Intermingling evolutionary and versioning approach for data warehouse by Versioning-Algebra

M.Khurram.Shahzad, J.A.Nasir, M.A.Pasha
{mkshahzad, janasir,mapasha}@pucit.edu.pk
Intelligent Information Systems Research Group (RG-IIS),
Punjab University College of Information Technology,
University of the Punjab, Lahore.

ABSTRACT

Traditional databases are unable to support analytical business requirements. As a result, the conception of data warehouse was floated. So, data warehouse with its subjective schemas facilitates analytical business requirements. But, Conventional data warehouse cannot handle changes to its operational sources, for such purpose two approaches had been proposed: i) evolution ii) versioning.

In this study, we have presented a blend of evolution and versioning approaches with the help of schema-versioning-functions, named: i) versioning function ii) reviving function iii) qualifying function. The paper formalizes algebra for version evolution operations (VEO) by modifying existing calculi. This algebra will provide strong foundations for data warehousing tools which evolutes and maintains multiple versions.

KEYWORDS

Data Warehouse, Information system automation, Versioning Operation, Schema Versioning, Multi-dimensional Schema, Versioning-algebra.

1. INTRODUCTION

Traditional databases have normalized schemas, but analytical requirements have not found to be met by operational databases. So, the concept of data warehouse (DW) was floated, 'a data warehouse is a subject-oriented, time-variant, non-volatile, integrated collection of data in favor of decision-making.'[1, 6]. DW has been meeting analytical requirements for years, but depends on operational system/s, to meet subjective-data requirements.

DW reflects real world data, stored in subjective-form, optimized to support decision-making. Nevertheless, there are some changes in operational sources that result in derivation of inconsistent analytical results [2, 3]. Researchers of the domain are trying to devise reliable strategies, which support dynamic user's requirements, which can further be used for analytical purposes but no complete solution is available.

DW depends upon its data source/s, changes to these sources my result in unrealistic results, thus affecting decision-making to large extent. 'These changes can generally be categorized into two types]:

1. Content Changes: insertion /updation/ deletion of records occurred as a result of DML commands on database.
2. Schema Changes: addition/modification/ dropping of attribute occurred as a result of DDL commands on database.'[4]

Several schemas have been developed based on Star-oriented approach like: Star schema [5, p-210], Star-flake schema [6, p-144], Snowflake schema [5, p-238] and Constellation schema [6, p-218] has been forestalled for data warehouse but these changes cannot be handled. Alternative approach to the problem is evolution of schema [7], and versioning of DW [8].

Contributions -- In this paper we use calculi called versioning-algebra (V-algebra) to make a blend of evolutionary and versioning approaches, for maintaining DW under changes. The framework is based on three functions: i) versioning, to decide the evolution of schema or building new version ii) reviving function, decide the relationship between versions i.e. parent-child relationship iii) decides validity of versions. Formal operations, that result in either evolution or versioning of DW has already been defined by Blaschka [7] In our work, we have modified her schema paradigms and make the algebra support evolution and versioning in DW for operations.

The rest of paper is organizes as follows: Section 2 discusses example illustrating potential problem. Section

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 295–300.

3 overviews attempts made to solve the problem. Section 4 has multidimensional formalization paradigms, Section 5 gives schema-versioning function, and Section 6 presents versioning operations, while Section 7 gives A-algebra to cater changes. Paper will conclude by concluding the guideline for a tool.

2. ILLUSTRATIVE EXAMPLE

To better understand the problem, consider Eder's [8] example of a DW for company's sales, sale points in multiple cities. Sales are inspected in various locations at certain time. Cities are grouped into administrative regions and products are grouped into categories. Multi-dimensional schema (MD schema) of company's DW, defined by Eder. It is composed of three dimensions and a fact table, two of the three dimensions has two dimension tables. Amount is the only fact available in the fact table, sales_fact. Amount is calculated as product of Rate and Qty of the product. i.e.

Amt = (Sales_Fact . Qty) * (Category . Rate)

The tables store the following data. Here are the dummy values inserted to the tables along with the queries executed to retrieve the results.

Select * from **product**;
Prod_ID (1, 2, 3)
Prod_Name (P1,P 2,P 3)
Cat_ID (Category1, Category 2, Category 3)

Select * from **category**;
Cat_ID (1, 2, 3)
Cat_Name (Category1, Category2, Category 3)
Rate (Category1, Category 2, Category 3)

Select * from **region**;
Region_ID (1, 2)
Region_Name (Region A, Regsion B)

Select * from **city**;
City_ID (1, 2, 3)
City_Name (BWP, ISL, KHI)
Region_ID (1, 2, 1)

Select * from **Sales_Fact**;

Time_ID	Prod_ID	City_ID	Qty	Amt
1	1	1	65	650
1	2	1	90	900
1	3	2	32	3200
1	1	3	20	2000

Eder illustrated problems related to the retrieval of inconsistent analytical results due to content changes i.e. changing the borders of regions may results in moving cities from one region to another. Such a change may have an impact on the analytical results received from DW. Assume that boundaries of region are changed in such a way that city "KHI" is moved from "Region A" to "Region B".

Let us assume a query-computing amount earned in every region before changing the boundaries of region is:
Region_Name (Region A, Region B)
Sum (Amount) (3550, 3200)

And the total amount earned in every region after changing the boundary of region is:
Region_Name (Region A, Region B)
Sum (Amount) (1550, 5200)

Firstly, as proved by example inconsistent results are produced due to content changes. Similarly, inconsistent results are also generated if region name is changed, city name is changed, product category is changed, product name is changed, and name of product category is changed. Also, when the schema is upgraded this may not give desired results.

Secondly, adding new attribute/s in one of the sources may require adding this attribute to the MD-schema if one would like to analyze values of this attribute in the DW. Also it is possible that this contributes to the calculation of facts, convectional MD-schema cannot handle these changes.

3. MD-SCHEMA FORMALIZATION PARADIGM

Two major attempts have been made, to handle dynamics to operational source [8]. These are i) Schema evolution ii) Schema versioning. First approach maintains only one schema; changes are applied to that schema, Blaschka [7] has presented evolution method with the help of algebra. In second approach changes are applied to new schema and both versions are maintained.

For our research work we need formalism that can serve as a basis for defining the schema versioning operations and their effects. Therefore, this section contains formal-mathematical notations for MD schema definition. Paradigm defined for MD-schema and model of Blaschka *et al.* [7] has been modified, by different set of attributes, to be used for versioning.

Here we assume a universal set U of alphabets to represent versions, its validity time, set of facts, dimension levels, attribute names with granularity level. So paradigm for it can be defined as:

Definition 1 – Versioned MD-Schema
Each version of MD-schema (ℵ) has seven attributes, defined as: