The Double Wave Data Warehouse Hardware Architecture for Business Intelligence Implementation

DP du Plessis, North-West University (Vaal Triangle Campus), Vanderbijlpark, South Africa, deondp@brmo.co.za
JH Kroeze, North-West University (Vaal Triangle Campus), Vanderbijlpark, South Africa, jan.kroeze@gmail.com

Abstract

South Africa is a developing country, which since 1994 has gone through a process of privatisation of some of its state departments. The Department of Posts and Telecommunications was one of those government departments. During this transition period, BI played a critical role in measuring the performance of the newly formed telecommunications company, against targets set by government. The new telecommunications company had to meet the targets set, whilst also preparing for future competition. This paper discusses the new DWDW lifecycle model that was developed to deal with the challenges faced whilst implementing the BI solution. The new DWDW lifecycle model called for a different hardware architectural design. The aim of this paper is to explain this new architectural design.

Keywords

Data Warehousing, Business Intelligence Strategy, Business Intelligence Architecture

Introduction

Burton et al. (2006:1) state that the management of business and operations in larger organisations is becoming more challenging. Managing this complexity means that Business Intelligence (BI) departments in organisations are called on to provide BI-related capabilities for understanding where and how value can be created in the business. This is done in order to respond quickly to market changes and opportunities in the contemporary business world. These macro business changes require that organisations view BI in different ways.

South Africa is a developing country, which since 1994 has gone through a process of privatisation of some of its state departments. The Department of Posts and Telecommunications was one of those government departments. During this transition period, BI played a critical role in measuring the performance of the newly formed telecommunications company, against targets set by government. The new telecommunications company had to meet the targets set, whilst also preparing for future competition. This paper discusses the new DWDW lifecycle model that was developed to deal with the challenges faced whilst implementing the BI solution. The new DWDW lifecycle model called for a different hardware architectural design. The aim of this paper is to explain this new architectural design.

The research for this paper is based on the case study mentioned above. Oates (2008:142) believes that a case study tests and investigates the real life situation. Yin (2003:1) defined a case study as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.” The BI architecture introduced in this study had to provide answers in real life business situations. Thus action research was also used and is an integral part of the case study, as the first author was involved in the planning and implementation of the BI projects undertaken.

Background to the Study

In 1994, just after the privatisation of the Department of Posts and Telecommunications, the new democratic government of South Africa was the sole shareholder of the only "fixed line" telecommunications company in the country. The government had vibrant discussions with all the relevant parties on how telecommunications might be restructured to create an even distribution of access of telecommunications services to all the people in the country. This resulted in a white paper on Telecommunications Policy, which was released in March 1996 (Anon, 1996:1).

The major proposal contained in the white paper was that the owner and operator of the fixed telephone infrastructure would be granted a limited period - the so-called "exclusivity period" - of monopoly in the provision of basic telecommunication services. This exclusivity period was to last for five years, but could be extended to six years if the telecommunications operator met network rollout and service targets. The rollout targets included doubling its number of subscriber access lines (an additional 2.7 million), installing 120,000 new public telephones, connecting 3,200 villages for the first time, and providing service to more than 20,000 priority customers such as schools and clinics (Anon, 1996:1). The exclusivity period was intended to allow the telecommunications company to expand the network as rapidly as possible in order to facilitate universal access and to move towards universal service. The agreement left the telecommunications provider with the challenge to plan and manage the implementation of targets set by government, while at the same time preparing for competition once the exclusivity period expired.

A BI solution was needed that could provide information on spare infrastructure and that could manage payment periods of debtors in order to minimise bad debts. There were several challenges whilst implementing this BI solution. The three main challenges addressed by this study were: the high BI implementation cost, the BI literacy of the BI end-users (business people do not know how to use the BI tool), and limited time to implement a BI solution (the company had to immediately start delivering on the targets of government).

The implementation of a BI solution can be very challenging at times. Sumathi and Sivanandam (2006:145) state that almost every single BI project follows a 2:2:50 pattern. This means that the project costs an average of $2 million, takes an average of 2 years to complete and has an expected 50% failure rate. The new South African telecommunications company had limited funds to develop a BI solution. The BI team therefore had to find ways to minimise the cost of implementing the BI solution. The BI solution was needed immediately and could not wait two years for completion. Failure, therefore, could not be tolerated.

The main process for developing software is called a System Development Life Cycle (SDLC) (Unhelkar, 2008:46-47). When looking at the 2:2:50 pattern that Sumathi and Sivanandam (2006:145) refer to, the BI team had to come up with a new implementation process to ensure the rapid, low cost and successful implementation of the BI solution. Du Plessis and McDonald (2007:107) therefore developed the DWDW lifecycle model to implement a BI solution in the telecommunications company where the above mentioned challenges exist.

The Double Wave Data Warehouse Lifecycle Model

While implementing the BI solution, the BI team realised that it is not possible to use any of the existing SDLCs, because all of them were very complex to follow, and there was no time to wait for the completion of a BI solution. There were two objectives for this BI project. Firstly the company needed the data almost instantaneously to measure the targets set by government. The second objective was to have an optimised BI solution for on-going reporting and analysis. The implementation of the first objective (instantaneous data) had limited time, while the implementation of the second objective

(optimisation of BI solution) needed a lot of time. The project was therefore divided into two phases. The first phase concentrated on providing the information needed to measure the company against the targets set by government. After the BI solution was running well, the optimisation was done.

Using this "two phased approach" resulted in the development of the Double Wave Data Warehouse (DWDW) Lifecycle Model, where the first phase was called wave 1 and the second phase wave 2 (Du Plessis and McDonald 2007:107). In order to stay in line with all the existing software development projects, the team decided to use the waterfall model of Boehm and Papaccio (1988:103) as a baseline to create the software development steps of the different waves of the DWDW lifecycle model. Figure 1 explains the two waves of the DWDW lifecycle model.

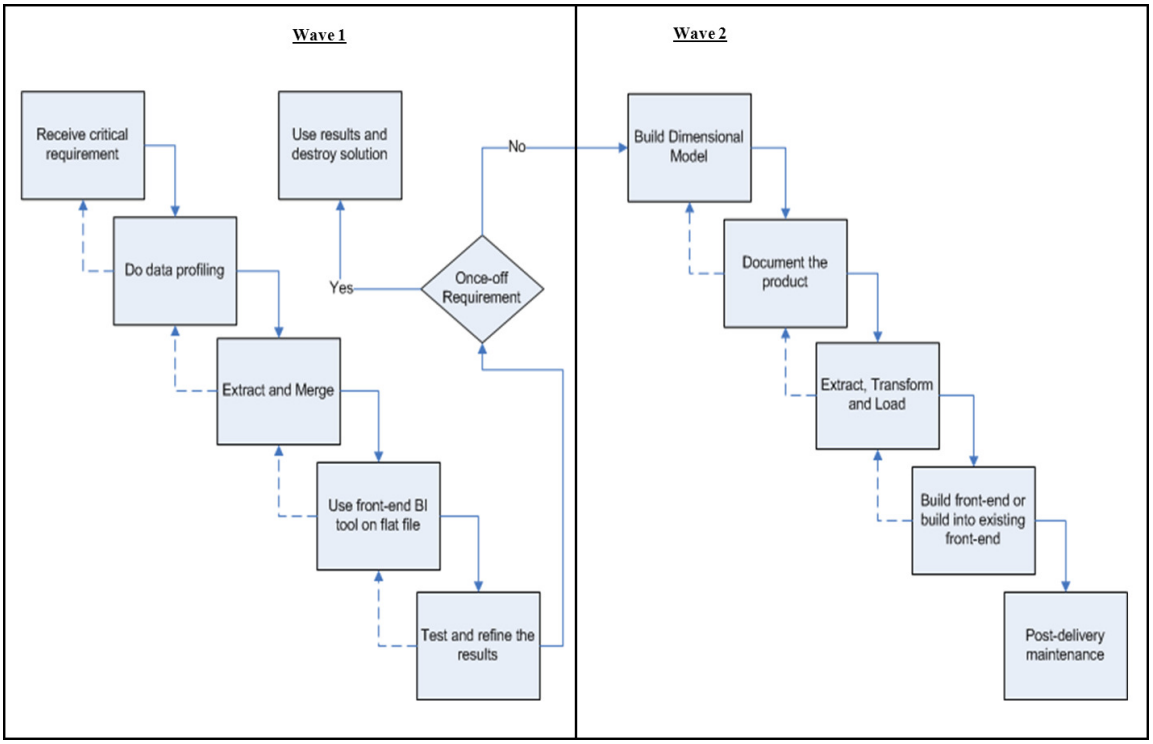

Figure 1. The Double Wave Data Warehouse (DWDW) Lifecycle Model

Wave 1

The first wave of the DWDW lifecycle model concentrates on ensuring immediate answers to the business question. There can be little or no time to wait for a project to finish. In wave 1 there is no focus on the query performance. The main objective here is achieving a result in a day or two. Query optimisation can take months or years, and is therefore left for wave 2.

The database in wave 1 is in a flat file structure. A flat file structure consists of several big tables. The advantage of a flat file structure, according to Hoffman (2003:172-173), is that not much time is needed to build this kind of database. The tables are in a spreadsheet format. Harts (2007:100) states that BI end-users find it much easier to understand and use data stored in spreadsheet format.

The biggest challenge of wave 1 is to find the correct source of information. The users (data capturers) of source systems are normally only familiar with the system screen where the data are entered, and not the tables where they are stored. Data can sometimes be inserted into one table by the front-end screen, and that insert activates triggers or store procedures that insert data in other tables or databases. Finding the correct source can be challenging at times and can result in long hours of data profiling.

Once the right sources are found, the Extract, Merge and Load (EML) process is built. Historically the Extract, Transform and Load (ETL) process was the only process known by data warehouse and BI professionals. The proposed DWDW lifecycle model uses the ETL process only in the second wave. The EML process extracts information from the different sources, merges it into one file, and loads it into a staging area where it can be accessed by a BI front-end tool. The code of the extract and merge parts of the EML process can be reused in the first phase of the ETL process during wave 2.

Wave 1 of the DWDW lifecycle model not only concentrates on producing information quickly, it also creates an easy platform where new business questions lead to a process of building a complete data warehouse, when a large dedicated budget is not available.

Wave 2

During wave 2 in the DWDW lifecycle model, the data warehouse is optimised for query and analysis. This wave concentrates on properly modelling the business requirement into the final data warehouse structure. The structure built in wave 2 is a de-normalised database structure with fact and dimension tables. The biggest challenge of wave 2 is to avoid duplication of dimensions. It might happen that the new source that needs to be added hosts a more complete set of attributes of a specific dimension, for example customer dimensions. Customer information is normally kept in the financial system for payments, and in the order system for placement of orders. If there are some sales that do not go through the order process, there may be customers in the financial system who are not listed in the order system. It sometimes makes sense to load the information from both systems to avoid possible problems.

After the data model is built in wave 2, it is ready for population. Because the right sources were already selected in wave 1, the biggest focus of the ETL process in wave 2 is the “transform” part of the process. Look-up queries should be built to write the foreign keys into the fact table.

Once the data warehouse or new part of the data warehouse is populated, the front-end can be adapted to include the new information. As soon as the end-users are happy, the documentation can be finalised to ensure that on-going maintenance can be done.

Benefits of the DWDW Lifecycle Model

Biffl and Boehm (2006:66) state that a company only invests in software if there will be a return on the investment. When implementing a BI solution, business people want to know what the benefit would be when investing in it. People involved in the implementation of a BI solution sometimes have different or no experiences of BI. Therefore the question could be asked when using the DWDW lifecycle model - how will it be different from other SDLCs used in the industry?

When implementing a BI solution using the older traditional SDLCs, the returns are only apparent right at the end of the project. In some instances it could be months or years, depending on the size of the company. The time it takes to complete the entire project is the time that the returns could have been realised.

Langit (2007:22) states that a BI solution gives the company a competitive advantage. That means that the company can easily identify opportunities. When the implementation of a BI solution takes longer to

finish, some opportunities could be missed. Opportunities in a telecommunications company sometimes become apparent by looking at the buying trends of the customers. If the customer is buying a data modem - a PC or a laptop can also be offered to the specific customer. If the company cannot identify these opportunities, the competition will offer these products to customers who might be lost forever. The company can, by using this opportunity, also attract new customers.

For a company to have the advantage of all these opportunities, it needs a BI solution. When the company is still in the process of implementing the BI solution, frustration develops because it is losing sales and customers. This is while the company is busy investing a fair amount of money into a solution that is not yet bringing any returns on investment.

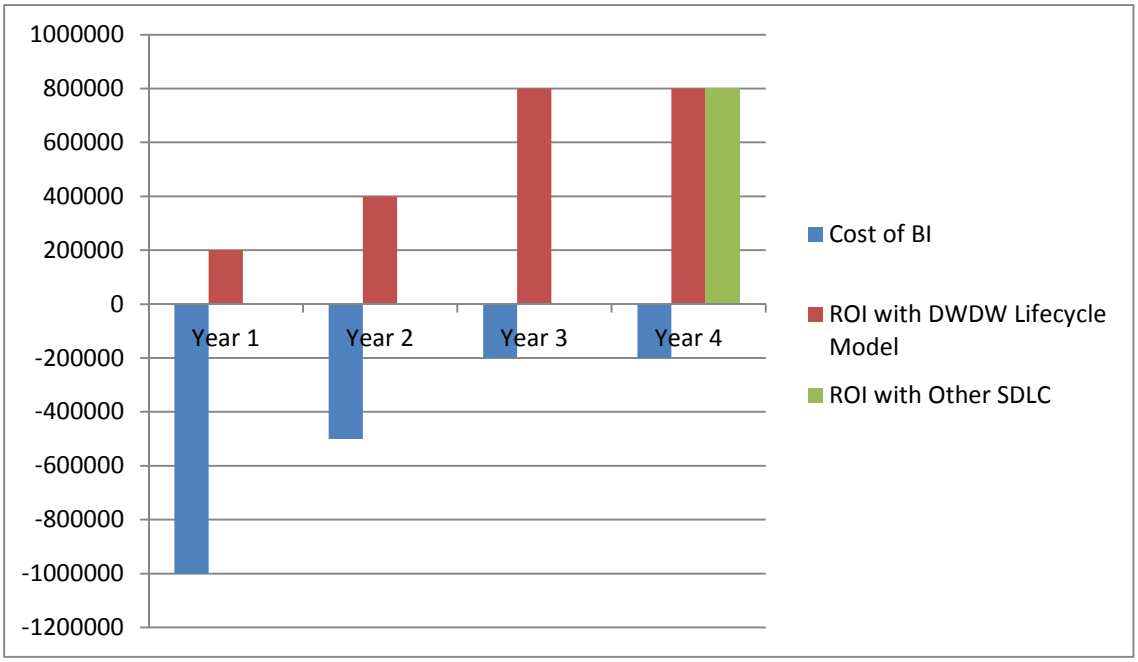

Figure 2. The DWDW Lifecycle Model Deliver Returns on the Investment Much Earlier

Figure 2 shows that the DWDW lifecycle model realises benefits much earlier than the other BI lifecycle model. When using the DWDW lifecycle model, the returns start after the first wave. The initial investment is smaller with the DWDW lifecycle model than with other BI SDLCs if you deduct the ROI from the investment total. During the fourth year the ROI is more than the investment, while with other SDLCs used for BI implementation the breakeven point only appears after the fifth year after implementation (Humphries et al., 1999:45-50).

Different Elements of Business Intelligence Architecture

When looking at the differences of the two waves of the DWDW lifecycle model, the following question presents: Will the hardware architectural requirements be different with the DWDW lifecycle model than with the other BI SDLCs used in the industry? This question will be answered by first looking into the different elements of BI architecture as explained by Ponniah (2001:129). Secondly, the requirements of the DWDW lifecycle model and the limitations of existing architecture discussed by Ponniah (2001:129) will be discussed. Finally a new DWDW architecture will be introduced.

Ponniah (2001:129) classified the different elements of a BI architecture into the areas of data acquisition and data storage.

Data Acquisition

Silvers (2008:139-141) states that data acquisition requires data from different sources of company information, for example from production and financial systems. Some of the sources can be external systems and data files like customer satisfaction survey files. These different data sources, as well as the "extract process" of the ETL element, represent the data acquisition area (Kelly, 2007:235).

Data Storage

The data storage area focuses on loading the selected data into the data warehouse, data mart or data cube. These three data repositories make use of a star schema design. According to Oppel (2009:358) data stored as star schemas consist of a fact table that connects to several dimension tables. Rainardi (2008:71) states that designing a data warehouse, data mart or data cube can be very time consuming and therefore increases the time spent on delivering a BI solution. It is therefore very difficult to respond to critical business questions when following a star schema design. The DWDW lifecycle model wave 1 requires a flat file database. This kind of database does not require any design because the data are only merged and stored. This dramatically speeds up the delivery of the solution to the BI end-user. Optimisation is done at a later stage of the project (wave 2).

Limitations of Existing BI Architecture when Using the DWDW Lifecycle Model

The biggest limitation of the existing BI architecture, when used with the DWDW lifecycle model, exists in the data storage area. Data are stored in a data warehouse or data mart. Both of these data repositories represent a star schema database which takes a lot of time to design and build. When a BI solution is needed urgently for business decisions, there is no time for designing and loading a star schema database.

The Different Environments of an Architecture

Rankins et al. (2003:547) believe that any architecture for software development normally needs at least the following two environments:

- Development environment
- Production environment

Loveland et al. (2005:16) state that it is important that these two environments look the same and use the same software, to eliminate any software compatibility problems. Neely (2006:175) argues that software compatibility refers to how the software system recognises and integrates with other software. Software incompatibility on the same platform can result in functionality that works for example in the development environment, but not in the production environment.

It is very difficult to run only one environment for development and production, because during this process the development environment is often unstable. Developers often change software code and the environment is often restarted because of defects in the code. The architecture only reflects one environment because the other environment is normally identical to the one represented in the architecture. With the DWDW lifecycle model, the development environment consists of a flat file structure and a star schema structure in the data storage area on the same server (see Figure 2). The production environment only has a star schema structure. The elements of the new architecture (DWDW architecture) are discussed in the next section.

The DWDW Architecture

Whilst addressing challenges experienced with the existing architecture of BI when using the DWDW lifecycle model, the new DWDW architecture was created. The name of the new architecture is concordant with the lifecycle model name. The DWDW architecture, like other architectures for software development, requires at least two environments: development and production (see Figure 3). The DWDW architecture differs from other architectures in that the development environment does not comprise the same elements as the production environment. The development environment makes provision for wave 1 and 2 of the DWDW lifecycle model. This means that some of the reports used by the BI end-user are still running in the development environment. The development environment, contrary to other development environments, caters for star schema and flat file databases.

Business people are not aware of the environment the report is running in, except that response times are slower in the development environment than in the production environment. The main reason for slow response is instability of the development environment and because the solution is not yet optimised. However, the focus of wave 1 of the DWDW lifecycle model concentrates on delivering a solution to the BI end-user. Optimisation of the solution only occurs in wave 2.

While the BI solution is used with the front-end connected to the flat file in the development environment, the star schema is built in another database in the same development area. On completion of the database design in the development environment, the database is moved to the production environment. The front-end is then disconnected from the flat file and moved to the new database tables in the production environment.

When moving the star schema of the newly developed solution to the production environment, the new design was joined to the rest of the data warehouse in the production environment through conformed dimensions. Conformed dimensions are two or more dimensions in two or more different data marts or data warehouses that are identical. Kimball and Ross (2002:394) stated that keys and row headers of conformed dimensions need to be exactly the same. Song and LeVan-Shultz (1999:379) argue that conformed dimensions are normally used to analyse facts in different data marts. In this study, however, conformed dimensions were used to link the existing BI solution with the new one. Therefore, the dimension used as a conformed dimension had to exist in the production environment and had to be part of the new BI requirement in the development environment. If the customer dimension was for example needed as the conformed dimension, it was copied from the production environment to the development environment and regularly updated from there during development. This was done to ensure that the two dimensions were always synchronised, in order to smoothen the linking process when moved into production.

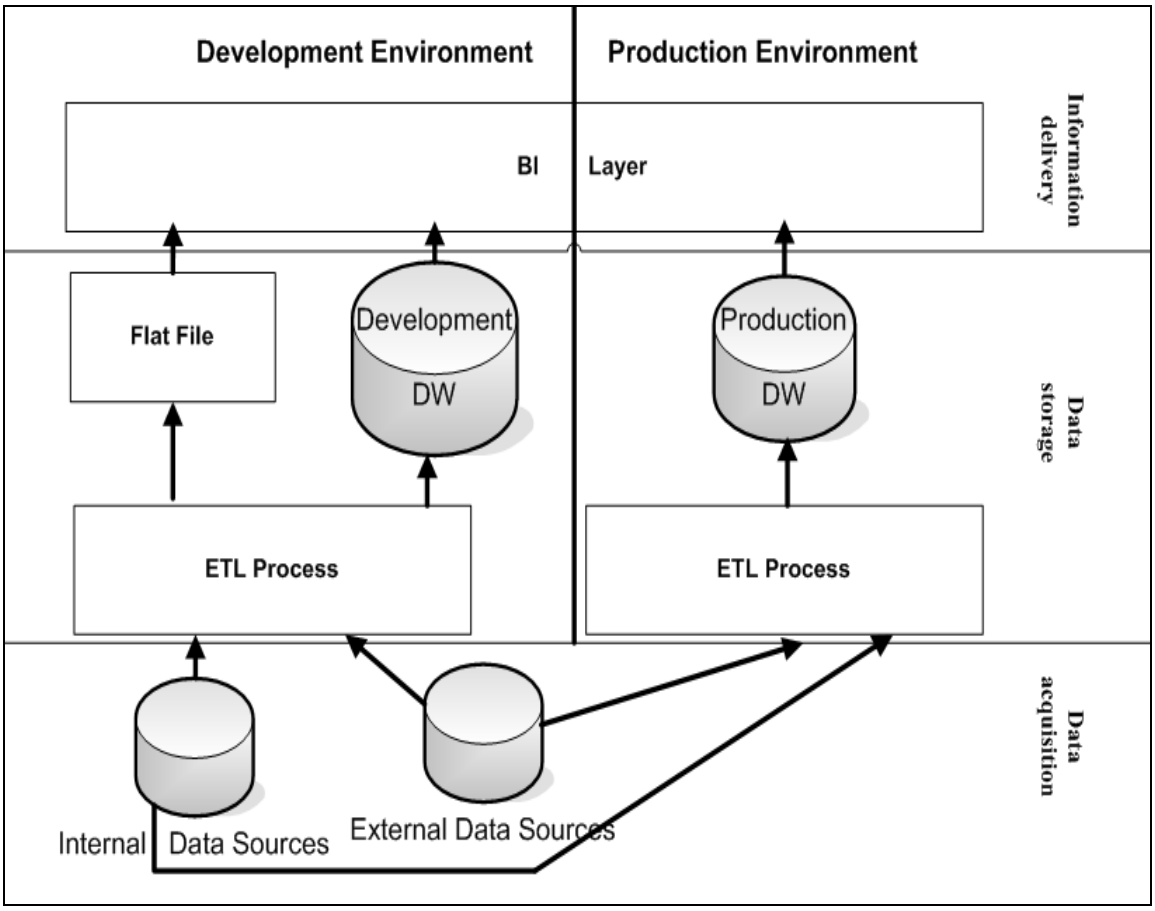

Figure 3. The DWDW Architecture.

Conclusion

This study concludes by looking back to the research question:

- Will the hardware architectural requirements be different with the DWDW lifecycle model than with the other BI SDLCs in the industry?

The research question was answered by firstly considering the BI architectures that exist in the industry. The limitations of these architectures were discussed in detail before the new DWDW architecture was introduced.
The newly developed DWDW Architecture supports the use of the DWDW Lifecycle Model to facilitate the movement of the BI solution between the development and production environments, so preventing any software conflicts.

References

Anon. (1996), 'White paper on telecommunications policy'. [Online], [Retrieved October 22, 2007], http://www.polity.org.za/html/govdocs/white_papers/telewp.html?rebookmark=1

Biffl, S. and Boehm, B. (2006), Value-based software engineering, Springer, New York.
Boehm, B.W. And Papaccio, P.N. (1988), 'Understanding and controlling software cost,' *IEEE Transactions on Software Engineering,* 14 (10), 1462-1477.

Burton, B., Geishecker, L., Schlegel, K., Hostmann, B., Austin, T., Herschel, G., Soejarto, A. and Rayner, N. (2006), 'Business intelligence focus shifts from tactical to strategic,' *Gartner Research,* ID Number: G00139352: 1-5.

Du Plessis, D. and Mc Donald, T. (2007), 'Strategic framework to implement a telecommunications business intelligence solution in a developing country,' Proceedings of the Ninth International Conference on Enterprise Information Systems, Funchal, Portugal, June 12-16, 227-232.

Harts, D. (2007), Microsoft Office 2007 Business Intelligence: Reporting, analysis, and measurement from the desktop, McGraw-Hill, Osborne.

Hoffman, D.R. (2003), Effective database design for geoscience professionals, PennWell Corporation, Tulsa.

Humphries, M., Hawkins, M.W. and Dy, M.C. (1999), Data warehousing: Architecture and implementation. Hall, New Jersey.

Kelly, S. (2007), Data warehousing in action. Wiley, Indianapolis.

Kimball, R. and Ross, M. (2002), The data warehouse toolkit: the complete guide to dimensional modeling, 2nd ed., Wiley, New York.

Langit, L. (2007), Foundations of SQL Server 2005 Business Intelligence, Apress, New York.

Loveland, S., Miller, G., Shannon, M. and Prewitt, R. (2005), Software testing techniques: Finding the defects that matter, Charles River Media, Hingham.

Neely, T.Y. (2006), Information literacy assessment: Standards-based tools and assignments, ALA Editions, New York.

Ooates, B.J. (2008), Researching information systems and computing, Sage, London.

Oppel, A.J. (2009), Databases: A beginner's guide, Mcgraw-Hill, s.l.

Ponniah, P. (2001), Data warehouse fundamentals, Wiley, New York.

Rainardi, V. (2008), Building a data warehouse: With examples in SQL server, Springer, New York.

Rankins, R., Jensen, P. and Bertucci, P. (2003), Microsoft SQL Server 2000 unleashed 2003, Sams, New York.

Silvers, F. (2008), Building and maintaining a data warehouse, CRC Press, Boca Raton.

Song, I. And Levan-Shultz, K. (1999), Data warehouse design for e-commerce environment, Springer, New York.

Sumathi, S. and Sivanandam, S. (2006), Introduction to data mining and its applications, Springer, Heidelberg.

Unhelkar, B. (2008), Practical object oriented analysis. Thomson, Tampa.

Yin, R.K. (2003), Case study research: Design and methods, 3rd ed., Sage, Thousand Oaks, CA.