Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 135 (2018) 587–595

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Computer Science and Computational Intelligence 2018

Development of a University Financial Data Warehouse and its Visualization Tool

Earl Von F. Lapura[a]*, John Kenneth J. Fernandez[a], Mark Jonathan K. Pagatpat[a], Dante D. Dinawanao[b]

[a]*School of Computer Studies, MSU – Iligan Institute of Technology, Andres Bonifacio Avenue, Tibanga, 9200 Iligan City, Philippines*
[b]*Information and Communication Technology Center, MSU – Iligan Institute of Technology, Andres Bonifacio Avenue, Tibanga, 9200 Iligan City, Philippines*

Abstract

In today's data-driven world, organizations which make use of the transactional data they have accumulated over time to come up with a more realistic picture of their operations can make more informed decisions towards attaining their goals and interests. However, due to the huge volumes of these accumulated transactional data, they cannot just be easily and readily used for reporting and analysis purposes. With this, a data warehouse is needed to store these accumulated data obtained from different sources within an organization where other decision-support applications can be built on to guide management decisions. In this study, a financial data warehouse was developed with a multidimensional construct that splits time, finance unit, account, and time dimensions, which is updated periodically with the accumulated transactional data sourced from a financial database of a university, and accessible via a Representational State Transfer application programming interface (REST API). To demonstrate the API's functionalities, we have created a data visualization tool which we integrated into our university web portal and subjected it to usability testing by its target end-users. It was shown that most of the respondents find the tool useful. Also, a query performance test was conducted comparing the execution of certain queries on the source transactional database and on the data warehouse. Result showed that the query time was greatly reduced by an average of over 50% when these queries were executed on the latter.

Keywords: decision support system, data visualization, database systems

* Corresponding author. Telefax: +63-221-4071 local 4281
E-mail address: earlvon.lapura@g.msuiit.edu.ph

10.1016/j.procs.2018.08.229

1. Introduction

As information technology (IT) evolved over time, more and more organizations have become dependent on IT to understand and transact with their clients, and have accumulated huge volumes of transactional data in the process. The International Data Corporation estimates that the enterprise data doubles every 18 months [1]. Moreover, Gartner, an American information technology research and advisory firm, reported in 2010 that enterprise data growth will be 650 percent over the next five years [2].

In today's data-driven world, organizations which make use of the transactional data they have accumulated over time to come up with a more realistic picture of their operations can make more informed decisions towards attaining their goals and interests. However, due to the huge volumes of these accumulated transactional data, they cannot just be easily and readily used for reporting and analysis purposes. With this, a data warehouse is needed where data from different sources in the business process can be transformed in a format ready to be used for reports and analysis [3].

A data warehouse provides value to users for effective management through business intelligence tools that present reports and metrics. One of the latest trends in information technology is the increasing number of visualization-based data discovery tools, estimating a 30 percent compound annual growth rate through 2015 [4].

1.1. Related Works

The following related studies involve the development of data warehouses accessible via a web-based interface:

Deodatta Bhoite published a study in 2004 on analyzing traffic flow patterns and predicting traffic conditions through a spatio-temporal data warehouse with capabilities of analyzing large amount of traffic data collected from sensors all over Minnesota. Built with a web-based, user-friendly interface, the warehouse has a capability to perform ad-hoc queries, the results of which can be visualized using map visualization, bar graphs, heat maps, and line graphs. It employed data mining algorithms, specifically, Apriori algorithm, for discovering association rules, performing clustering analysis and classifying the data. The data warehouse was also optimized using indexes, thereby reducing query times drastically. This work also provided insights in data warehouse architecture and its novel design [5].

Another similar system is a data warehouse, dubbed "Sherlock", which supports air traffic management (ATM) research at NASA's Ames Research Center. Sherlock comprises an Oracle 11g database, a web-based user interface, and supplementary services built on open-source packages for query and visualization. Raw data collected from the National Airspace System (NAS), parsed and processed data, derived data, and reports derived from pre-defined queries are retrieved and stored in a central file system after metadata processing. These data sources are then transformed daily through an ETL process using Pentaho Data Integration into a Star Schema model optimized for adhoc user queries. A web-based interface was then developed using the Oracle Application Express (APEX) to provide data access to the warehouse [6].

1.2. Motivation

The Mindanao State University-Iligan Institute of Technology (MSU-IIT) is a state-owned higher education institution located in Iligan City, Philippines. Like all other well-established organizations, MSU-IIT has become reliant on IT, especially in the management of its financial transactions, and consequently, has accumulated transactional data over the years since it started using its electronic financial management information system (FMIS). In an attempt to support informed decision-making, a financial data visualization application - a subsystem of MSU-IIT's My.IIT university web portal, was developed by Orge, Melecio and Abing [7] in 2011 to produce visual reports out of the transactional data stored in the FMIS database. However, it was removed from My.IIT due to security, usability, utility and performance issues. Moreover, it was implemented using a clone of the entire FMIS database which was difficult to update. Also, the aggregation and specification of values along dimensions take a long time to retrieve for they entail the execution of complex queries on the FMIS database clone. Thus, the development of a financial data warehouse and its visualization tool, is proposed in this study.

2. Methodology

The study was conducted in four (4) phases, namely: (1) planning, (2) data warehouse development, (3) visualization application development, and (4) system evaluation, as shown in Figure 1. They are discussed in detail in the following subsections.

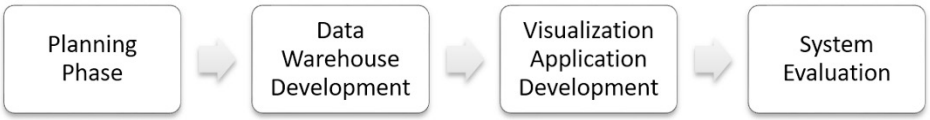

Fig. 1. General Methodology Process of the Study.

2.1. Planning Phase and Elicitation of Financial Data Specifications

This phase focuses on formulating the core objectives of the system determined by evaluation of the problem of current system and its processes. This also includes the determination of the target users and the scope and limitations of the system. This phase also involves further study of related literature and development of skills needed for the development of the system.

The researchers were able to determine the key users of the system aside from the super admin (responsible for authorization rights). The super users or administration officers are authorized to view the aggregate financial data, and the users, whose views are only limited to the finance units under their respective administrative scopes. The administrative officers are concerned with the reports of the allocations, obligations and balances of Institute funds, according to funds, categories, clusters, pap, accounts, cost centres, and combinations thereof. They are also concerned with security and authorization rights given to the users of the system. Cost centre heads were concerned with same type of visualization as to that of the administrative level but is limited to the finance units under their authority. They are also interested in viewing their obligations that are still under process.

The researcher determined through the requirements that the system must be able to view the aggregated financial amounts in terms of allocations, obligations and balances in various dimensions: per fund, per category, per cluster, per pap, per account, and per cost centre. The researchers were also able to find hierarchies among these dimensions in order to form aggregated values. These models are key factors in building the data warehouse.

2.2. Data Warehouse Development

This phase involves the building of the data warehouse, which includes the identification of the purpose of the warehouse, the identification of the data sources and determination of the different formats for those sources, the identification of the dimensions and measures, the determination of the granularity of the data warehouse and the actual design of the data warehouse.

For the study, the Kimball Data Warehouse Model is used, as this advocates a bottom-up approach in building data warehouse, which is suitable for independent repositories such as the implementation of Institute's information systems. Kimball's approach is also deemed nimble, flexible, and easier to build than another known data warehouse approach by Bill Inmon, which supports a top-down approach, calling for a rigorous centralized data warehouse development [8]. The Kimball approach means that output data from the staging area are transformed and stored into individual data marts.

Based on the Kimball Modelling Techniques [3], in building a data warehouse, one must consider the following steps:

- Selecting the business process. Business processes are low-level activities, which users want to analyse performance measurements from. In this study, this involves viewing the aggregated financial amounts in terms of allocations, obligations and balances in various dimensions: per fund, per category, per cluster, per pap, per account, and per cost centre.

- Declaring the grain. This means that each piece of data in the measure of the warehouse that relates to a single row in the fact table must represent a level of detail of measurement. In this system, an individual row in the fact table represents an amount of allocation, expenditure and balance of an individual fund of a basic finance unit in an account category at a certain day.
- Identifying the dimensions. Dimensions are sets of all possible descriptions that take on single values in the context of each measurement. Based on the hierarchies and relationships determined in the requirements elicitation, the researchers are able to determine the following dimensions: 1) fund dimension which divides financial data across the general funds of the Institute, i.e., General Appropriations Act (GAA), Income, and Income Generating Project (IGP); 2) finance unit dimension which divides financial data across the financial units of the Institute, i.e., center, primary cost centers, clusters, programs, or pap; 3) account dimension which divides financial data across the accounts of the Institute; 4) time dimension which divides financial data across time by days, months, quarters, or years.
- Identifying the facts. Facts are the performance measurements resulting from an organization's business process events. The researchers used several attributes as measures: initial allocation, additional allotments, transferred funds to other cost centres, transferred funds from other cost centres, total allocation, paid obligations, unpaid obligations, total obligations and balance.

2.2.1. System Architecture

Based on the requirements and the steps outlined in the Kimball approach, the architecture was formulated as portrayed in Fig. 2. This section provides details on each component of the system, as well as its functions.

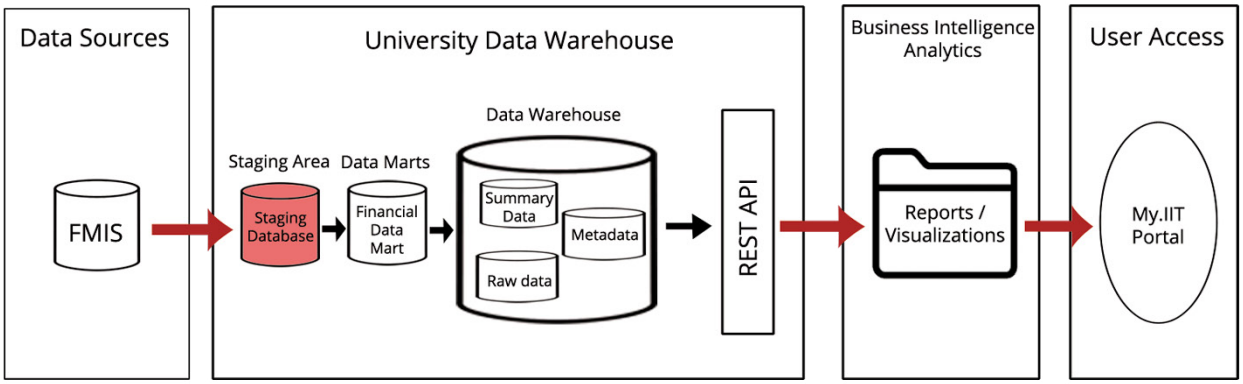

Fig. 2. System architecture diagram showing components of the implemented system

2.2.1.1. Data Sources

The operational database stores financial data of the Institute generated through its Financial Management Information System (FMIS). This database is implemented in a PostgreSQL database server, which is optimized to store inputs in day-to-day transactions, like expenditures, allocations, fund transfers and other financial transactions. Thus, to analyze its information, complex queries across different database schemas must be executed in order to provide comprehensive information.

2.2.1.2. University Data Warehouse

The university data warehouse was developed in order to perform rapid querying and analysis of the MSU-IIT financial data for visualization tools. The financial data can be summarized over various dimensions for multi-view reports. The data warehouse provides the means of storing and aggregating historical financial data

The data warehouse needs periodic loaded data in order to meet its core business processes. To achieve this, the researchers formulated an Extract – Transform – Load (ETL) Tool in order to fetch data from the source database, transform it into a valid format consistent with the requirements and schema of the data warehouse, which will then be loaded or inserted into the data warehouse. The researchers used a freeware platform called Pentaho Data

Integration (or Kettle), which is ideal for the project since it can run parallel processing on multi-core systems, to create an ETL script to obtain a "snapshot" of the fund amounts in the transactional database at a certain time of the day, and subsequently process these data for loading. The Kettle script can be run in a time-based job scheduler called cron in the web server at a daily basis.

At this staging area, the ETL tool extracts data from the data source and store it into a staging database. Data from different schemas at the date which the tool is ran, are consolidated and saved into a temporary PostgreSQL storage, which is dropped after the ETL process.

The financial data mart is then extracted using the ETL process from the consolidated data in the staging area, and transformed according to business rules and data constraints, and stored into permanent tables in the financial data mart.

The researchers formulated the schema outlining all the dimensions and measures, using the star schema dimensional model, as seen in Figure 3. Since the dimension tables are not so large, the star schema is ideal, allowing faster querying with drill-down, roll-up, slicing and dicing operations, due to its denormalized form [9].

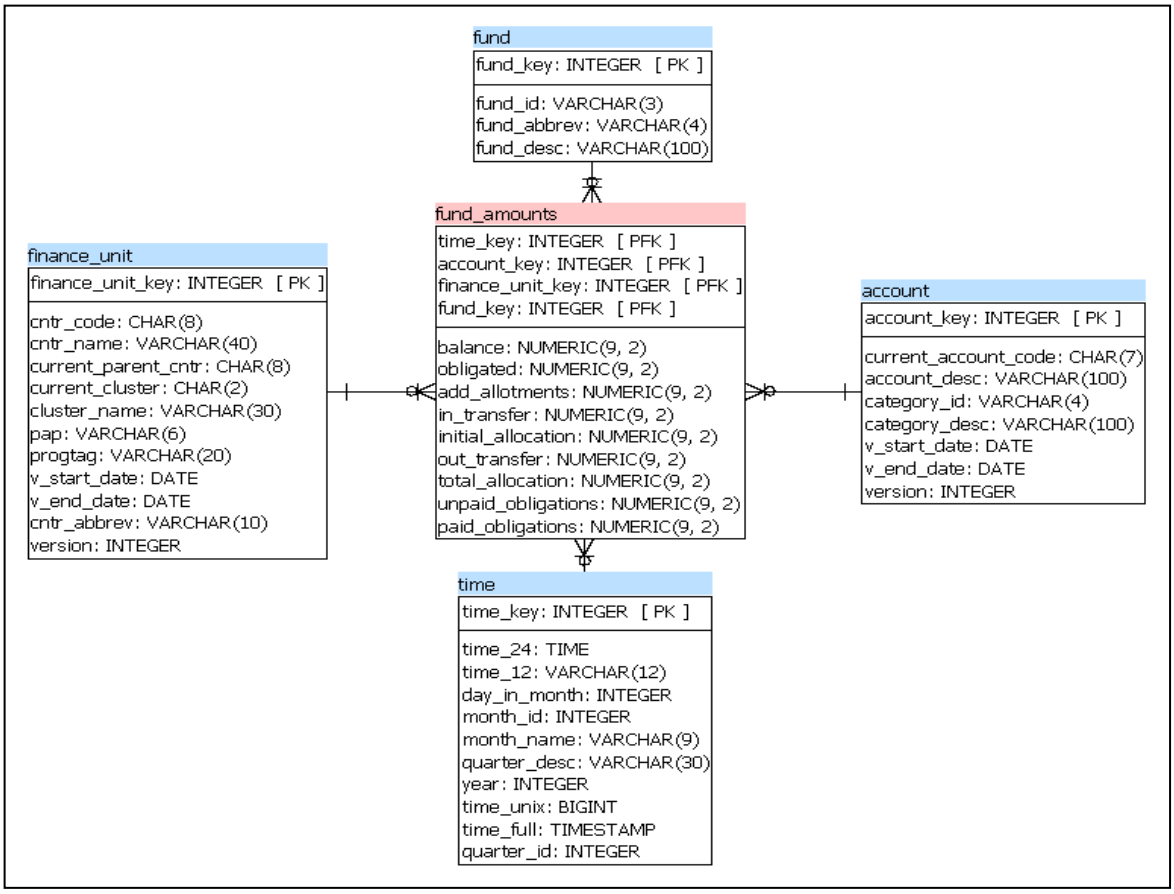

Fig. 3. Star-schema Diagram of the Data Warehouse

The researchers identified the granularity of the data mart to be most optimal for the dimension at the day level, based on user requirements, with enough detail to keep the data interesting, but will not choke up the query performance. The lowest attribute at the time dimension is by day; at the account dimension, by account; at the fund dimension, by fund; and at the finance unit dimension, by cost centre. The possible combinations, including the optimal granularity (highlighted) and the expected amount of rows stored in the data mart are shown in Table 1.

Table 1. Choices for granularity for dimensions with the row in bold denoting the most optimal granularity.

Time Periods per Year	Rows in		
	1 year	5 years	10 years
8765 hours	932105160	4660525800	9321051600
720 half-days	76567680	382838400	765676800
365 days	**38815560**	**194077800**	**388155600**
52 weeks	5529888	27649440	55298880
12 months	1276128	6380640	12761280

All dimensions have been assigned with surrogate primary keys for reasons involving performance and requirement changes. Performance of the query improved since the data types of surrogate keys tend to be more compact, allowing for faster querying even in multiple columns. Moreover, surrogate keys allow for future changes in business keys, making it less expensive for changes in the dimension tables. This type of dimension is called slowly changing dimension, which contain descriptive data that change slowly but unpredictably [10]. The finance unit and account dimensions have been implemented as slowly changing dimensions, due to the possibility of changes in the primary parent center of a particular finance unit, in transferring of cluster membership, or changes in account code system by the government.

To optimize the performance of the data warehouse, the researchers created indexes on the tables. Using the query planner invoked using the EXPLAIN query to determine the number of rows executed by a query, the execution time is reduced when frequently called conditionals on the queries were created with indexes. Primarily, the surrogate keys were created with indexes, which improve greatly the performance of the queries.

Since the financial data of the university is the only subject area in the development of the data warehouse, thus the data warehouse currently has one data mart. Additional data marts that will be developed can then be combined and integrated into the schema.

To serve as an interface through which interactions happen between the data warehouse and the applications using it, the researchers designed an API using with a RESTful architecture, which is a platform-independent service that relies on HTTP protocol in order to read data. This allows the data warehouse to be easily accessed regardless of the type of application, device or programming language used. The researchers used a PHP framework called Slim, which is ideal for API operations due to its lightweight requirements and its specification on easier API programming. It was based on a RESTful architecture for multiplatform capabilities, while employing OAuth2, a de-facto standard for HTTP APIs for security.

2.2.1.3. Business Intelligence Analytics

The stored data in the data warehouse can then be portrayed through meaningful visualizations and report through a business intelligence tool. The development of the web-based application tool is thoroughly discussed in Section 2.3

2.2.1.4. User access

Authorized users can access the application through My.IIT, the university portal of Institute. The integration of the application is discussed in Section 2.3.

2.3. Data Visualization Application Development

The front-end visualization application and the REST API were then developed by the programmer team using an agile software development methodology called extreme programming or XP. The application was built using Bootstrap, a framework using HTML, CSS and JavaScript for developing responsive web projects. A JavaScript charting library, Highcharts, was used to portray data in a graphical manner due to its simple and dynamic chart type support. JQuery, a JavaScript library was also used in order to simplify complicated code, especially in performing nested API calls.

There were several considerations in choosing the visualization scheme for the user interface because it must be both functional and optimal in displaying data, but also invoke a good user experience. It was analysed that the summaries were comparisons by categorical value in nature, and the trends were comparisons by changes of values over time. Thus, it was appropriate to use visualizations that are suited to these natures. The researchers chose the bar chart and pie chart for the summaries and the line chart for the trends. A sample of the charts portrayed in the web application is shown in Figure 4.

The web application is then integrated into the university portal called My.IIT, for the access of its users. The application is then protected by a role-based access control to prevent unauthorized access. A snapshot of this is found in Figure 5.

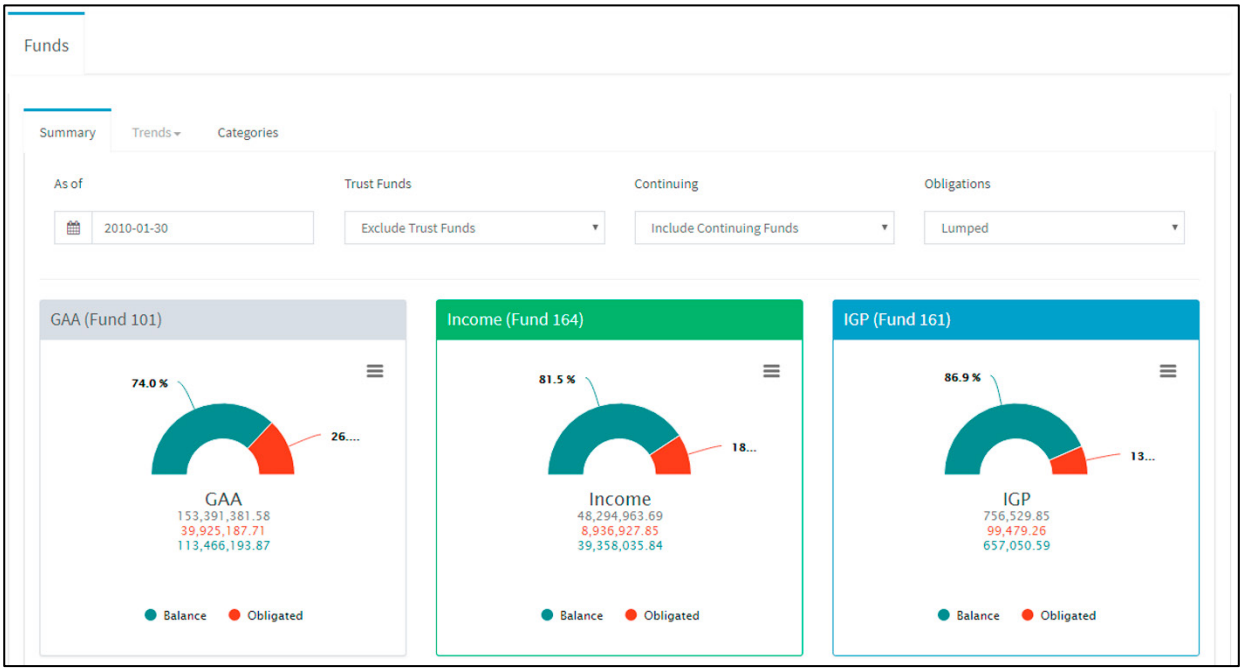

Fig. 4. A snapshot of the web application tool with half-circle aggregate charts

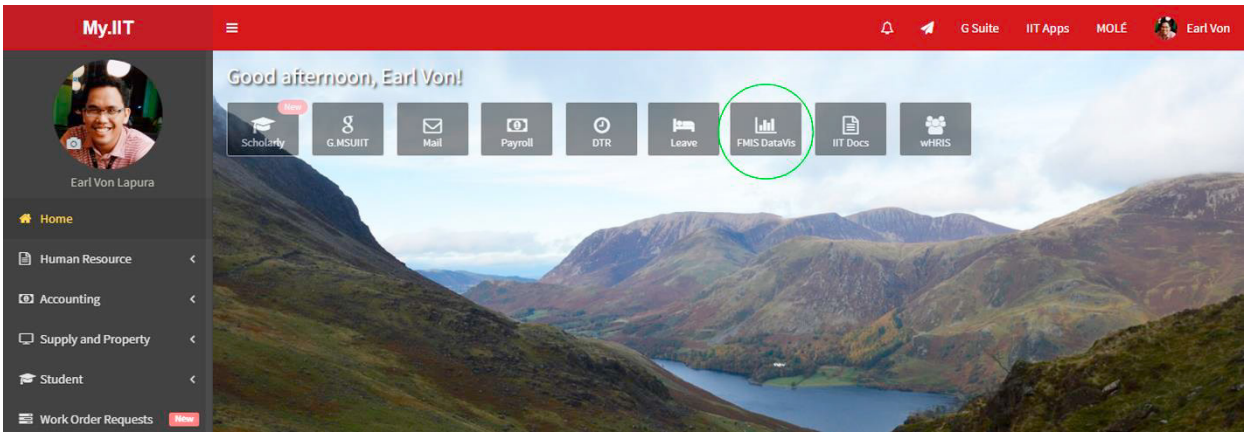

Fig 5. The application as deployed in the My.IIT portal, emphasized by the green circle.

3. System Evaluation

3.1. Query Performance Test

The system was then subjected to a query performance test which would gauge the difference between the corporate database architecture that is currently used for day-to-day transactions and the proposed architecture of the data warehouse for analytical processing. The researchers developed a separate API using the transactional database architecture, which connects directly to the source database. The time spent on fetching data by each function from the database was then captured using the Chrome Developer Tools. The time spent was measured from the moment the resource was requested from the server to the download of the content request. A sample of five API functions was tested for query performance, shown in Table 2. It can be observed that 4 out of 5 functions tested have a percentage reduction of over 50%, with an average of 52.45%.

Table 2. Duration of API calls using the operational corporate database and the implemented data warehouse

API Function	Time Spent Using Architecture (in milliseconds)		Time Reduced (in milliseconds)	Time Reduction (In %)
	Operational Database	Data Warehouse		
get_year()	218	143	75	34.40367
get_costcentername()	5210	1850	3360	64.49136
get_clustername()	406	203	203	50
get_fundsbycostcenters()	404	201	203	50.24752
get_costcenteroverview()	450	166	284	63.11111

3.2. Usability Test for Quality of Use

The system was then subjected to a usability test conducted to determine whether the system has satisfied the quality of use and its attributes. The researchers followed a set of tasks to perform so the user could exhaustively explore the system. The test was conducted to a sample of five users. Three of five users have an experience with analytical and data visualization tools. The users have an average computer literacy level of 4 with 5 as highest. 96.92% of the users were positive on the system's data display quality. Eleven out of thirteen attributes have a complete positive feedback. While 85% of the users were positive on the system's data display quality, it is noted that only 60% of the users had their attention focused on design elements and 80% of the users on the substance of the data. The users agreed unanimously on the ease of use of the tool and perceived ease of use for financial analysts. However, 60% agreed that too many steps were necessary in order to produce the graphs displaying the user's needed data. All of the users commented that the graphs must be displayed immediately after the selection of date, thus eliminating the need of pressing the "Visualize" button. The users also commented that the chosen year must be passed on as parameters for all other pages. Moreover, 60% think that the tool is easy to use for administrators. All users are satisfied with the time needed to display the visualization and agreed that the tool provided the information needed. All users agreed that the graphs displayed needed, reliable, interesting, complete and useful information. Overall, users are satisfied with the content of the information displayed.

4. Conclusions and Future Work

We have developed a data warehouse based on Kimball's approach which stores aggregated data sourced from the transactional database of our university's Financial Management Information System (FMIS). Its design was a result of a series of interviews we conducted among the current users of the FMIS and the would-be users of its envisioned data visualization tool, focusing on what they wanted to see displayed and also of other reports which can readily be generated to support management decision making. Following our university's advocacy on the use of open source software, we have implemented the data warehouse using PostgreSQL and employed a star schema for faster query

performance. Also, the appropriate granularity was determined and its schema was optimized through indexing. By design, the dimensions used can accommodate slowly changing attributes where an ETL process was implemented to extract data from the source transactional database, format them, and then update the data warehouse. Also, a Representational State Transfer application programming interface (REST API) was created to allow access to the data warehouse and enable decision-support applications development. And to demonstrate the API's functionalities, we built a web-based data visualization tool to display the data comparisons and trends using appropriate visualization techniques, such as line and bar graphs.

The web-based data visualization tool was subjected to usability testing by its target end-users (i.e., administrators and financial personnel). It was shown that most of the respondents find the tool useful. Also, a query performance test was conducted comparing the execution of certain queries on the source transactional database and on the data warehouse. Result showed that the query time was greatly reduced by an average of over 50% when these queries were executed on the latter.

In the future, we hope to extend the data warehouse to include other sources of financial data and to create other data mining tools needed for decision-support purposes. We would also like to try other database technologies, such as NoSQL, and see if they will outperform the current implementation.

Acknowledgments

We would like to thank the reviewers for their valuable comments and suggestions for the improvement of this paper.

References

1. Olhorst F. Turning Big Data into Big Money Canada: Wiley; 2012.
2. Paquet R. http://www.gartner.com/. [Online].; 2010 [cited 2015 January 13. Available from: http://www.gartner.com/it/content/1258400/1258425/january_6_techtrends_rpaquet.pdf.
3. Kimball R, Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. New York: Wiley; 2014.
4. Sommer D, Sallam RL, Richardson J. Emerging technology analysis: Visualization-based data discovery tools. ; 2011.
5. Bhoite D. A Traffic Data Warehousing and Visualization Schema. Minnesota: University of Minnesota, Graduate School; 2004.
6. Eshow MM, Lui M, Ranjan S. Architecture and capabilities of a data warehouse for ATM research. In 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC); 2014; Colorado Springs.
7. Orge K, Melecio A, Abing M. Understanding MSU-IIT's Financial Data through Visualization. Undergraduate Thesis. Iligan City: Mindanao State University - Iligan Institute of Technology, School of Computer Studies; 2011.
8. Alsqour M, Matouk K, Owoc ML. A survey of data warehouse architectures — Preliminary results. In 2012 Federated Conference on Computer Science and Information Systems, FedCSIS 2012; 2012; Wrocław, Poland. p. 1121-1126.
9. Oracle. https://docs.oracle.com/. [Online].; 2000 [cited 2015 March 19. Available from: https://docs.oracle.com/cd/A87860_01/doc/server.817/a76994/schemas.htm.
10. Kimball R. Kimball Group. [Online].; 2008 [cited 2015 March 19. Available from: https://www.kimballgroup.com/2008/08/slowly-changing-dimensions/.