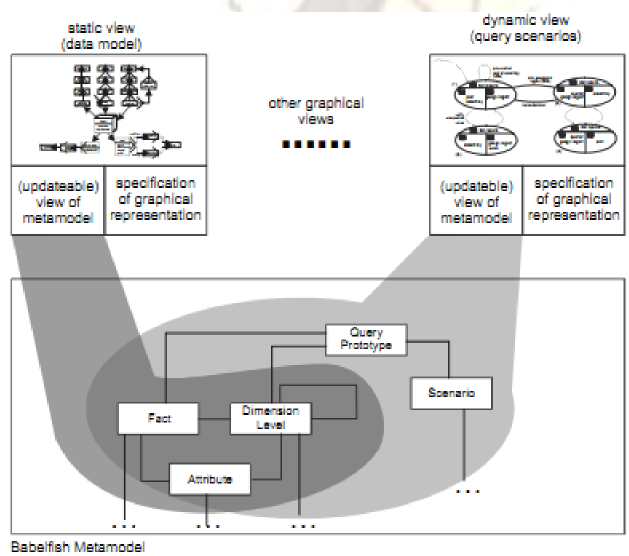


Figure 1. Graphical views of the comprehensive BabelFish metamodel are used to specify the DW

A single object of the warehouse model can be participated in different views. It is shown in fig (1) that the class dimension level are part of static and dynamic data model view. The new dimension level can also be included automatically in dynamic system view for the user query behavior. E.g. a new dimesion is added to the dynamic system view is also included in static syatem view. The BabelFish metamodel divided into three layers :

- Conceptual layer describes the system architecture and tools for implementing the objects for multidimensional schema.
- Logical layer contains the technical objects. External interfaces are exhibited by the tools which are used for implementation.
- Physical layer are tool specific issues i.e., indexing or clustering the strategies.
- Different layers are linked via interrelationship in the metamodel by tracing out the object in logical layer belongs to which object in conceptual layer. This mapping is very important while transforming schema operations from conceptual to the logical layer[7].

4.1. The Prototype Implementation

In a repository syatem the complete metamodel is stored along with view definitions and graphical representation of the elements[8]. The repository system acts as a central coordination channel between different components. It provides the updeateable services and if it is not providing this service it has to be simulated on the top of the repository system [9].

The modeler uses a generic graph editor (GraMMi) to specify the different models through the view provided onto the metamodel. The information about the graphical representation of the different objects is read by this editor and configures its interface and enforces the integrity checks. Therefore, the same editor can be used for all and different views.

The generator component called MERTGEN is described in this paper which reads all the information about the multidimensional schema and generates a tool specific executable configuration for an OLAP tool.

The evolution component called EWOK reads the evolution jobs from the repository and generates the corresponding logical evolution commands which transforms the database schema and adapt the data persistently stored in the database.Both MERTGEN and EWOK are the two specific products for our prototype implementation.

4..2 The Static Data Model View

Static data model is viewed using the ME/R notation. This static data model will be the central view of the BabelFish datamodel. ME/R(Multidimensional Entity-Relationship Model) is based on ER model and is specially designed for multidimensional modeling.

The rest of the paper refer this example schema shown in fig 2.

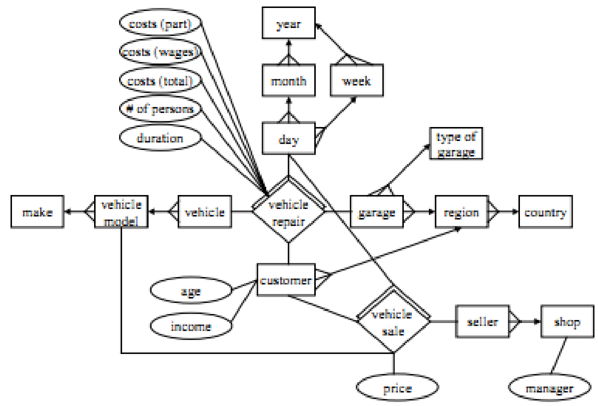


Figure 2. ME/R example schema

dia It contains two fact relationships, describing attributes and dimensions, that can be shared by the two fact relationships. The BabelFish method is represented as typed graphs and typed graph over a set of edge types $\sum_E$ and a set of node types $\sum_N$ is defined as a tuple(N, E, t_N, t_E, s, t) [6],[1]. The graphical representation of the ME/R graph elements is shown in fig 3.

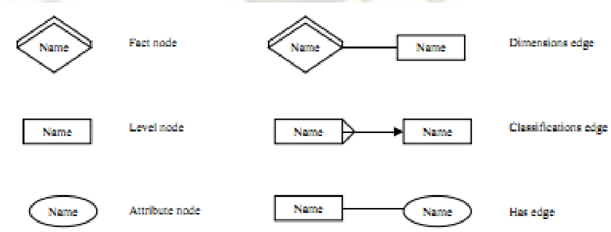
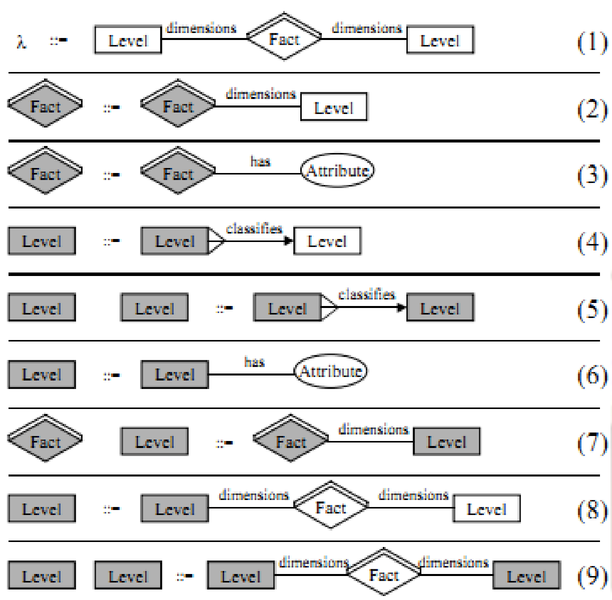


Figure 3. Representation of ME/R as a directed graph

Fig 4 gives an overview of the productions of the ME/R grammar.



The nodes, marked gray on the left hand side represent nodes that exist in the graph in order to apply the production. The nodes marked white are new nodes, created by the application of the production.

4.3 The Generation Process

The generation process is divided into four phases.

1. Conceptual schema is developed from repository system
2. The graph is parsed according to the graph grammar which is given in fig 5 to ensure syntactical correctness.
3. The syntactically correct graph is now adapted to the data model of the target system.
4. The target system is now created. Now we can construct the target OLAP system from ME/R construct. The logical schema is written back into the repository system.

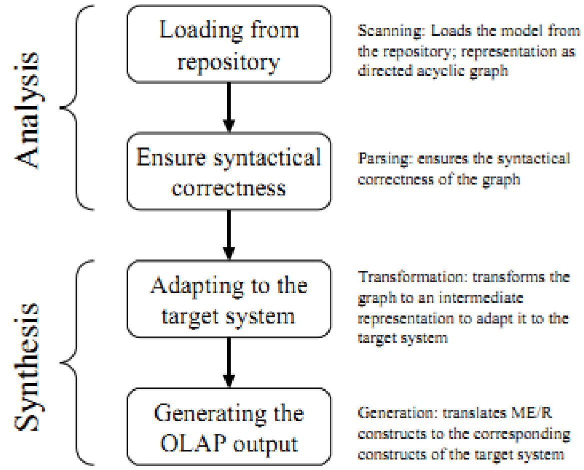


Figure 5. Four phases of the generation process

The generation process has 4 phases.

- Loading from repository (Scanning): Loads the model from the repository, representation as directed acyclic graph.
- Ensure syntactical correctness(parsing): ensures the syntactical correctness of the graph.
- Adapting to the target system(Transformation): transforms the graph to an intermediate representation to adapt it to the target system.
- Generate the OLAP output(Generation): Translate ME/R constructs to the corresponding construct of the system.

In this the first phases are called analysis phase and next two are called synthesis phase.

4.3.1. Building a parser for a graphical notation:

Syntactical correctness of a schema has to be ensured in this phase. This phase give about syntactical errors to modeler and avoid errors during generation process and the creation of faulty OLAP schemata.

4.3.2. Adapting the Schema for the Target System:

Four cases were found while adapting the schema for the target system. Dimension level will be first one which describe the attributes and it can not be modeled in some MOLAP system. If attributes are skipped during thehis phase then there may be chance of information lost. To avoid this, a new fact node is created with all the attributes and this is shown below.

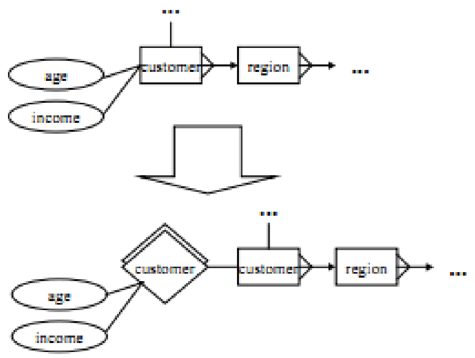
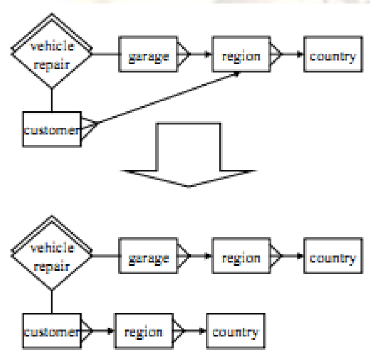


Figure 6. Attributes of a dimension level

Merging dimension is the second case that has to be solved. To obtain full hierarchy for all dimension we have to duplicate the hierarchy starting at the dimension level, where the merging occurs. The diagram is shown below.



The next problem is to resolve alternative paths inside a dimension as shown in fig below.

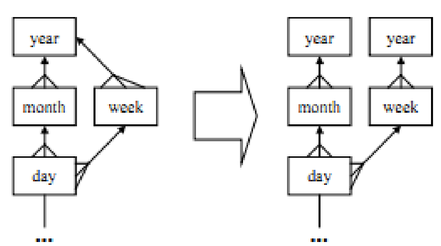


Figure 8. Alternative paths inside a dimension

It can handle dimension schemata with a tree structure. We duplicate the dimension to obtain the full hierarchy.

Finally, Powerplay schema describe multidimensional cube. We have to create a multidimensional cube to analyze different subjects in a single powerplay schema for each.

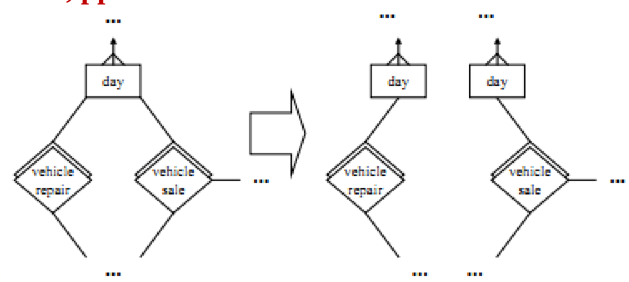


Figure 9. More than one facts (with shared dimensions)

To summarize this, the queries can be answered by both ME/R and OLAP schema after all the transformation process.

4.4. Generating the OLAP output:

The following table presents the mapping of elements of the ME/R data model to the corresponding elements of the target data models.We have to use language that can create the schemata of the OALP products to write scripts containing the full schema. In the case of powerplay the language called MDL(Model Defination Language) can be used to create Powerplay schemata. To build the multidimensional data definition statements in MDL the ME/R graph has to be traversed multiple times, because the creation of the construct has to be done in a special order in MDL.

Table 2. Mapping of ME/R elements and the elements of the target data models

ME/R data model	Powerplay data model	Metacube data model
Schema	Multiple models.	DSS System.
Fact node	Single model.	Fact table.
Dimension level node	Level.	Default attribute of a dim. element.
Attribute of a level node	Not provided[2].	Attribute of a dim. element.
Attribute of a fact node	Measure.	Measure (internal terminology: fact).
Dimensions edge	Not necessary due to the limitation of one fact per model.	Modeling element *dimension*. Relation-ship between a dim. and fact is stored in a separate table.
Classifications edge	Stored with the target node[3].	Hierarchy (internal terminology: rolls up).
Has edge	No corresponding element.	No corresponding element, stored with a measure.

5. Conclusion and Future work:

BabelFish approach is supported by CAWE environment. This environment generates the implementation of the conceptual design models, thus hiding the implementation details from the modeler.The different expressiveness of the conceptual notation compared with the data model of the commercial OLAP tool turned out to be the main challenge for the generation process. OLAP tools offer different tuning parameters for the implementation of a logical multidimensional schema on the physical layer. In order to exploit these mechanism it is necessary to have information exceeding the information contained in the static schema. Therefore, it seems promising to research the possibility of using larger parts of the warehouse metamodel as input to the generation process.

5. 6. REFERENCES

[1] W. Kiebling. Foundations of preferences in Database Systems. IN Proc. VLDB pages 311-322, Hong Kong, China, 2002.

[2] J. Chomicki. Preference formulas in relational queries. ACM Trans. On Database Systems, 28(4):427-466 2003.

[3] W. KieBling and G. Kostler. Preference SQL – design, implementation, expressions. In Proc. VLDB, pages 990-1001, Hong Kong, China, 2002.

[4] Stefano Rizzi. OLAP Preferences: A Research Agenda, Lisboa, Portugal, 2007.

[5] C.Sapia, M. Blaschka, G. Hofling, B. Dinter: Extending the E/R Model for the Multidimensional Paradigm, In Int. Workshop on Data Warehouse and Data Mining (DWDM’98), Singopore, 1998.

[6] C. Sapia: On Modeling and Predicting user Behaviour in OLAP Systems, In Proc. of the Int. Workshop on Design and Management of Data warehouse (DMDW’99), Heideberg, Germany, 1999.

[7] M. Blachka: FIESTA – A Framework for schema Evolution in Multidimensional Databases, PhD thesis, Technische Universitat Munchen, Germany, 2000.

[8] C. Sapia, M. Blaschka, G. Hofling: GraMMi – Using a Standard Repository Management System to Bluid a Generic Modeling Tool, In Proc. Hawaii Int. Conference on System Sciences(HICSS’00), Maui, Hawaii, USA, 2000.

[9] K. Hahn, C. Sapia, M. Blaschka: Automatically Generating OLAP Schemata from Conceptual Graphical Models, Technical Report, FORWISS, Munich, Germany, October 2000.