Ways to Increase the Efficiency of Information Systems

Ph.D. MANOLE VELICANU
mvelicanu@yahoo.com
Ph.D. Candidate DANIELA LITAN
litan_daniela@yahoo.com
Ph.D. Candidate LARISA COPCEA (TEOHARI)
larisa.copcea@yahoo.com
Ph.D. Candidate MIHAI TEOHARI
mteohari@yahoo.com
Ph.D. Candidate AURA-MIHAELA MOCANU (VIRGOLICI)
mocanuaura@yahoo.com

Ph.D. Candidate IULIA SURUGIU
isurugiu@totalsoft.ro
Ph.D. Candidate OVIDIU RADUTA
ovidiu.raduta@gmail.com

Economic Informatics Department
Academy of Economic Studies
Bucharest
ROMANIA

Abstract: - Fractions of seconds can change a person's life or a company's destiny as we live in a dynamic world, in a permanent move. Thus, the need of being informed, regardless of place or time, is very great. Therefore, the need for efficient information systems is very high. The efficiency can be achieved by using various methods/ techniques and technologies "to build" the information system. The ones, we will present in our paper, are: *methods of Enterprise Application Integration (EAI)*: the purpose of EAI strategies is to create an integrated, complex, coherent software solution, beyond any geographical, social, national or heterogeneous business types limits; *techniques of databases optimization*: in order to improve the performance of database processing can be applied several optimization techniques; and "in terms" of technology, we will consider: *mobile databases*, *spatial databases* and *data warehouses* (using Oracle Warehouse Builder).

Key-Words: - efficiency of information systems, Enterprise Application Integration, database optimization, mobile databases, spatial databases, data warehouse, Oracle Warehouse Builder

1 Introduction

Nowadays companies cannot cope with competition successfully unless they have performing information systems. However, things have not always been like this: in time, as technology advanced, the companies have developed separate information systems in order to simplify some activities which used to be carried out "manually" and thus a medium-sized company started to hold maybe tens of information systems which were not interconnected, thus having difficulties in updating and administering them. On the other hand, as company grew and extended their activities in different parts of the world, other needs emerged: *the need for keeping their employees or at least the strategic management permanently informed and connected to the company's systems, even when they are not at their workplace*; *the need for the strategic management to have easy and very fast access to the company's reports* as „today's managers rely more than ever before on management information systems to provide them with the business data needed to make critical decisions."[1]; *the need to process the information in a very short time*, from various operators simultaneously and the list can go on.

In this paper we are going to present and analyze to what extent the new technologies and methods/techniques: methods of integrating the information systems; techniques of databases optimization; mobile databases; spatial databases; data warehouses, have increased the efficiency of nowadays' information systems.

2 Methods of Enterprise Application Integration (EAI)

EAI is a strong and important issue when discussing about information systems efficiency.

Through software integration we manage to get heterogeneous systems to behave as a whole, connected and complete, prepared to interact and communicate in order to exchange and consume information whenever the need arises. Especially designed for both managerial and operational purposes, integrated software solutions offer the best support tool for any type of business because they combine efficiently all necessary input to achieve any target.

There are several methods to integrate software products. EAI principles subscribe to architectural styles like Service Oriented Architecture (SOA), Web Oriented Architecture (WOA) and Cloud Computing.

SOA is an architectural style that offers the ability to provide functionality in the form of services. These services communicate by messages and contracts, and implement business functionalities, policies and hold states.

SOA refer to a set of architectural principles for building autonomous services that is based on four main concepts.

1. Explicit service boundaries – both access and implementation must obey service contracts.
2. Services are autonomous – services are independently deployed, versioned, secure and available, responsible for the integrity of the business data they manage.
3. Service interaction contracts are based on message formats and message exchange patterns – the only knowledge a service consumer needs in order to use a service is what messages to send and what messages to receive.
4. Services compatibility is based on policies and common standards such as those of the Web Services Interoperability (WS-I).

The main technical advantages of using SOA, once software products are split into independent services, are the usage of Enterprise Service Bus (ESB) for integration, XML for data exchange, different repositories of data sources and code/service reuse facility. Based on SOA approach resulted WOA concepts that enriched integrated services with internet capabilities and cloud computing.

Referring to software architecture and design principles, virtualization and modularization will be the main purpose of integration: integration at the data source level, deployment of independent services, loosely coupled and being able to compound themselves into new complex service structures with minimum effort, binary decoupling using versioning, including business entities and business functionalities.

EAI standards are commonly used to fasten development of complex software projects, they improve development and deployment agility and offer conceptually appropriate solutions, open standards at any level, a complete independence between functionalities exposed as services. Some disadvantages and aspects to be taken care with software integration will be related to a detailed management of dependencies, deployment configurations or development efforts and expected big business and technical analysis. Anyway, EAI projects benefits exceed costs in case of complex implementations, so efficiency is guaranteed.

By definition, EAI refers to "*the process of integrating multiple applications that were independently developed, may use incompatible technology, and remain independently managed*"**.** Within its two basic components, like Business Process Integration (BPI) and Enterprise Information Integration (EII), EAI it's the most commonly and efficient approach in complex software development.

3 Techniques of Databases Optimization

Often, quick access to a database is observed when running *stored procedures*. These play a vital role in enhancing the performance of the database.

One way to optimize table access is using "*nolock*" option; considering that transaction safety is not mandatory for most of the database access, this is apparent in database product that it does not contribute to any record locking capability. Another hint is to use "*nocount*" option. For each operation, stored procedures returns information concerning the number of rows affected. If we turn on nocount option, stored procedures will block row count information so that it will reduce the network overhead occupied with information communication to the user.

Using return values – a query is used to validate information or to return a single value. The returned value of a stored procedure can be used to retrieve a single value from a query and mainly useful when inserting a new record. The information necessary for retrieval is the primary key value. Place "($result)" like argument (eg.: *myData base_free_result($result);*) at the end of the query to utilize the return value in stored.

Another advantageous method for increasing the performance of queries is the *indexing* (efficient indexes). Thus, it will be avoided the necessity of scanning the entire table for results. If you have created an index, database server (DB server) will automatically measures and stores the equivalent statistical information regarding the distribution of values in the indexed column. The optimizer to determine the optimal strategy for evaluating a query uses this statistical information. There are two types of indexes: clustered and non-clustered, each with some unique advantages depending on the data set. A clustered index dictates the storage order of the data in a table. When searched for range of

values the clustered indexes are more efficient on columns of data as the data is sorted. These index types also shine at finding a specific row when the indexed value is unique. Using a non-cluster index, the data is stored in one place and the data value in another place. When a query search for a data value, first the query searches the non-cluster index for the corresponding data value in the table to find the location and then retrieves the data from the location. Most often non-cluster indexes are used for the queries resulting in exact matches.

For analyzing and optimizing the performance of a query the measurement and collection of statistical data is inevitable. Indexes should be kept as slight as possible, to reduce the amount of processing related with each query. The statistical performance data about the index keys are maintained in DB server. If it's configured, non-indexed keys can also be measured statistically. The main statistical concepts, to be understand regarding performance optimization, are: *cardinality* (how many unique values exist in the data set), *density* (is determined by dividing the number of rows that correspond to a given key by the number of rows in the table and measures the uniqueness of values within a data set) *selectivity* (is determined by dividing the number of keys requested by the number of rows they access and measures the number of rows that will be returned by a particular query). To calculate the relative cost of a query plan, the optimizer needs a valid selectivity measure. DB server automatically updates the statistical information periodically as the data in the tables change. Sampling the data, rather than analyzing all of it minimizes the cost of this automatic statistical update.

Optimal performance: designing and choose indexes in a complex database table is normally an inefficient task. DB server has a built-in Index Tuning Wizard to establish an optimal set of statistics and indexes. Thus, running it, a list of suggestions based on a scripted workload will be provided in order to improve the query performance of the database. Generally it is enough to implement only the recommendations of the wizard, applied to a particular situation. But for a dynamic system, it is necessary to update periodically our analysis on optimal database performance.

4 Mobile Databases (Mobile DBs)

As technology keeps developing very fast "day by day", at present office computers are being replaced more and more by much smaller gadgets which are easy to use and transport such as laptops, mobile phones, PDA's (personal digital assistant), smart phones, etc. But „a key contributor to this success is a database's ability to provide a consistent view of shared data across many geographically dispersed users, even in the face of highly concurrent updates at fine granularity."[7]

However, the using of a mobile gadget brings also, some constraints: „*limited resources, limited energy, disconnection, bandwidth asymmetry.*"[www3]

Consequently, it is easy to understand that both applications and databases for mobile gadgets have a number of "specific" features compared to the "classic" ones meant to improve the gadget performance and to increase its efficiency.

"A mobile DB is a database that can be connected to by a mobile computing device over a mobile network. The client and server have wireless connections. A cache is maintained to hold frequent data and transactions so that they are not lost due to connection failure."[www4] „The client application can work with the mobile DB asynchronously, and needs to connect to the central database only when it is necessary to synchronize."[www5] Thus, different from the applications that are using "classic" databases where the connection to the server (where the data base is to be found) is absolutely necessary for the proper functioning of the system, here a permanent connection to the database server is no longer an absolute necessity. This is the main advantage of using applications connected to mobile DBs. "Mobile DBs have benefited through the rapid advancement of global positioning systems (GPS) and geographic information system (GIS), as well as the mobile device itself."[8]

From a thorough analysis of this "trump" of the mobile DBs we can easily infer other benefits:
"*easily protects user privacy* because push applications run mostly at the client machine and client's profile"[9]; "*flexibility and reliability*: asynchronous operation makes the application more flexible and tolerant to network failures"[www5]; *saving on network costs*: the fact that an application that uses mobile DBs can work and than disconnect from the network leads to the reduction of costs caused by the use of the network.

The architecture of a system which uses mobile DBs is made up of "three main parts": a) *fixed hosts*: perform the transaction and data management functions with the help of database servers; b) *mobile units(MU)*: are portable computers that move around a geographical region that includes the cellular network (or "cells") that these units use to communicate to base stations(BS); c) *BS*: are two-way radios, installations in fixed locations, that pass

communications with the MU to and from the fixed hosts: mobile phones, portable phones, or wireless routers."[10] (Fig. 1)

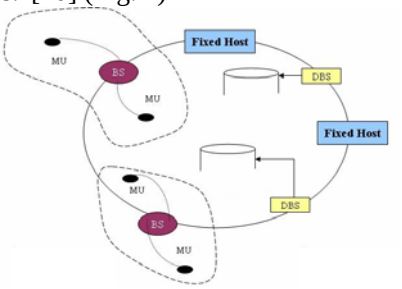

Fig. 1 – Mobile DB System Architecture, Source: [10]

Among the products that are now available on the market for using the mobile DBs technologies, we would like to mention: „*Sybase Inc.'s SQL Anywhere*, IBM's *DB2 Everyplace, Microsoft SQL Server Compact* and *Oracle9i Lite*."[www4]

5 Spatial Databases (Spatial DBs)

The spatial DBs are a recent development in the databases software. They are the bases of the very complex geographical information systems and could be successfully used in many areas: urban planning [11], market research [12] or even in top business such as the implementation of wind parks [13].

Geospatial data means data which describes the location of an object on the surface of the Earth, so a spatially-enabled database must store information about: how the object looks (the representation of the object mainly as points, lines or polygons), where is it located (the longitude and latitude) and in which spatial reference system the data is given. In order to store this information in one single column and to be able to work with it, the database must have defined a special geometry object with several mandatory attributes and methods defined in the Open GIS Consortium (OGC) standard "application objects" [www6]. The main methods defined by OGC for the geometry object are: GetCoordinateReferenceSystem(), GetBoundry(), GetDistance(), GetCentroid().

Oracle database has a special schema named MDSYS where all the spatial objects are stored. The storage format for the spatial data is the Oracle object SDO_GEOMETRY:

CREATE TYPE sdo_geometry AS OBJECT (
SDO_GTYPE NUMBER, **SDO_SRID** NUMBER,
SDO_POINT SDO_POINT_TYPE,
SDO_ELEM_INFO DO_ELEM_INFO_ARRAY,
SDO_ORDINATES DO_ORDINATE_ARRAY);

PostGIS extension for the open source PostgreSQL database stores the spatial object in the standard well-known binary (WKB) format, while an other open source database, MySQL stores it in a proprietary way.

Store the information is not enough, one must retrieve the information also, in real time. That is why the spatially-enabled databases also have defined a special type of indexes, named RTree indexes, used by all the three mentioned spatial DBs. This index approximates each geometry with a Minimum Bounding Rectangle (MBR).

Spatial analysis functions are very important feature of these databases and the one that makes a difference between the databases. For example, the most advanced spatial DBs from this point of view is Oracle Spatial, which offers a special package SDO_SAM with many subprograms which can be used in spatial analysis and data mining, such as: AGGREGATES_FOR_GEOMETRY,
BIN_GEOMETRY,
COLOCATED_REFERENCE_FEATURES,
SIMPLIFY_GEOMETRY, SPATIAL_CLUSTERS.

PostGIS does not have a special package with functions to be used in the spatial analysis but it can be done by writing some smart SQL queries using the spatial functions ST_DISTANCE, ST_INTERSECTS, ST_OVERLAPS [12].

The spatial DBs have become very robust, reliable and more user friendly in recent years, but they still need improvements on all these aspects. New extensions in the SQL standard, oriented towards the spatial analysis, such as CLUSTER_BY, proposed in paper [14], would increase their popularity.

6 Oracle Warehouse Builder (OWB)

OWB is a full-featured data integration, data warehousing, data quality and metadata management solution designed for the Oracle database (Oracle DB). OWB is an integral part of Oracle DB and it is installed as part of every database installation.

6.1 OWB Background

OWB provides data quality, data auditing, fully integrated relational and dimensional modeling, and full life cycle management of data and metadata.

OWB supports data integration and management activities including: *providing extraction, transformation, and loading (ETL); consolidating*

ISBN: 978-960-474-273-8

data from disparate data sources; migrating data from legacy systems; data modeling of relational and dimensional structures; designing and managing corporate metadata; cleaning data to provide quality information; profiling and auditing data quality.

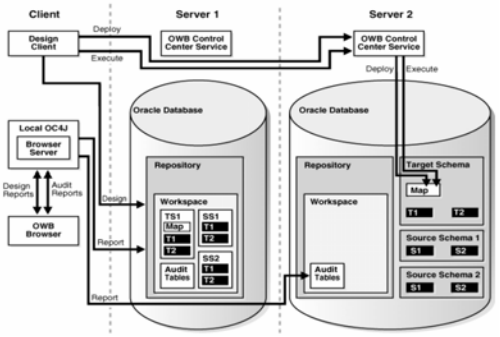

Fig. 2 - OWB Components, Source: [www7]

OWB supports sources from: Oracle DB releases 8.1 and later, databases accessible through Oracle Heterogeneous Services (Gateways), data stores accessible through the Code Templates (which use JDBC) as DB2, DRDA, Informix, SQL Server, Sybase, and Teradata.

The data in the data system that you implement with OWB is stored in target schemas. This data is in the form of data objects such as tables, views, dimensional objects, and cubes.

OWB can be extended to manage metadata specific to any application, and can integrate with new data source and target types, and implement support for new data access mechanisms and platforms.

OWB adds value as a solution for data integration, data movement, and data quality.

6.2 OWB Quality and Security

The core capability of Warehouse Builder is to deliver quality information at the right time to the analytic user.

Data can only be transformed into actionable information when you are confident of its reliability. Before you load data into your target system, you must first understand the structure and the meaning of your data, and then assess the quality.

Using OWB‘s option Data Profiling is good to better understand the quality of your source data. With the Data Profiling Option, you can correct the source data and establish a means to detect and correct errors that may arise in the loading of transformed data.

In the quality assessment phase, you determine the quality of the source data. The first step in this phase is to import the source data into OWB. You can import metadata and data from both Oracle and non-Oracle sources.

After you load the source data, you use data profiling to assess its quality. Data profiling is the process of uncovering data anomalies, inconsistencies, and redundancies by analyzing the content, structure, and relationships within the data.

The quality design phase consists of designing your quality processes. You can specify the legal data within a data object or legal relationships between data objects using data rules.

As part of the quality design phase, you also design the transformations that ensure data quality. These transformations could be mappings that are generated by OWB as a result of data profiling or mappings you create. The quality transformation phase consists of running the correction mappings you designed to correct the source data.

After data has been transformed into quality information, OWB integrates with business intelligence enterprise tools, so that you can easily create reports that transform this information into knowledge. *The organization applies this knowledge to business opportunities and problems to receive the benefits.*

Data transformation is the term for converting data from a source data format into a destination data format. Data transformations typically require two steps: a) *data mapping* (from source to target) to capture any transformations that must occur, and b) *code generation* to create the actual transformation process. After you import your source data and define the target, you decide how to transform the source data into the output desired for the target.

Transformations are PL/SQL functions, procedures, packages, and types that enable you to transform data. You use transformations when designing mappings and process flows that define ETL processes.

Mappings provide a visual representation of the flow of the data and the operations performed on the data. Based on the ETL logic that you define in a mapping, OWB generates the code required to implement your design. OWB can generate code for the following languages: PL/SQL, SQL*Loader, SAP ABAP, Code Templates (CT mappings).

OWB provides a dynamic workspace for your projects and one, common look-and-feel for all editors, including automatic layout, dockable panels, and zoom capabilities.

OWB enables you to design security on the metadata you store in the design repository. The design repository is an Oracle DB with users, roles, and access privileges already defined. OWB

ISBN: 978-960-474-273-8

metadata security operates in addition to the Oracle DB security. The Oracle DB provides security for data while OWB provides security for the metadata.

OWB provides a way to create data auditors, which are processes that provide data monitoring by validating data against a set of data rules to determine which records comply and which do not. Data auditors gather statistical metrics on how well the data in a system complies with a rule by auditing and marking how many errors are occurring against the audited data.

7 Conclusion

The complexity of the information systems used in a company has grown along with its expansion and increase in its volume of sales or along with the increase in their number of employees. So, the gathered information is useful only if it is of dependable quality and is delivered at the right time. In the same time, the need of software integration comes as a must for disconnected businesses that need to interact and the EAI is the tendency to create complex, robust, efficient and finally, complete software solutions. On the other hand: the spatial databases or the mobile databases, usually they are extensions of normal/old databases which contain "special" objects, procedures, mechanisms in order to store and manipulate data, are used for increasing the efficiency of the information systems.

References:

[1] E. Bendoly, E. D. Rosenzweig, J. K. Stratman, The efficient use of enterprise information for strategic advantage: A data envelopment analysis, *Journal of Operations Management*, No.27, 2009, pp. 310-323, (ISI Thomson).

[2] A. N. K. Chen, S. Sen, B. B. M. Shao, Strategies for effective Web services adoption for dynamic e-businesses, *Decision Support Systems*, No.42, 2006, pp. 789-809, (ISI Thomson).

[3] S. Pastore, The service discovery methods issue: A Web services UDDI specification framework integrated in a Grid environment, *Journal of Network and Computer Applications*, No.31, 2008, pp. 93–107, (ISI Thomson).

[4] S. Chandra, J. de Vries, J. Field, H. Hess, M. Kalidasan, K. V. Raghavan, F. Nieuwerth, G. Ramalingam, J. Xue, Using logical data models for understanding and transforming legacy business applications, *IBM Systems Journal*, Vol. 45, No.3, 2006, pp. 647-655, (ISI Thomson).

[5] J. Stephens, C. Russell, *Beginning MySQL Database Design and Optimization: From Novice to Professional*, Appress, 2004.

[6] N. T. Nguyen, M. T. Le, J. Swiatek, *Intelligent Information and Database Systems*, Springer-Verlag, Berlin Heidelberg, 2010.

[7] N. Tolia, M. Satyanarayanan, A. Wolbach, Improving mobile database access over wide-area networks without degrading consistency, *MobiSys'07*, Puerto Rico, USA, 2007, pp. 71-84.

[8] K. Xuan, G. Zhao, D. Taniar, W. Rahayu, M. Safar, B. Srinivasan, Voronoi-based range and continuous range query processing in mobile databases, *Journal of Computer and System Sciences*, Article in Press (Received in revised form 11 Jan. 2010), (ISI Thomson).

[9] V. Kumar, *Mobile Database Systems*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.

[10] B. K. Mohanty, Mobile Database Systems, *Computer Science and Engineering*, 2008.

[11] A. Tsouchlaraki, G. Achilleos, Z. Nasioula, A. Nikolidakis, Designing and creating a database for the environmental quality of urban roads, using GIS, *7th WSEAS Internat. Conf. on Environment, Ecosystems and Development*, Tenerife, Spain, 2009, pp. 109-113.

[12] D. Litan, A. M. Mocanu (Virgolici), S. Olaru, A. Apostu, Modern information technologies used in market research, *9th WSEAS Internat. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics*, Merida, Venezuela, 2010, pp. 245-250.

[13] A. Bara, A. Velicanu, I. Lungu, I. Botha, Natural factors that can affect wind parks and possible implementation solutions in a Geographic Information System, *The Internat. Conf. on Development, Energy, Environment, Economics*, Tenerife, Spain, 2010, pp. 50-54.

[14] C. Zhang, Y. Huang, Cluster By: a new sql extension for spatial data aggregation, *15th annual ACM internat. symposium on Advances in geographic information systems ACM*, New York, 2007.

[15] B. Griesemer, *Oracle Warehouse Builder 11g Getting Started*, Packt Publishing, 2009.

[16] M. Teohari, L. Copcea, XML Authoring Tool, *4th European Computing Conf.*, Bucharest, Romania, 2010, pp. 143-147.

[www1] www.integrationconsortium.org

[www2] www.ws-i.org

[www3] http://blizzard.cs.uwaterloo.ca/eaoliver

[www4] en.wikipedia.org/wiki/Mobile_database

[www5] http://utopia.duth.gr/~pefraimi

[www6] www.opengeospatial.org

[www7] www.otn.oracle.com