

Extension of the Goal-Decision-Information Approach to Ensure Quality in Data Warehouse Development Using Software Agent

S.Venkatesan¹, S.Mohamed Saleem²
Senior Lecturer of CSE¹, Student of CSE²
V.M.K.V. Engineering College¹, V.M.K.V. Engineering College²
Salem, India¹, Salem, India².
venkats_be@yahoo.co.in¹, saleemsweetsms@gmail.com²

Abstract - Data warehouses are complex systems that have to deliver highly-aggregated data from heterogeneous sources to decision makers. It is essential that we can assure the quality data warehouse in terms of data as well as the services provided by it. But the requirements and the environment of data warehouse systems is dynamic in nature. To handle these changes efficiently, data warehouses depend largely on the meta databases. In this paper, the proposal is to extend the Goal-Decision-Information approach to model the quality of the data warehouse. In order to fulfill the specific quality goals, dependencies among the various quality factors is exploited in this model.

Keywords: Data warehouse, quality, GDI model

I. INTRODUCTION

Nowadays organizations can store vast amounts of data obtained at a relatively low cost, however these data fail to provide information [1]. In order to solve this problem, organizations are adopting a data warehouse, which is defined as a "collection of subject-oriented, integrated, non-volatile data that supports the management decision process" [2]. Data warehouses come up as the key trend in corporate computing in the last years, since they provide managers with the most accurate and relevant that information to improve strategic decisions. Jarke et al. [3] Forecast 12 Millions American dollars for the data warehouse market. Different life cycles and techniques have been proposed for data warehouse development [4] [5] [6]. However the development of a data warehouse is a complex and very risky task.

A data warehouse architecture model has various layers of data in which data from one layer are derived from data of the lower layer. The lowest layer comprises of operational databases (data sources). They may include structured data stored in open database systems and legacy systems, or unstructured or semi-structured data stored in files. The next layer of the architecture is *primary data warehouse*, also termed as global data warehouse. The global DW keeps a historical record of data that result from the transformation, integration, and aggregation of detailed data found in the data sources.

Usually, a data store of volatile, low granularity data is used for the integration of data from the various sources, known as the *Operational Data Store (ODS)*. The data transformation and cleaning processes are also carried at the ODS so that the data populated into DW is clean and homogeneous. The top layer of views is the *local*, or *client* warehouses, which contain highly aggregated data, directly derived from the global warehouse. There are various kinds of local warehouses, such as the *data marts* or the *OLAP databases* which may use relational database systems or multidimensional data structures.

So, the data warehouse systems consists of many components, involve a large number of stakeholders with different goals, and they are constantly being monitored via administration tools. Figure 1 shows the traditional, that understanding of data warehouse. They scan by so-called wrappers huge data sets and materialize them in a central database system. Clients for data analysis and decision making access the materialized data sets to generate and the valid hypotheses about the enterprise.

All the DW components, processes and data are -or at least should be- tracked and administered from a *metadata repository*. Indeed, the DW is a very complex system; recording vast amount of data, involving a large number of processes employed for its extraction, transformation, cleansing, storage and aggregation, time-varying and change sensitive. The metadata repository acts as a path to trace all the design choices and a history of changes performed on its architecture and components.

It is essential that we can assure the quality of the data warehouse as it became the main tool for strategic decisions [7]. However, the design and analysis of the quality of a data warehouse is not well-understood and is a great problem from the perspective of the users [8]. To tackle the problem, a rich semantic data model was proposed [9] for the components of a data warehouse linked to a quality model that which been using GQM approach. The architecture model and its corresponding meta model (see Figure 2) provides for modeling objects at source, data warehouse and client level

with perspectives for the conceptual view (a variant of the ER semantic data model), the logical view (relational data model enhanced by aggregation data types), and the physical view (an extension of data flow diagrams).

II. DATA WAREHOUSE QUALITY

In [13], a definition and quantification of *quality* is given, as the fraction of Performance *over* Expectance. Taguchi defined quality as the loss imparted to society from the time a product is shipped [13]. The total loss of society can be viewed as the sum of the producer's loss and the customer's loss. It is well known that there is a tradeoff between the quality of a product or service and a production cost and that an organization must find equilibrium between these two parameters. If the equilibrium is lost, then the organization loses anyway (either by paying too much money to achieve a certain standard of quality, called "target", or by shipping low quality products of services, which result in bad reputation and loss of market share).

Quite a lot of research has been done in the field of data and software quality. Both researchers and practitioners have faced the problem of enhancing the quality of decision support systems, mainly by ameliorating the quality of their data.

The idea of GDI is that quality goals can usually not be met directly, but their realization is circumscribed by decisions that need to be taken, making use of the information available. Such decisions again can usually not be taken directly but rely on metrics applied to either the product or the process which relates to the goal in question; specific techniques and algorithms are then applied to derive the answer from the measurements. In the next subsection we provide a quick review of the GDI model.

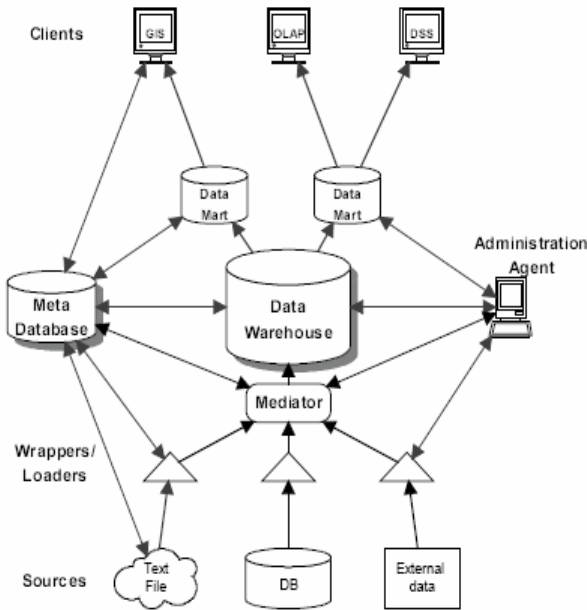


Figure 1: Traditional Data Warehouse Architecture

Therefore, the meta database of a data warehouse is the right place to explicitly represent quality goals of stakeholders and to transform them into executable queries on results of quality measurement. The results of quality measurements are also stored in the meta database. Thus, quality of a data warehouse shall be analyzed via queries to the meta database. However, it does not suggest how to use these measurements to improve upon quality of data warehouse.

A. The GDI Model

The Goal-Decision-information (GDI) model [10] is shown in Fig.3. In accordance with goal-orientation [11], [12], a goal as an aim or objective that is to be met. A *goal* is a passive concept and unlike an activity/process/event it cannot perform or cause any action to be performed. Once the goal is defined, its realization requires an active component. This active component is decision. Further the decisions need appropriate *information* for their fulfillment.

As shown in Fig.3 a goal can be either simple or complex. A simple goal cannot be decomposed into simpler ones. A complex goal is built out of other goals which may themselves be simple or complex. This makes a goal hierarchy. The component goals of a complex one may be mandatory or optional.

A decision is a specification of an active component that causes goal fulfillment. It is not the active component itself: when a decision is selected for implementation then one or more actions may be performed to give effect to it. In other words, a decision is the intention to perform the actions that cause its implementation.

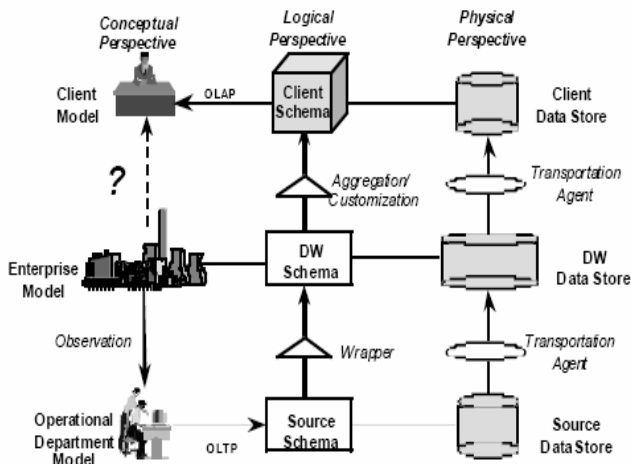


Figure 2: Levels and Perspectives for Data Warehouse models [9]

In this paper, the proposed quality framework adapts the Goal-Decision-Information (GDI) model.

Decision-making is an activity that results in the selection of the decision to be implemented. It is while performing this activity that information to select the right decision is needed. As shown in Fig. 3, a decision can be either simple or complex.

A simple decision cannot be decomposed into simpler ones whereas a complex decision is built out of other simple or complex decisions. Fig. 3 shows that there is an association 'is satisfied by' between goals and decisions. This association identifies the decisions which when taken can lead to goal satisfaction.

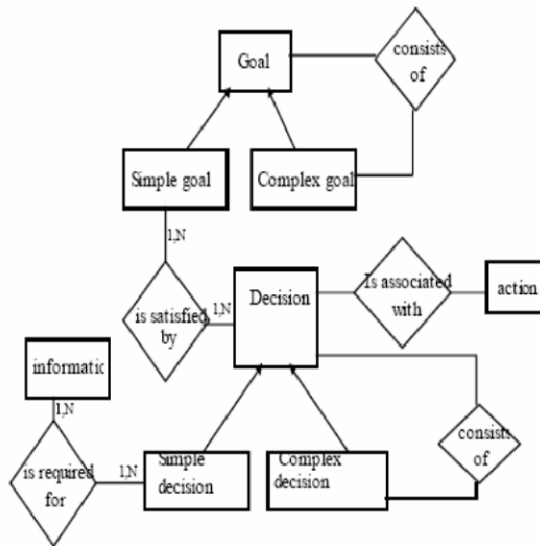


Figure 3 The Goal-Decision-InformationModel [10]

This information is a specification of the data that will eventually be stored in the Data Warehouse. Fig.3 shows that there is an association 'is required for' between decisions and information. This association identifies information required to take a decision.

B. The Proposed Quality Model

The proposed quality that model (see Fig.4) coherently supplements the architecture model consisting of three levels and perspectives mentioned in the previous section. We adopt again three layers of instantiation. At the top layer, a generic framework that follows GDI is given, extended with associations applicable to any data warehouse environment. The next lower layer specifies quality goals concerning each particular data warehouse. Thereafter, concrete values are the traces of measurement in the real world.

A *quality goal* is a project where a stakeholder has to manage the quality of the data warehouse, or a part of it. This roughly expresses natural language requirements like 'achieve the availability of source s1 at least once per week in the viewpoint of the DW administrator'. The *purpose* of the goal is obtained from the policy and the strategy of the organization. *Quality criteria* are used as the vocabulary to define abstractly different aspects of quality, as the stakeholder perceives it. Of course, each stakeholder might have a different vocabulary and different preferences in the

quality. The concrete *measurements* are carried for the quality questions, making use of *information* stored. The comparison of the actual measurement obtained and the acceptable measurement directs to the specific decision. This decision is the intention to perform the required action. This model assumes that the acceptable values, stored in the meta data repository, are provided by the stakeholder.

Formally,

Goal: is a project where a stakeholder has to manage (e.g., evaluate or improve) the quality of the data warehouse or a part of it. A quality goal can be decomposed into sub-goals, which are recursively described as quality goals. A GOAL is defined by the following assertions:

- it refers to an DW_OBJECT,
- it has a direction taken from the PURPOSE TYPE possible instances,
- it has a reference among the instances of QUALITY CRITERIA entity,
- it is defined with respect to a specific viewpoint of a given STAKEHOLDER.

A goal is refined to several QUALITY QUERIES.

Purpose: is any action to take in order to reach a certain quality goal (improve, optimize, enforce, etc.).

Stakeholder: a person who is in some way related to the data warehouse project (administrator, analyst, designer, etc.).

Quality Criteria: a subjective, high-level, user-oriented characterization of a given object. Actually, the dimensions serve as the stakeholder's vocabulary for different aspects of quality

Dw-Object: Are any object of the data warehouse framework, of any abstraction perspective (conceptual, logical, and physical) and any level (client, enterprise, and source).

Quality Query: It is placed between quality goal and quality measurement. The purpose of quality query is to mediate between the quality goal (an abstract requirement that cannot be assessed directly) and a measurement (yielding a concrete quality value).

Measurement: is a datum used to evaluate the quality goal. A measurement is done for a specific DW-OBJECT, at a specific point in time (Timestamp) (since we need present and previous values).

Actual Measurement: Is a measurement representing the fact that some quantification of the quality of a Dw-Object has been performed. This is done using a certain agent (i.e., software program in the architecture model) for the computation and producing a specific value (which is the final quantification of the question made to a answer a Goal).

Expected-Measurement: is a measurement defining the interval of allowed values of the actual Measurements. The interval must have the same domain with the produced values.

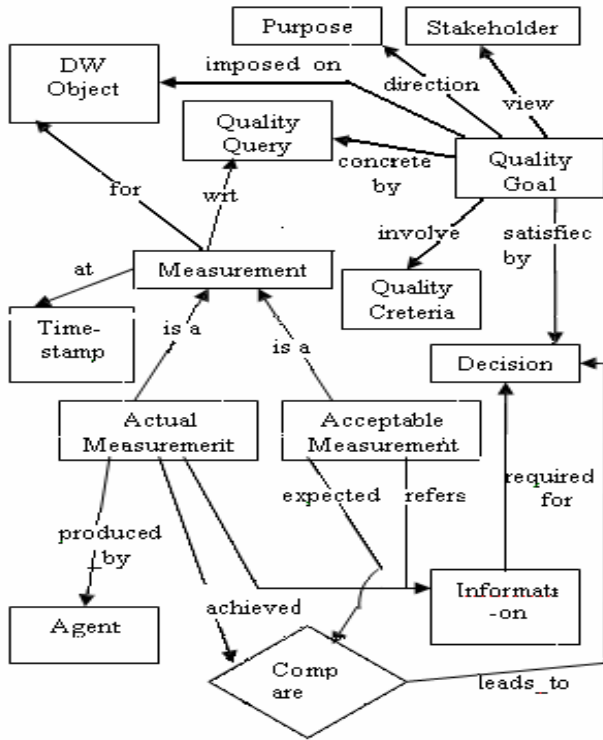


Figure 4: The proposed quality model

Agent: a software program of the architecture model. Each agent is characterized by a description for its functionality.

C. Specialization and instantiation of model

When we instantiate quality goals, we encode actual quality goals of stakeholders. Instances of quality measurements encode the plan to measure an instance of DW_Object, e.g. The relation 's1' for its quality value on Availability. The figure below defines a quality goal 'AvailGoalforRel' that states that stakeholders 'DW Administrator' may in principle be interested in achieving a certain level of availability for (source) relations. The formulation of the quality goal mentioned above is shown in Figure 5.

Figure 5 shows the instantiation and specialization of the quality goals. It expresses instance quality goals like 'Goal#10' of stakeholder 'Mr. A'. The figure is not a one-to-one translation of the above model. Instead, one part of the example goes to the simple class level, and the other to the instance level. At the simple class level, a goal 'GoalRelAvail' is formulated that refers to object type 'Relation'. The purpose is set to 'Achieve'. Moreover, it is stated that the DW Administrator is in principle interested in such a goal. At the instance level, 'Goal#10' represents the fact that 'Mr.A' (being a real DW

administrator) has instantiated this goal for the (source) relation 's1'. Note the different abstraction levels of the 'Goal#10' and 's1'! By simple instantiation, the same quality goal can be attached/detached to multiple data warehouse objects, here relations. The middle layer of Figure 5 represent patterns of quality goals rather than actual quality goals which are found in the lower layer. Thus, the middle Layer constitutes re-usable quality goals that only have to be Parameterized by the object type (here: an instance 'Relation') and the stakeholder (here an instance of 'DW Administrator').

The instantiates the quality measurement follows the same ground of instantiation and specialization. Again, it defines a pattern for measurements at the simple class level. Here, the class 'measureNullValues' defines to measure relations into percentages of null values per tuple (100% means that all values of all tuples are NULL). A range '[0; 2]' is defined as the expected interval. The agent is 'nv_counter' which possibly accesses a small portion of the relation to estimate the achieved quality value. At the instance level, the actual measurements are done. Based on the actual values.

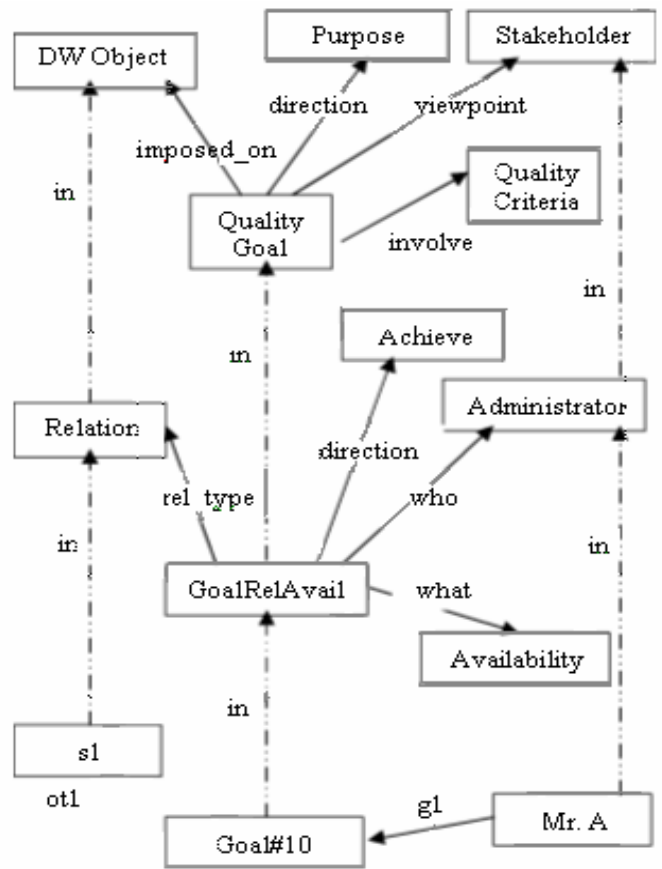


Figure 5: Instantiation and Specialization of quality goal

III.CONCLUSION AND FUTURE WORK

The presented quality model for data warehouses can be used for both design of quality and analysis of quality measurements to reach at desired decisions.

Its main features are:

Any kind of measurable object is allowed as long it is represented in the meta database of the data warehouse.

Specifically, static objects like schema concepts are supported.

x Quality goals can be formulated from the perspectives of an extensible set of stakeholders. Each stakeholder can assess the data warehouse quality from his/her perspective by evaluating the quality queries attached to his/her quality goals.

x Quality queries are executable queries on the meta database. At any time quality queries can be which have inserted , extended, modified, and removed. This is due to the fact that the meta database is not just a CASE repository but an integral part of the runtime data warehouse system.

x Quality measurements are explicitly stored in the meta database. By materializing sequences of quality measurements of the same type in the meta database, one can realize more advanced quality goals about trends by appropriate quality development area. Second, a suitable strategy for materializing the quality measurements is missing. For the moment, we assume that there are external metric agents that compute some quality value for a given measurable object. But when should the agent is activated? Supposedly, measuring the quality of a component like a source relation is computationally expensive.

queries. Their answers are the evidence for a stakeholder to One decide whether the quality is appropriate or not and to take required decisions.

There are a couple of research questions to be addressed. First, a suitable collection of quality metrics for data warehouses has to be investigated. Starting point is simply cannot afford to measure the quality for

A further research goal is to extend to method to the design of a data warehouse which includes selection of the right source databases, filters, transport agents etc. based on their quality properties.

REFERENCES

- [1] S.R. Gardner. *Building the data warehouse*, Communications of the ACM, Vol. 41, Nr.9. 52-60, September, 1998
- [2] W. H. Inmon. *Building the Data Warehouse*, second edition, John Wiley and Sons, 1997.
- [3]M.Jarke, M.Lenzerini, Y.Vassiliou and P. Vassiliadis. *Fundamentals of Data Warehouses*, Ed. Springer, 2000.
- [4] T. Hammergren. *Data Warehousing Building the Corporate Knowledge Base*. International Thomson Computer Press, Milford, 1996.
- [5] S. Kelly. *Data Warehousing in Action*. John Wiley & Sons, 1997.
- [6] L.English. *Information Quality Improvement: Principles, Methods and Management, Seminar, 5th Ed.*, Brentwood, TN: Information Impact International, Inc., 1996.
- [7] M. Janson. *Data quality: the Achilles heel of end-user computing*,Omega J. Management Science, 16, 5, 1988.
- [8] Anton, A.I.: *Goal based requirements analysis*. Proceedings of the 2nd International Conference on Requirements Engineering ICRE'96, (1996) 136-144.
- [9]D.H.Besterfield,C.Besterfield-Michna,G.Besterfield and M.Besterfield-Sacre. *Total Quality Management*. Prentice Hall, 1995.
- [10] Y.Vassiliou, T.Sellis, S.Ligoudistianos. *Data Warehouse Quality Requirements and Framework*. Technical Report D1.1, DWQ Consortium (1997).
Available at <http://www.dbnet.ece.ntua.gr/~dwq/>.