The Proposal of the Essential Strategies of DataWarehouse Testing

Pavol Tanuška, Oliver Moravčík, Pavel Važan, Fratišek Miksa
Faculty of Material Science and Technology
Slovak University of Technology
Paulinska 16, 917 24 Trnava, Slovakia
{pavol.tanuska, oliver.moravcik, pavel.vazan, frantisek.miksa}@stuba.sk

Abstract. *The analysis of relevant standards and guidelines proved the lack of information on actions and activities concerning data warehouse testing. The absence of the complex data warehouse testing methodology seems to be crucial particularly in the phase of the data warehouse implementation. The aim of this article is to suggest essential strategies of data warehouse testing as a part of the DW test methodology. Other important attributes can involve: prerequisites of testing, testing philosophy, traceability metrics, and specifications of test case and test procedure.*

Keywords. Datawarehouse, test strategies, methodology, UML

1 Introduction

In the 90's, William Inmon defined a data warehouse (DW) as a subject-oriented, integrated, time resolved and constant database supporting the decision-making process in management. [4]

A data warehouse was designed as a particular architecture component with the aim to transform and integrate a large amount of unprocessed data from original systems (DBs and other external data) into a new structure. A data warehouse comprises summarised historical outlook of data in production systems, thus providing the users (via OLAP and Data Mining) with a complex insight into corporate structures, and enabling them to process specialised reports and analyses, as well as gain knowledge. [6]
Main parts of datawarehouse are shown on Fig.1.

The basic structure for a multi-dimensional model (represented in Multidimensional database - MDDB) is the Star scheme with a central fact table and a set of dimension tables arranged around it, while primary keys are transformed from the dimension tables into the fact table as foreign keys.

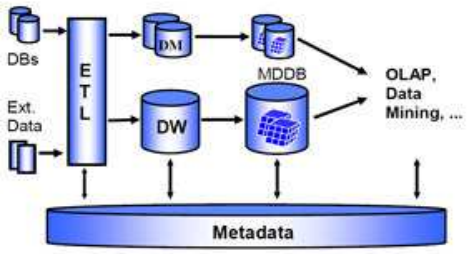

Figure 1. DataWarehouse parts

The fact table is a collection of inter-related data elements. Each fact usually represents an individual element, transaction or phenomenon subject to frequent changes and can be used in analysing any process (sales, invoices, payments, grades).

The Star scheme provides fast feedback with minimum connections and is supported by higher number of "front-end" tools.

A Snow flake scheme is an alternative to the Star scheme. The Snow flake model is a result of a decomposition of one or more dimensions with hierarchies. The Snow flake scheme comprises several dimensions consisting of several inter-related tables.

A disadvantage of the Snow flake scheme is demanding administration, limited orientation in general data structure and slower feedback.

2 DataWarehouse testing

The main difference between information systems (IS) and data warehouse (DW) testing is that the test cases revolve around queries and analytical/decision support scenarios. These cases should be written with the types of queries that representative users plan to perform, the types of scenarios that the users plan to use the DW to analyze and the various tools that make the DW "work" (ETL, reporting, querying, etc.).

There is no possible to forget to test the initial loads of the DW and the updating, since updating is a primary activity in a DW, and also it is no possible to forget to test as many different kinds of queries and scenarios that user can performing. Relations between artifacts are shown on Fig. 2.

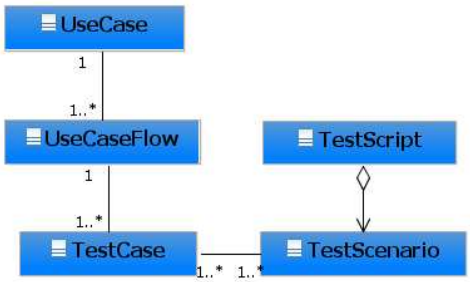

Figure 2. Relations between artifacts

There exist several levels of testing that can be performed during data warehouse testing [1] :

- Constraint testing. During constraint testing, the objective is to validate unique constraints, primary keys, foreign keys, indexes, and relationships. The test script should include these validation points.
- Source to target counts. The objective of the count test scripts is to determine if the record counts in the source match the record counts in the target. Some ETL processes are capable of capturing record count information such as records read, records written, records in error, etc.
- Source to target data validation. No ETL process is smart enough to perform source to target field-to field validation. This piece of the testing cycle is the most labor intensive and requires the most thorough analysis of the data.
- Error processing. Understanding a script might fail during data validation, may confirm the ETL process is working through process validation. During process validation the testing team will work to identify additional data cleansing needs, as well as identify consistent error patterns that could possibly be diverted by modifying the ETL code.

Absence of data warehouse global testing methodology seems to be crucial in the stage of data warehouse implementation**.** Since this topic exceeds the range of the present paper, the authors will focus on a partial objective, i.e. the proposal of essential strategies of DW testing which will be a component of the submitted DW test methodology.

3 UML and testing

An UML model represents specification documents which provide the ideal bases for deriving tests and developing testing environments. A test always requires some specification, or at least a description or documentation of what the tested entity should be, or how it should behave. Models are even more valuable if UML tools that support automatic test case generation are used. In general, we can discriminate between two ways of how to use the UML in tandem with testing activities [10], [2]:

- Model-based testing – this is concerned with deriving test information out of UML models.
- Test modeling – this concentrates on how to model testing structure and test behavior with the UML.

3.1 Model-based testing

A test case consist of one or more operations, a precondition that defines constraints on the input parameters for the test and determines the initial state of the object, and defines the final state of the object after test execution. The concepts could be mapped of a component exactly to the concepts of a test case for a component. These concepts are needed two times, first for the specification of what should happen in a test and second time for the observation of what really happens. A validation action can then be performed to determine whether the test has failed or passed, and this is called the verdict of a test. The concepts of a test case are therefore a bit more complex, but the UML Testing Profile defines them sufficiently.

3.2 Test modeling

The OMG'S Unified Modeling Language is initially concentrating only on architectural and functional aspects of software systems. This manifests itself in the following different UML diagram types:

- Use case diagrams describe the high-level user view on a system and its externally visible overall functionality
- Structural diagrams are used for describing the architectural organization of a system or its parts thereof
- Behavioral diagrams are used to model the functional properties of these parts and their interactions

- Implementation diagrams can be used to describe the organization of a system during runtime, and how the logical organization of an application is implemented physically.

The modeling and development of the testing infrastructures also involves the description and definition of testing architectures, testing behavior, and physical testing implementation including the individual test cases. So, test development essentially comprises the same fundamental concepts and procedures as any other normal software development that concentrates on function rather than on testing. Out of this motivation, the OMG has initiated the development of a UML testing profile that is specifically addressing typical testing concepts in model-based development [5]

The UML testing profile is an extension of the core UML, and it is also based on the UML meta-model. The testing profile supports particularly the specification and modeling of software testing infrastructures. It follows the same fundamental principles of the core UML in that it provides concepts for the structural aspects of testing such as the definition of test components, test contexts, and test system interfaces, and for the behavioral aspects of testing such as the definition of test procedures and test setup, execution, and evaluation.

4 The proposal of essential DW testing strategies

When proposing the essential strategies, the authors used the test procedures acquired from the results of analysis of relevant standards and guidelines. [8]

The outcome is the statement that testing is inevitable and important. There are numerous testing methods and approaches, yet individual standards do not provide detailed information, thus offering room for various disinformation. As for the design of data warehouses, it is worth to state that the current standards and regulations do not cover the activities relevant for building a data warehouse. Neither particular procedure nor activities regarding the process of a multidimensional databases proposal, ETT process, metadata or optimisation of scripts for OLAP reports are available. [3]

4.1 The ETL testing

One of the most basic tests is to verify that all expected data are loaded into the data warehouse. This includes validating that all records, all fields and the full contents of each field are loaded. [9]

Strategies can contain the following activities [3]:

- To test ETL process in dependence on architecture.
- To check data completeness with comparison of source and goal checksum. From this we can obtain information about record missing, even whole updates missing.
- To test ETL recovery after any failure by simulation of defined failure case.
- To compare unique values of key fields between source data and data loaded into datawarehouse. To validate unique constrains, primary keys, indexes and relationships.
- To test single fields with data that contains full range, to detect any problems as data reduction or data confusion.
- To create table processor with own scenario of inserted values and expected results and verify it.
- To create and use test data that cover as many as possible test cases.
- To verify of ETL process correctness is necessary use generated data for all available fields.
- To verify whether data types in datawarehouse are specified in design process and in data model.
- To verify relations between basic data and derived data.
- To set up data scenario to test relation integrity and relations between tables.
- To test data completeness – assure whether all expected data are correct transformed in compliance with rules and design specification.
- To implement attributes of quality. ETL process has to correct refuse, replace or ignore incorrect data and has to provide messages about process status.
- To test efficiency – assure whether all data are correct loaded and queries are done in expected time intervals.

4.2 The Metadata testing

The metadata are very important throughout datawarehouse life cycle. All relevant information is recorded in metadata database, it means, if there is any problem in this database, we have to expect serious problems with datawarehouse operation, include ETL problem. The metadata are used to perform ETL recovery.

The default values are defined and stored in metadata database and therefore we mustn't forget to check if these values are loaded correctly. Also is necessary to check if correct data are written into "audit_tables". These tables contain data about users, access, updates, logins and so on.

In the process of metadata testing is necessary to validate the time stamping, if every record has allocated the time identification. The next step is to check if every metadata element in metadata file utilizes specified data type and if is included in specified domain. It is very important to keep actual metadata, because any small problem with actualization can result in chain of serious troubles.

No actualized metadata in ETL process can cause incorrect data understanding. Therefore is very desirable to validate script for performing of metadata actualization. [3]

The metadata has to describe single changes in data. It means, if some changes are done, is required to record this modification. It is necessary to realize that metadata are used from design phase to operation phase. From this reason, metadata testing is second the most important activity in the datawarehouse test process.

4.3 The functionality testing

The datawarehouse functionality testing is also very important and can contain the following activities:
- To choose the single values on the ground of boundary values - as are minimum and maximum values.
- To compare results obtained from simple reports or queries that are done on source systems and thereafter on datawarehouse.
- To validate datum dimension. To insure that all datum attributes are correct – for example fiscal year, week, day indicators, indicator of last day in the month, numbers of weeks and so on.
- To verify the functionality of ETL process by loading of simulated data into datawarehouse. To check, if all data has been loaded correctly and to compare simulated data with required data. [3]
- Data testing is very important to practise together with functionality testing, because is necessary to invest a big effort to create test data and conditions.
- There has to exist a requirement to test as much as possible conditions, to ensure reliability and correct operation of datawarehouse.

4.4 The performance testing

In the phase of performance testing is necessary to do following actions:
- To realize cumulative loading of expected amount of data with aim to verify that ETL process can run in real time.
- To realize incremental loading (hourly, daily, weekly dose) of expected amount of data to verify ETL process.
- Various tools can be used for performance detection, for example, SQL Server Profiler, where is possible to set up a many timing information about execution SQL operations into logs.
- To use relevant data from ETL log, where are information containing timestamps. This can be utilized in performance tracing process of incremental data loading.
- To perform simple and complex JOIN query to make sure that it is possible to do it on huge databases, too.
- To utilize data from ETL log to find weak points that causes performance restriction. Subsequently we can start to perform datawarehouse processes tuning.
- To simulate using of application with big amount of users and to test a performance and system behavior.
- In the phase of testing use both – normal and exception scenarios. The normal scenarios contain a list of activities that user perform routinely, daily. The exception scenarios contain activities and steps that user perform in the case of error appearance.
- To trace data override process and find out if is possible to manipulate amount of overridden data.
- Together with real users to realize some queries and determine acceptance criteria for these queries. [3]

4.5 The end-to-end testing

The datawarehouse is in this phase almost prepared for operation. All datawarehouse components has been tested, functionality too. In the phase of end-to-end testing ETL process run a few days and we can simulate real conditions and cases.

ETL batching works automatically. We try to find out whether data or transactions are complete or some of them are missing.
In the same way we check data quality, fact table(s), dimension tables and loading of check tables and logs. The next step is checking of reports, data cubes and data mining models and methods. [7]

Having finished these tests is required to document all details, settings, datum of interruption, safety details and so on.
It is necessary to record everything, step by step, because according to these records, system will run in phase of operation. [3]

5 Conclusion

The testing phase as one of the stages of DW development lifecycle is very important, since the

cost depleted for the elimination of a potential error or defect in a running data warehouse is much higher. A data warehouse as well as an information system can be physically correctly tested only when the working database is loaded. A complete test should ideally approximate to real working conditions; a few data in tables is not sufficient. It is also inevitable to know multi-dimensional database structure, metadata structure, or ETT process.

The aim of the present paper is to propose the essential strategies of DW testing as a part of the database warehouse testing methodology.

6 Acknowledgments

This contribution as a part of the project No. 1/4078/07 was supported by VEGA, the Slovak Republic Ministry of Education's grant agency.

References

[1.] Cooper, R., Arbuckle, S.: How to Thoroughly Test a Data Warehouse. STAREAST, 2002.

[2.] Gross, H-G. *Component-Based Software Testing with UML.* Berlin : Springer, 2005. 316 s. ISBN 3-540-20864-X

[3.] Hrušovský Peter: NÁVRH ŠPECIFICKÝCH ČINNOSTÍ V PROCESE TESTOVANIA DÁTOVÝCH SKLADOV. Diplomová práca MTF STU Trnava, 2008

[4.] Inmon, W.H.: Building the Data Warehouse. Wiley Computer Publishing. 2002.

[5.] OBJECT MANAGEMENT GROUP. *Uml's Testing Profile.* [online]. Available at http://www.omg.org/technology/documents/formal/test_profile.htm

[6.] Ponniah, P. Data Warehouse Fundamentals – Comprehensive Guide. London: John Willey and Sons, 2001

[7.] Schreiber, P., Kebísek, M.: The possibility of neural networks utilisation at the data mining. In Proceedings of the 12th International Scientific Conference CO-MAT-TECH 2004. Bratislava, STU, 2004, s.589-595. ISBN 80-227-2117-4, Slovakia

[8.] Tanuška, P., Schreiber, P.: Validation of DATAWAREHOUSES in term of significant standards and guidelines.. In: Proceedings the 6th International Scientific – Technical Conference Process Kontrol ŘÍP 2004. Vydala Univerzita Pardubice 2004. ISBN 80-7194-662-1, Czech Republic.

[9.] Theobald J.: Strategies for Testing Data Warehouse Applications. [online]. Available at http://www.dmreview.com, Published August 28, 2007

[10.] Zeman, J., Tanuška, P., Drobná, K.: THE UTILIZATION OF UML FOR THE TEST CASE DESIGN. 8th International Scientific - Technical Conference PROCESS CONTROL 2008,June 9 – 12, 2008, Kouty nad Desnou, Czech Republic.