Architecture for Virtualization in Data Warehouse

J. A. Nasir *M. Khurram Shahzad*
janasir@pucit.edu.pk *mkshahzad@ciitlahore.edu.pk*

Punjab University College of Information Technology,
University of the Punjab, Lahore.

ABSTRACT

Conventional data warehouse (DW) due to structure of its schema and contents is unable to: a) support any dynamics in its source structure and contents b) unable to support hidden-subjects c) unable to provide data on-the-fly i.e. real-time data and populate hidden-subjects on their evolution. To handle these problems the concept of virtualization in DW is floated here.

In this study we have proposed architecture of virtualization approach. According to this approach, conventional DW is replaced by: i) a storage component called data-store ii) a Synthetic warehouse (SWH). Data-store is a non-subjective, content consistent, time-variant and integrated storage. On the other hand, SWH is only a structure, with no instances attached. It acts as a schema source for analytical processing and is mapped to its data-store. Subjective conversion is expected to be done on-the-fly. We are hopeful that this architecture will qualify all the evaluation parameters of: i) scalability ii) hidden-subjective support iii) source dynamics.

KEYWORDS

Data Warehouse, virtualization, on-the-fly integration, architecture, synthetic warehouse.

1. INTRODUCTION

Data warehouse (DW) integrates these various operational sources and depends upon operational sources for all changes. Changes in operational sources may result in derivation of in-consistent results [2, 3, 4]. These changes can be, schema changes or data changes. A number of efforts [3, 4, 5] have been made by the researchers of the domain to handle these changes.

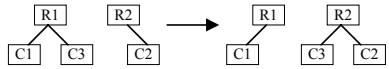
Observations-- It is observed that above approaches, due to structure of its schema and its contents are unable to: a) Support hidden-subjects b) Unable to provide real-time data availability, its importance can be found in [8] c) Cannot handle source dynamics.

Contribution-- Our approach in this paper to address the issues is based on Synthetic Warehouse (SWH). Non-subjective data from multiple sources is stored in a central repository after integration, cleaning and removing in-consistency. The repository is to be called Data Source (DS). Integration, cleaning and inconsistency removal is done by a component called Real-time ETL (R-ETL). To meet real-time what-if analysis requirements data is transformed from DS by mapping through SWH.

2. PROBLEM DEFINITION

DW describes real world, by providing integrated and subjective access to operational sources that are likely to change with time. These changes may result in production of obsolete results, so DW don't meet dynamic users requirement for decision support [3]. Changes to these sources has been categorized many times into two types [4, 7, 8] '(i) Content Changes (ii) Schema Changes. Multiple attempts have been made in [3, 5, 8] but are not successful due to one reason or the other. The problems addressed in this paper are:

1) At times, changes to DW schema are required to propagate content changes of operational sources as given in example by Eder [3]. The example is: consider a company, with sale points in multiple regions. Each region has multiple cities, if boundaries of region are changed which results in transferring the city from one region to another as shown in fig.1. For these changes, obsolete results are produced.



Rj represents Region with ID j
Ck represents City with ID k1

Fig. 1. Changing Region of City [9]

2) Conventional DW does not support subjective-scalability, i.e. hidden subjects cannot be catered by conventional subject-oriented data schema in DW. Usually, data from sources is transformed into subjective structure of DW after removing inconsistencies and

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 243–248.
© 2007 *Springer*.

anomalies; also operational data sources are either dissolved or off-lined after a specific time. In such case if subject is evolved, it is not possible to populate new subjective-schema, due to the absence of operational sources'.

3) Real-time integration is not provided in conventional data warehousing techniques. But recent study [6] has identified that real-time data is required in DW for better performance and optimization of effective decision-making in DW, which is not maintained in the conventional DW.

3. EXISTING APPROACHES

According to the best of our knowledge, first approach to handle changes to the source was, isolate changes of schema, this can be accomplished with the help of a middleware. Schema evaluation [5] was another approach. This approach supports only one version, but, regular up-gradation of DW schema, transformation package and multi-dimensional structures increases the maintenance cost. Third approach to handle the issue of source dynamics is versioning approach [10], in which changes are applied to new version and both versions are maintained.

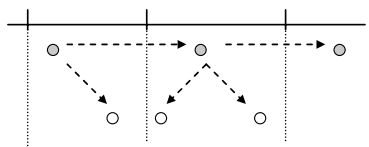


Fig. 2. Schema Versioning Graph [11]

According to recent version approach [10], changes to schema are applied to new version called 'Parent Version' which is explicitly derived from 'Child Version'. Schema versioning graph is used for demarcating parent and child relationship, as shown in fig.2. All these approaches may some-how manage source dynamics but doesn't support on-the-fly refresh facility for DW as well as subjective scalability.

4. VIRTULIZATION APPROACH

Our approach to handle above mentioned problems is based on virtualization of data warehouse. This approach avoids the evolution problem of 'hidden- subjects' by maintaining selected time-variant dataset. This data set acts as a source, for Synthetic Warehouse, for derivation of dimensions and most importantly facts.

Due to easy and costless availability of increased amount of space it is not a problem to maintain tera-bytes of data for decision-making. So for such purpose the conception of data-source (DS) has been floated which stores time-variant datasets in its non-subjective schema. For such purpose real-time transformation is done by component called R-ETL. It integrated data on the fly from operational data-sources for DS.

OLAP being the information delivery component refreshes data from DS by using Synthetic data warehouse (S-WH). S-WH is a virtual data warehouse with subject-oriented, time-variant logical structure. The source to S-WH is the data-store. Data is propagated to OLAP by mapping S-DW to DS.

4.1. ARCHITURAL COMPONENTS

Typical architecture of virtualization in DW is shown in figure 3. It includes tools for extracting data from multiple operational databases and external sources; for cleaning, transforming and integrating this data; for loading real-time data into the DS; and for periodically refreshing DS to reflect updates at the sources and to purge data from DS by S-DW to be used for OLAP by rule-mapping. The major components are:

4.1.1. *Data Sources*

Also called operational source systems and transaction processing systems. These systems keep data for a particular application [1] like order-processing, stock status, customer retentions etc. The left side of figure 3 shows set of heterogeneous transactional sources, which are used to meet the transactional requirements of the organization. It is to be note that these sources can be relational, object-oriented data sources, legacy systems and web sources. Heterogeneity in these sources can broadly be categorized into two types: Schema heterogeneity, content heterogeneity.

4.1.2. *Real-time Extractor, Transformer and Loader (R-ETL):*

Transactional sources act as data sources for subjective data [1]. For precise decision-making it is required to provide end users with most recently available data called on-the-fly data [6]. R-ETL performs integration tasks, real-time extraction and transformation and loading operations. Loading is done into a new database called data-store. Once data is loaded, it can further be used for various analytical purposes. Loading process must have following properties: a) Real-time loading b) Scheduled loading c) Consistent loading