Some Information Technologies to Improve the Performance of an ERP System

Ph.D. Candidate DANIELA LITAN
litan_daniela@yahoo.com
Ph.D. Candidate ANCA APOSTU
ancaiapostu@gmail.com
Ph.D. Candidate LARISA COPCEA (TEOHARI)
larisa.copcea@yahoo.com
Ph.D. Candidate MIHAI TEOHARI
mteohari@yahoo.com
Economic Informatics Department
Academy of Economic Studies
Bucharest
ROMANIA

Abstract: - ERP (Enterprise Resource Planning) implementation differs in organizations structured into nearly independent business units, as they will each have different processes, business rules, data semantics, authorization hierarchies, every-day business operations, decision-making criterias and decision centers. In order to be able to respond to today's customer needs, software designers are struggling to provide *custom-integrated solutions* that would ensure standard system's requirements and also a platform for customized solutions. Still, customizing an ERP package can prove to be expensive and resource consuming and can influence the implementation of the standard benefits of an integrated system. Nevertheless, customizing an ERP suite gives the scope to implement smart and new solutions for excellence in specific areas while ensuring that industry best practices are achieved in less sensitive areas. In this paper, we will proceed to the analysis, presentation and comparison of the current most important technologies (like information methodologies and data warehouses) which are being used in development of the ERP systems.

Key-Words: - Enterprise Resource Planning, information technology, information methodology, information system, data warehouse, query optimization

1 Introduction

The acronym ERP (Enterprise Resource Planning) was first used in 1990 by firm Gartner Group in connection to MRP (Material Requirements Planning, later Manufacturing Resource Planning) and CIM (Computer-Integrated Manufacturing) and since then it has become widely used for streamlining *not only manufacturing process but all other business processes*. ERP packages integrate solutions for different directions such as accounting, contracts, payroll, maintenance and human resources management, attempting to provide technology solutions for all core functions of an enterprise, regardless of its speciffic business flow, such as non-manufacturing businesses, non-profit organizations and governments.

There are two main areas in which successfully implemented ERP systems proove their results: the first refers to *operational efficiency* and increased *productivity through automated transactions* offering improved support for decision-making. The second aspect is regarding *improved customer service* taking into consideration the market performance. Both types of ERP benefits are generally expected to result in an improved competitive position, low production cost, expectations of revenue growth, ability to compete globally and the desire to re-engineer the business to respond to market challenges.

But, on the one hand the rigidity and the difficulty of adapting the ERP systems to the specific business requirements of the companies, and on the other hand the complexity of the packages, the hardware and software necessary resources for implementing ERP systems can be very expensive, especially if customization is needed, are two of the main causes for failure in ERP systems implementation. Therefore a number of companies prefer to develop their own ERP systems.

The organization which plans to develop and "implement ERP project, usually employs multiple experts from different sections in selection process"[1] because the development and implementation of an ERP system requires first a *business analysis* for establishing the flow of activities and speciffic documents, the necessities of the customer and the needed customization for a normal development of the business flow; second, *the methods and models* for developing, implementing and maintaining the ERP system are discussed in order to integrate the standard and the customized functionalities and to provide a unitary solution.

Next we will precede to the analysis, presentation and comparison of the current most important information methodologies (for business analysis) and databases/ data warehouses which are being used in development of the ERP systems.

2 Methodologies for Development of the Information Systems

"The increased exigencies on the quality of information systems, their importance in improving the performances of an organization have required that well structured software development and design strategies be defined and used, which should have priority to any activity related to information programs developing."[2] Thus "information modeling, as a part of requirement management process, is mainly dealing with capturing user needs and understanding system complexity."[3] But, despite this, the currently used analysis and design methodologies "can not handle the uncertain situations, communications with users or superior management development"[www2], that is why in practice, the methodologies are partially used, as a result of their deficiencies.

The methodologies are used to indicate the development manner of the design and analysis process of the information systems, "establishing:

- components of the information system execution process (stages, sub-stages, activities, operations) and their contents;
- running (execution) flow of the components; methods, techniques, procedures, instruments, utilized norms and standards" [www3].

Moreover, the execution methodologies of the information systems comprise:

- "approaching method of the systems, in order to clarify the relationship between the variations of the system and its dynamism;
- data formalization and editing processes;
- instruments for documentation conception, execution and drafting;
- project performance method and the specific actions of each stage (life cycle);
- defining of the working method, analysts and designers' role and the relationship between them;
- project management methods (planning, programming, follow-up)."[www3]

Although it seems unbelievable, in the "software world" only a low percentage of the projects on information systems development are successfully completed. „After two decades of this problem reoccurring, one of the leading causes for the high failure rate is still poor process modeling (requirements' specification). Therefore both researchers and practitioners recognize the importance of business process modeling in understanding and designing accurate software systems."[4]

2.1 Comparative Approach of Some Information Methodologies

Development of big systems, such as ERP type systems or e-government type portals require the usage of information methodology (for analysis and development) early at the beginning of the development cycle of the new system. Though, "conceptual modeling is central to information systems development"[5], in practice, this stage is most of the times skipped in development of small projects. This is the main reason why currently a great deal of the projects end up in a failure or with clients' requirement for modifications of the final products.

Because "many methodologies have been developed to improve business processes and to investigate how well current business processes have achieved their goals"[6], before taking a decision regarding the best methodology to use, we hereby propose an analysis and comparison of the main information methodologies.

As we can see in Table 1 and Fig. 1, we can easily notice that each methodology has both advantages and disadvantages:

- *SSADM* (The Structured Systems Analysis and Design Method): "It is a detailed method, covering almost all elements of the information system"[www2];
- *UML* (The Unified Modeling Language): "It is an expressive modeling language, covering all development elements of the information system"[www2];
- *SSM* (Soft Systems Methodology): "Whereas most of the methodologies deal only with soft

issues, SSM covers the hard system, as well. SSM supports the activities and processing, creating a conceptual representation pattern of the source activities"[www2];

- *MERISE* (Methode d'Etude et Realisation Informatique par le Sous – Ensemble representatif): this methodology allows:
  - "closeness to the information system and to the ideal structure of a database;
  - a three level description of the information system;
  - utilization of precise, simple and exact representation *formalism* for data description. (*Formalism,* as used above, means a group of definitions and rules, combined with a group of diagrams and/or tables). This formalism is internationally regulated by ISO standard under ENTITY-ASSOCIATION name;
  - detailed description at conceptual level, allowing the creation of an information system independent of the organization of the firm and choice of the automation technique;
  - the visual representation used in the conceptual pattern facilitates the dialogue among all partners involved in the creation of the information system."[www4]
- *STRADIS* (Structured Analysis, Design and Implementation of Information Systems): aceasta metodologie este bazata pe metoda descompunerii funcționale top-down şi utilizarea Data Flow Diagrams;
- *YSM* (Yourdon Systems Method): this methodology is similar to STRADIS methodology, "its approach being focused upon the data structures importance"[www2].

Yet, if we are to draw a line a provide a "verdict" concerning the choice of one of the methodologies in order to develop an information system, this would prove impossible to do, without taking into consideration the type of the information system that is to be developed, the size of the future system, the time allotted for the entire project (see Table 1 – utilization of some methodologies require more time than others) or even the members of the teams or their competencies in charge with the future information system (see Fig. 1).

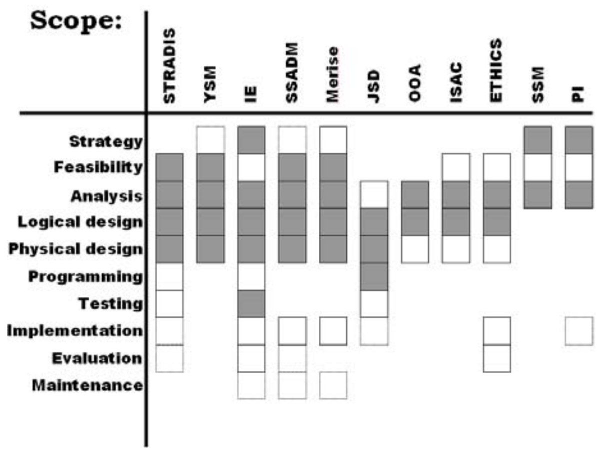

Fig. 1 - Purpose of the information methodologies, source: [www5]

Table 1 - Comparative Development Methodologies
DFDs = Data Flow Diagrams

	SSM	IE	STRADIS	YSM	SSADM	MERISE	RUP
Objectives	aims at much more than developing an IT system	have clear objectives to develop computerized information systems					
Domain	address general planning, organization, and strategy of information and systems in the organization		are classified as specific problem-solving methodologies				
Target	more applicable in human activity 'messy' situations	designed for large systems	general-purpose, DFDs not suitable for management information systems or web-based systems	designed for large systems			general-purpose, not very useful for small systems
Model	DFDs (play a less significant role than in STRADIS)	integrate both processes and data	uses primarily DFDs	DFDs (play a less significant role than in STRADIS)	integrate both processes and data		
Techniques	does not heavily use techniques and tools	explicitly suggests that the techniques are not a fundamental part of the methodology	is largely described in terms of its techniques	specify techniques and view them as important for the methodology		ranges from simple drawing tools to those which support the whole development process, including prototyping, project management and code generation	specify techniques and view them as important for the methodology
Product	comes only with some academic papers	some online specs	some online specs	some online specs	comes with a large set of manuals	has a range of books, and online specs	has a range of books, and online specs
Practice	• academic origin • both business and technical people	• commercial origin • professional technical developers				• academic origin • professional technical developers	

3 Information Systems Using Data Warehouses

A data warehouse is a relational database that is designed for query and analysis. "The data warehouse is, in fact, a database used for decision-making, totally separate from the operational database of a company."[7] It usually contains historical data derived from transaction data, and can include data from multiple sources.

A data warehouse environment can include an extraction, transportation, transformation, and loading (ETL) solution, analysis, reporting, data mining capabilities, other applications that manage the process of gathering data, transforming it into useful information to business users.

A classic approach to providing a single version of the truth, where multiple transactional applications exist, is to build a data warehouse. This is a common practice where the goal is to store and analyze years of transactional history and where data quality in source systems is a known issue. However, alternative integration strategies are sometimes used where only recent transactions are needed for business intelligence.

Oracle Warehouse Builder (OWB) is an infrastructure builders' tool most often used during the design and deployment of data warehouses. The data warehouse design might include schema in third normal form, star schema, OLAP cubes, or as a hybrid schema of multiple types. OWB provides an interface to also define the source to target data extraction, transformation and load (ETL) mappings, generate the ETL script, coordinate workflows, and perform metadata management. OWB-generated scripts can pull data from other Oracle databases, other relational sources accessible via ODBC or Oracle Transparent Gateways, and from flat files with fixed width or delimited columns. Integrators are available to enable ease in building extractions from source tables in the Oracle E-Business Suite, PeopleSoft, or SAP applications. The generated scripts can be scheduled using Oracle Enterprise Manager or other popular schedulers and can leverage Oracle Workflow.

If you decide that data warehouses offered as applications (i.e.Oracle/PeopleSoft EPM, Oracle/Siebel Business Analytics Application) are not a good match for your business you may determine it is better to build a custom data warehouse.

After the data warehouse is up and working we would go to try to obtain a better data warehouse performance and how other way than by optimizing.

One of the most common causes of poor performance is poor application or data warehouse design. Such designs should provide optimal storage and maintainability and access to the underlying data in a flexible and highly performing manner. To achieve these design goals, the architect and administrator leverage modeling approaches such as star schema and hybrid designs and utilize data warehousing features such as bitmap indexes, partitioning, and materialized views.

3.1 Data Warehouse Performance - Query Rewrite

When tables contain large amounts of data, it is expensive and time consuming (queries can take minutes or even hours) to compute the required aggregates or to do joins between these tables. In such cases, a data warehouse employs a powerful process called query rewrite to quickly answer the query. A query undergoes several checks to determine whether it is a candidate for query rewrite.

The first method used by the optimizer to recognize when or if to rewrite a query is based on matching the SQL text of the query with the SQL text of the materialized view definition.

The simplest case to rewrite a query occurs when the result stored in a materialized view exactly matches what is requested by a query, but the number of queries eligible for this type of query rewrite is minimal. The optimizer does this type of identification by comparing the text of the query with the text of the materialized view definition.

For the text match check, the optimizer uses two methods:

- *Full text match:* In full text match, the entire text of a query is compared with the entire text of a materialized view definition (that is, the entire SELECT expression).
- *Partial text match:* When full text match fails, the optimizer attempts a partial text match. In this method, the text starting with the FROM clause of a query is compared with the text starting with the FROM clause of a materialized view definition.

Example, if the query requests AVG(expr), the materialized view's SELECT list must contain either AVG(expr), or SUM(expr) and COUNT(expr).

If the first method fails, the optimizer uses a more general method in which it compares joins, selections, data columns, grouping columns and aggregate functions between the query and materialized views.

The optimizer uses data relationships on which it can depend. For example, primary key and foreign key relationships tell the optimizer that each row in the foreign key table joins with at the most one row in the primary key table. Data relationships are important for query rewrite because they show what type of result is produced by joins, grouping, or aggregation of data.

The data warehouse optimizes the input query with and without rewrite and selects the least costly alternative. The optimizer rewrites a query by rewriting one or more query blocks, one at a time.

Query rewrite operates on queries and subqueries in the following types of SQL statements: SELECT, CREATE TABLE … AS SELECT, INSERT INTO … SELECT. Query rewrite operates on subqueries in data manipulation language (DML) statements: INSERT,DELETE,UPDATE. It also operates on subqueries in set operators: UNION, UNION ALL, INTERSECT, and MINUS.

Because query rewrite occurs transparently, it is not always evident that it has taken place. Of course, if the query runs faster, rewrite should have occurred; but that is not proof. In Oracle there are two ways to confirm that the query rewrite has occurred:

- Use the EXPLAIN PLAN statement and check whether the OBJECT_NAME column contains the name of a materialized view.
- Use the DBMS_MVIEW.EXPLAIN_REWRITE procedure to see whether a query is rewritten or not.

When a query is rewritten, Oracle's cost-based optimizer compares the cost of the rewritten query and original query and chooses the cheaper execution plan. Oracle Database optimizes the input query with and without rewrite and selects the least costly alternative. The optimizer rewrites a query by rewriting one or more query blocks, one at a time. If query rewrite has a choice between several materialized views to rewrite a query block, it selects the ones which can result in reading in the least amount of data. After a materialized view has been selected for a rewrite, the optimizer then tests whether the rewritten query can be rewritten further with other materialized views. This process continues until no further rewrites are possible. Then the rewritten query is optimized and the original query is optimized. The optimizer compares these two optimizations and selects the least costly alternative.

The graphic below shows the cost-based approach used during the rewrite process:

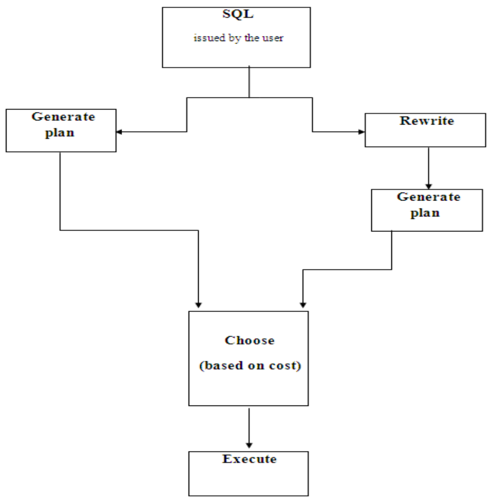

Fig. 2 - The cost-based approach used during the rewrite process

3.2 Data Warehouse Performance – Example of the Query Rewrite

Consider the following materialized view, month_sales_mv, which provides an aggregation of the euro amount sold in every month:

```
CREATE MATERIALIZED VIEW
month_sales_mv
ENABLE QUERY REWRITE AS
SELECT t.year_month_desc,
SUM(s.amount_sold) AS euros
FROM sales s, times t
WHERE s.time_id = t.time_id
GROUP BY t.year_month_desc;
```

The number of sales in the store for a month is around several millions and we created the materialized aggregate view above that has the precomputed aggregates for the euro amount sold for each month.

The following query, asks for the sum of the amount sold at the store for each year month:

```
SELECT t.year_month_desc,
SUM(s.amount_sold)
FROM sales s, times t
WHERE s.time_id = t.time_id
GROUP BY t.year_month_desc;
```

In the absence of the previous materialized view and query rewrite feature, the data warehouse will have to access the sales table directly and compute the sum of the amount sold to return the results. This involves reading many million rows from the sales table which will invariably increase the query response time due to the disk access. The join in the query will also further slow down the query response as the join needs to be computed on many million rows. In the presence of the materialized view month_sales_mv, query rewrite will transparently rewrite the previous query into the following query:

```
SELECT year_month, euros
FROM month_sales_mv;
```

Because there are only a few rows in the materialized view month_sales_mv and no joins, the data warehouse returns results instantly. This simple example illustrates the power of query rewrite (optimization) with materialized views in a data warehouse.

4 Conclusion

The major challenge during the implementation of an ERP is finding the appropriate balance between the standard set of functionalities and the necessary customization for adapting the system to the customer's needs. Problems with ERP systems are mainly due to the fact that customization of the ERP software is limited. The process of re-engineering of business flows in order to ensure compatibility with the standard version of the ERP applications may lead to a loss of competitive advantage. Therefore, companies prefer to develop their own ERP systems, using the latest information technologies to achieve a more efficient system.

References:

[1] X. Liao, Y. Li, B. Lu, A model for selecting an ERP system based on linguistic information processing, *Information Systems*, No.32, 2007, pp. 1005-1017, (ISI Thomson).

[2] G. Cucui, Considerations on software engineers and project management of ICT, *Annales Universitatis Apulensis Series Oeconomica* („1 Decembrie 1918" University, Science Faculty, Alba Iulia, Romania), Vol.2, No.8, 2006.

[3] K. Siau, Y. Wang, Cognitive evaluation of information modeling methods, *Information and Software Technology*, No.49, 2007, pp. 455-474 , (ISI Thomson).

[4] J. Barjis, The importance of business process modeling in software systems design, *Science of Computer Programming*, No.71, 2008, pp. 73-87, (ISI Thomson).

[5] C. Flender, Ecological modelling of information systems, *The 20th International Conference on Advanced Information Systems Engineering*, Montpellier, France, June 17th, 2008.

[6] F. Aburub, M. Odeh, I. Beeson, Modelling non-functional requirements of business processes, *Information and Software Technology*, No.49, 2007, pp. 1162-1171, (ISI Thomson).

[7] A. Mocanu, D. Litan, S. Olaru, A. Munteanu, Information Systems in the Knowledge Based Economy, *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, Issue 1, Vol. 7, 2010, pp. 11-21.

[8] M. Teohari, L. Copcea, XML Authoring Tool, *Proceedings of the 4th European Computing Conference (ECC'10)*, 2010, pp. 143-147.

[9] R. Stackowiak, J. Rayman, R. Greenwald, *Oracle data warehousing and business intelligence solutions*, Wiley Publishing, Inc., Canada, 2009.

[10] R. Wrembel, C. Koncilia, *Data warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press Hershey, Germany, 2009.

[www1] http://www.erp.com

[www2] I. Chivu, D. Popescu, The effects of implementing SSADM (Structured System Analysis and Design Methodology) in the management process of European public institutions - British experience in the use of SSADM, Available: http://www.miteapl.ro/r1/studiu-1.pdf

[www3] Information Systems and Information Applications in Business Administration, „Lucian Blaga" University from Sibiu, Faculty of Economic Sciences - lecture notes, Available: http://webspace.ulbsibiu.ro/florin.martin

[www4] http://www.icbe-ct.com/pacuraru

[www5] Information Systems Development Methodologies: Making the right choice, Available: http://www.cems.uwe.ac.uk

[www6] www.otn.oracle.com