Contact
Calea Dorobanților, no. 15-17, room 2017, Bucharest, Romania
Web: http://dbjournal.ro
E-mail: editor@dbjournal.ro

Contents:

Architecture of Automated Database Tuning Using SGA Parameters

Hitesh KUMAR SHARMA[1], Aditya SHASTRI[2], Ranjit BISWAS[3]
[1]Assistant Professor, ITM University
[2]Vice-Chancellor, Banasthali University, Rajasthan-304022, India
[3]Head and Professor,CSE Dept, Jamia Hamdard (Hamdard University)
hiteshsharma@itmindia.edu, adityashastri@yahoo.com, ranjitbiswas@yahoo.com

Business Data always growth from kilo byte, mega byte, giga byte, tera byte, peta byte, and so far. There is no way to avoid this increasing rate of data till business still running. Because of this issue, database tuning be critical part of a information system. Tuning a database in a cost-effective manner is a growing challenge. The total cost of ownership (TCO) of information technology needs to be significantly reduced by minimizing people costs. In fact, mistakes in operations and administration of information systems are the single most reasons for system outage and unacceptable performance [3]. One way of addressing the challenge of total cost of ownership is by making information systems more self-managing. A particularly difficult piece of the ambitious vision of making database systems self-managing is the automation of database performance tuning. In this paper, we will explain the progress made thus far on this important problem. Specifically, we will propose the architecture and Algorithm for this problem.
***Key Words**: Database Tuning, DBA, SGA, SGA Parameters, Automated Tuning, TOC.*

1 Introduction

As we have seen, hardware costs fall rapidly while human costs remain relatively static. This leads to a condition there the human costs of manual tuning activities outpaces the costs of faster hardware (see figure 1). Most large databases are managed by DBAs who are responsible for the good performance of the database but manual physical design is both time consuming and very tedious, as the database administrator (DBA) needs to find the benefits of different individual design features that can possible interact with one another. Motivated not only by the difficulty of tuning but also from the need to reduce the total cost of ownership in their products, several commercial DBMS vendors offer automated physical design tools with several features.

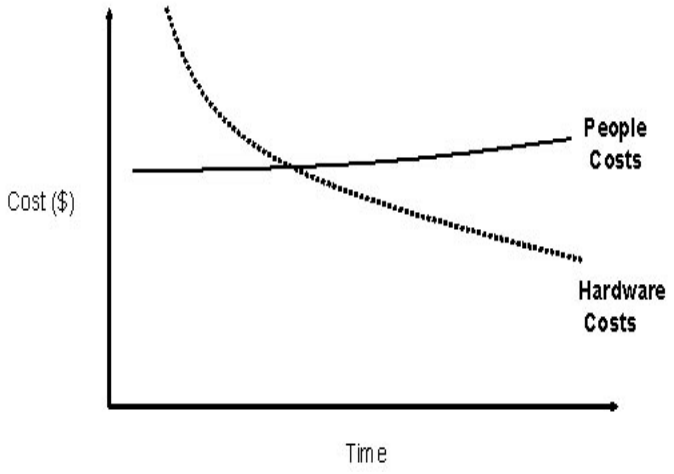

Fig.1. H/W cost vs. Human Cost [Ref. 2]

With the dramatic drop of hardware and software prices, the expenses due to human administration and tuning staff dominate the cost of ownership for a database system [4].

The physical design problem involves searching a potentially very large space of different candidate configurations. Searching the space of alternative configurations is impractical. Therefore, recent physical design tools are based on greedy heuristics that prune the search space. The repeated calls to the optimizer each time we want to evaluate a query under a different configuration impose a serious bottleneck in the execution of physical designers. Based on experimental results 90% of the tuning time is spent on waiting results from the optimizer instead of evaluating potentially promising configurations [5,6].

2. Performance Tuning

Most systems will respond to increased load with some degree of decreasing performance. A system's ability to accept higher load is called scalability, and modifying a system to handle a higher load is synonymous to performance tuning.

Systematic tuning follows these steps:

- Assess the problem and establish numeric values that categorize acceptable behavior.
- Measure the performance of the system before modification.
- Identify the part of the system that is critical for improving the performance. This is called the bottleneck.
- Modify that part of the system to remove the bottleneck.

A performance problem may be identified by slow or unresponsive systems. This usually occurs because high system loading, causing some part of the system to reach a limit in its ability to respond. This limit within the system is referred to as a bottleneck. A handful of techniques are used to improve performance.

Data drives today's businesses, and managing databases often involves complex planning, time management and system wide routine task implementation. Database automation helps enterprises better manage their database operations, reducing down-times as well as the overall time taken in database management. Automation anywhere works with any SQL database, like Oracle, MS SQL, Sybase, SQL DB2, etc. Unlike other automation solutions, it does not require significant training. Simple, easy-to-use yet powerful, it can automate any database task.[1]

3. Manual System Architecture

Database Administrator is responsible for enhancing the performance of database system. The detection of performance degradation is achieved by continuously monitoring system performance parameters. Several methods including the usage of materialized views and indexes, pruning table and column sets, usage of self healing techniques, usage of physical design tuning etc have been proposed that proactively monitor the system performance indicators, analyze the symptoms and auto tune the DBMS to deliver enhanced performance. The performance degradation is due to increased workload on the system. This increased load has to be minimized to enhance the response rate of the system. In order to achieve this objective, either the administrator decreases some amount of load by closing some files or he may increase the RAM. The administrator has to check continuously or we can say, at regular intervals the Buffer Cache Hit (BCH) ratio. Based on this hit ratio, the database administrator determines if more amount of RAM has to be allocated. This task of load reduction by increasing RAM requires manual intervention and thus may take even years to complete.[1]

However, Oracle manages RAM memory demands according to the demands of each task by using sophisticated algorithms to improve the speed of RAM intensive tasks. Oracle DBA can dynamically de-allocate RAM memory as well as re-allocate it. But since database administrator is a normal human being, he cannot calculate the actual amount of RAM memory required by an application.

Due to this limitation of DBA, the allocation of RAM manually for optimizing

performance of database system becomes a complicated as well as costly task.

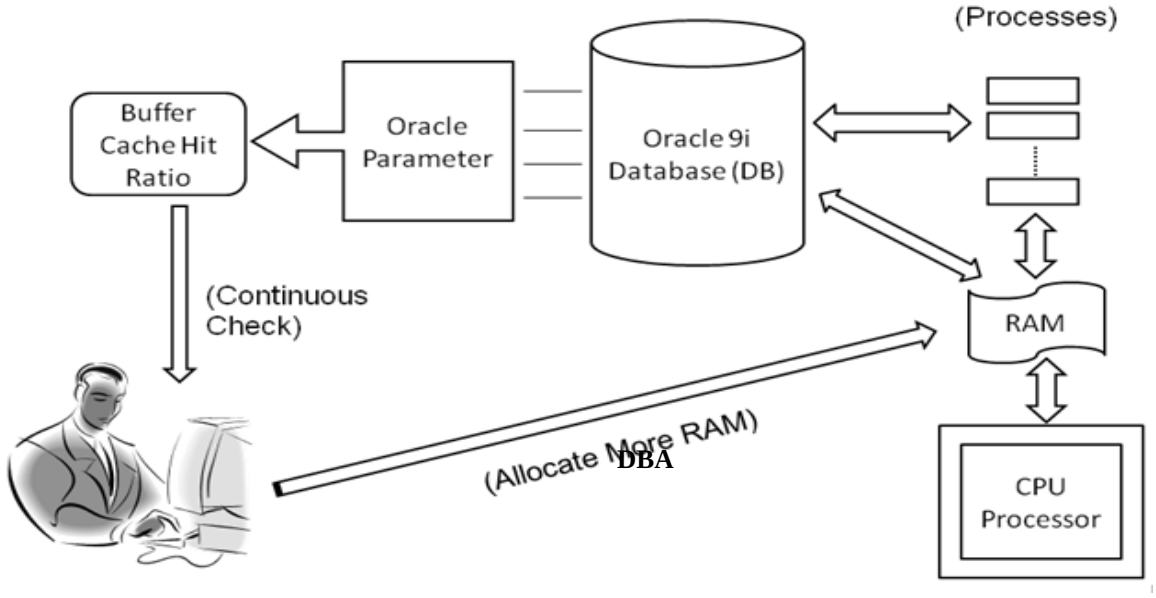

Fig.2. Manual Database Design

Sometimes, more amount of RAM is allocated than needed which wastes the extra portion of RAM. [3] Thus, there is a great need of dynamic memory allocation features to create a self tuning database. In Oracle Database 10g, a self tuning feature such as Automatic Memory Management (AMM) allows the database system to detect shortages and adjusts the main memory regions according to the changing demands on the Oracle environment. Therefore, researchers are now focusing on the development of self tuning techniques such as the COMFORT automatic tuning project [5] or the MAPE approach given by IBM [6] for a continuous adaptation.

Ranking of various tuning parameters based on statistical analysis is presented in [7]. The ranking of parameters is based on the amount of impact they produce on the system performance for a given workload. A formal knowledge framework for self tuning database system is presented in that define several knowledge components which include Policy knowledge, Workload knowledge, Problem diagnosis knowledge, Problem Resolution Knowledge, Effectors knowledge, and Dependency knowledge. The architecture presented in this paper involves extracting useful information from the system log and also from the DBMS using system related queries. This information gathered over a period of time is then used to run the SQL scripting for a desired output response time. The application framework would then estimate the extent of correction to be applied to the key system parameters that help scale up the system performance. The classical control is modified and a three stage control involving Monitor, Analyze and Tune [7] is employed to ensure system stability. The architecture presented in for self healing database forms the basis for the new architecture presented in this paper. This paper presents a new DBMS architecture based on modular approach, where in each functional module can be monitored by set of monitoring hooks. These monitoring hooks are responsible for saving the current status information or a snapshot of the server to the log. This architecture has high monitoring overhead, due to the fact that when large number of parameters to be monitored, almost every module's status information has to be stored on to the log and if done frequently may eat up a lot of CPU time. Moreover, this architecture focuses more on healing the system and does not consider tuning the DBMS for performance

improvement.

4. Automated System Architecture

Many business applications demand the use of complex database systems which should be administered and optimized for better performance. As suggested in [2], physical tuning should be avoided as it is expensive. As the physical design of database suffers from various limitations, a new script based automated database architecture is proposed in order to achieve high grade of performance. The architecture as shown in figure 3 is employed for identifying the symptoms and altering key system parameters. The DBMS system log file will be the primary source of information checks the current status of the system. The data miner tool compresses the data into smaller information base since the log file may contain huge amount of data. The architecture has three basic building blocks comprising of Data Miner, Script and Tuner. After the extraction of meaningful information, the extent of correction required is estimated by the proposed script and algorithms.

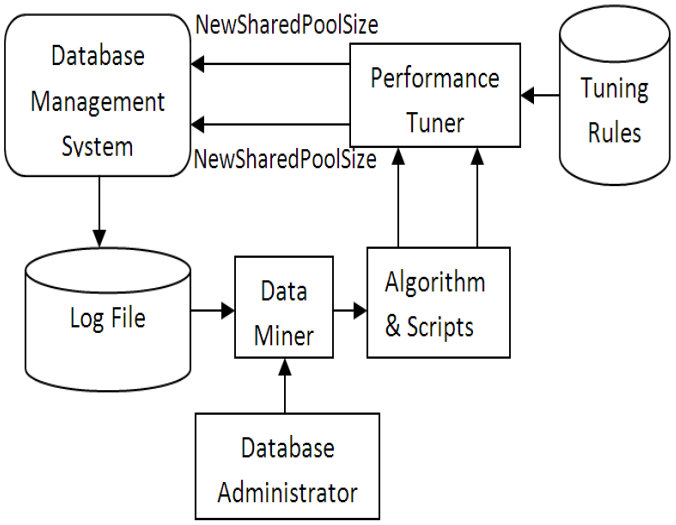

Fig.3. Script based Tuning architecture

These algorithms and scripts would tune the database using various tuning rules as well as system parameters. However, several parameters can be altered simultaneously for better performance gain. The algorithm estimates the required buffer size based on the current DBMS input parameters and the tuner applies the necessary correction to the buffer size based on the tuning rules. Most importantly the internal corrective measure such as altering the buffer size of the DBMS used in query processing is explored in this architecture.

In this research proposal, we provided a self tuned database system architecture as shown in fig 4 in order to enhance system performance. Since DBA is responsible for administration and optimization of various tasks, he can either increase RAM or can decrease the amount of load on CPU for the purpose of performance optimization. But this would be time consuming technique as DBA is a normal human being who cannot perform complex calculations within seconds like a computer system. [5]

DBA may not know exactly how much RAM is to be allocated for enhancing system performance. So, we propose an approach to automate this optimization task of DBA as shown in fig i.e. the task

which DBA has to do for performance enhancement would now be done by the computer system within small timelines.

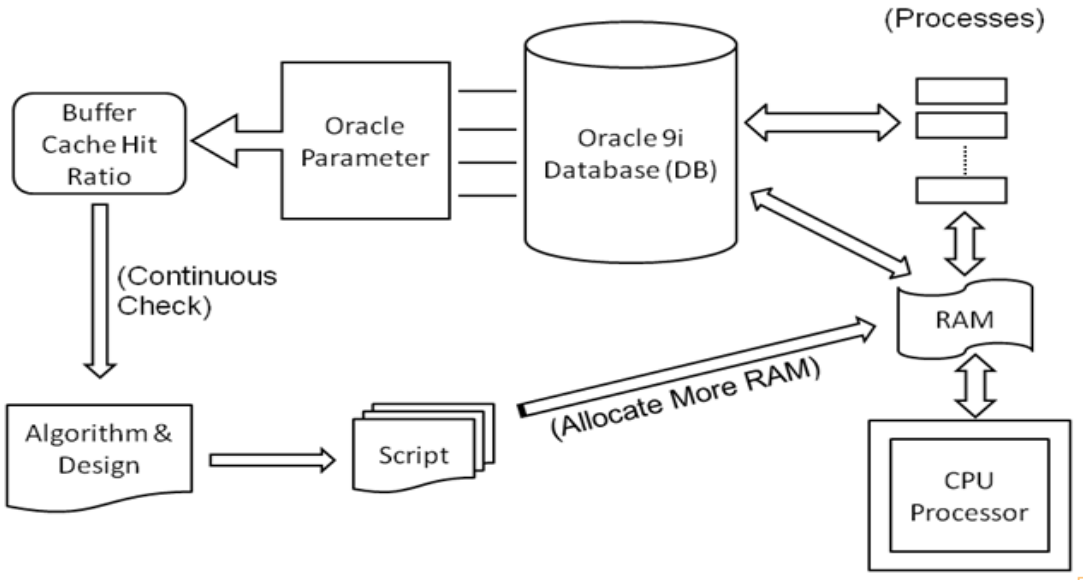

Fig.4. Automated Database Design

5. Algorithm & Flow Chart

The algorithm defines three variables: ΔRT abbreviates for change in response time, BUFFER_SIZE denotes the current size of buffer, CACHE_SIZE corresponds to the size of cache memory

```
ALGORITHM

1. dbTuner (ESTIMATED_CACHE_SIZE)
2. Begin
3.      Run application, algorithm and process
4.              Calculate the change in response time (ΔRT)
5.              If (ΔRT>0)
                {
                        Run Script
                        {
                                BUFFER_SIZE = BUFFER_SIZE + 1
        Allocate more RAM and update CACHE_SIZE
        }
      Else IF (ΔRT<0)
      {
       Run Script
       {
                BUFFER_SIZE = BUFFER_SIZE - 1
        Reduce RAM and update CACHE_SIZE
        }
      }
6.   Go To Step 4
6.   Stop application, algorithm and process
7.  End
```

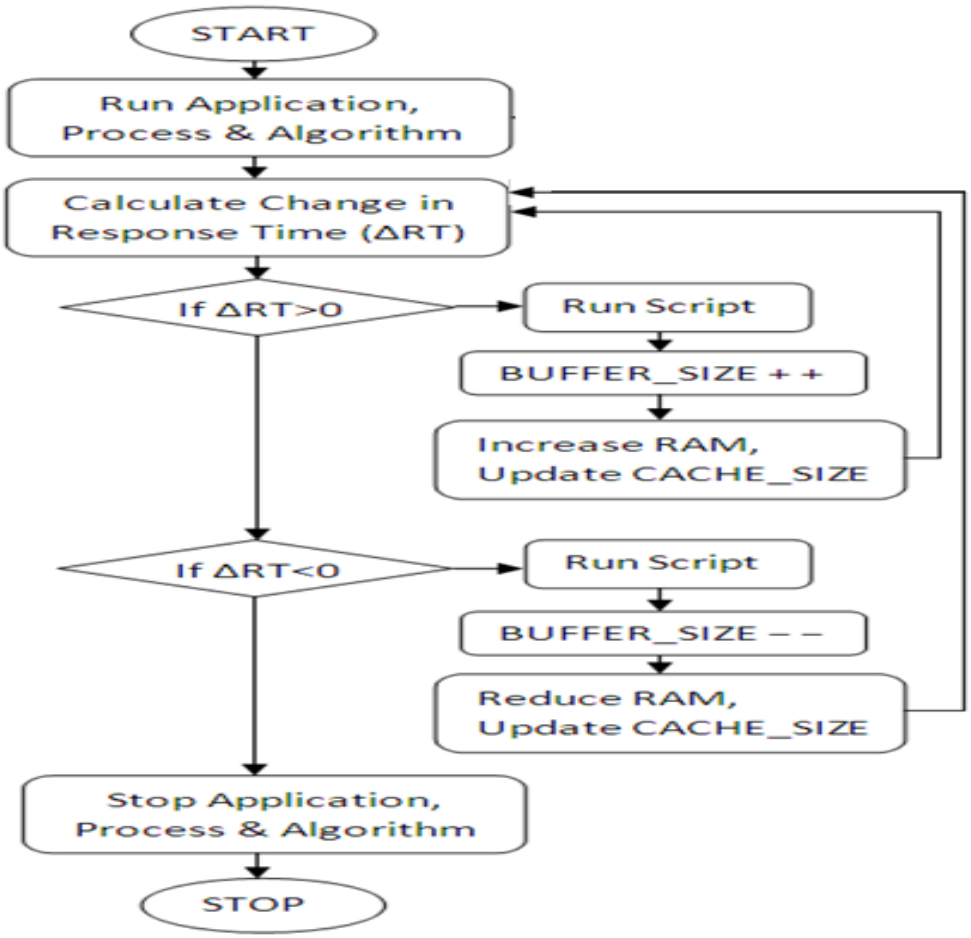

Fig.5. Flowchart for Automated Tuning

6. Experimental Results

Table 1 shows the sample training data. A training data set of size 100 was used to test the proposed system. As can be seen from the table, the buffer size is adjusted for increased table size, Number of user processes and Buffer Miss Ratio so that query execution time is reduced and the memory is used efficiently.

Table 1. Sample Training Data Set

Tab. Size (in no. of records)	Buff.Miss Ratio	Shared Pool size (in MB)	Buff. Size (in MB)
1000	0.9624	32	4
1000	0.9152	32	4
1000	0.9791	32	8
1000	0.9613	32	8
2000	0.9371	32	8
2000	0.9453	40	8
3000	0.8931	40	16
3000	0.8253	40	16

7. Conclusion

Tuning the database can become quite complex, but Oracle9i offers the administrator an unparalleled ability to control the PGA and SGA. Until Oracle9i evolves into a completely self-tuning architecture, the DBA will be responsible for adjusting the dynamic configuration of the system RAM. Automated SGA adjustment scripts can be used to allow theDBA to grow and shrink the SGA regions. These scripts are placed in dbms_job for scheduled processing. Oracle provides enhanced views in v$process, v$pgastat to allow you to monitor the behavior of the RAM sort area The v$ views in Oracle9i also provides insights about the RAM usage for individual SQL statements within the library cache.

References

[1] I. Alagiannis, *Towards Adaptive, Flexible, and Self-tuned Database Systems*, (DIAS, I&C, EPFL) in EDIC-ru/05.05.2009

[2] D. Burleson, *Oracle Tuning,* The Definitive Reference Second Edition.

[3] *Foundations of Automated Database Tuning*, VLDB '06, September 12–15, 2006, Seoul, Korea.Copyright 2006 VLDB Endowment, ACM

[4] *Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System in Cairo*, Egypt, 2000

[5] *AutoAdmin:Self-Tuning Database Systems Technology*, Copyright 2006 IEEE

[6] *Self-Tuning Database Systems: A Decade of Progress*, Copyright

[7] *SQL Memory Management in Oracle9i*, Hong Kong, China, 2002

Hitesh KUMAR SHARMA,
The author is An Assistant Professor in ITM University. He has published 8 research papers in National Journals and 2 research paper in International Journal. Currently He is pursuing his Ph.D. in the area of database tuning**.**

Aditya SHASTRI,
Ph.D. MIT, Published about 200 research papers in international journals on Graph Theory with applications in Communication, Computer Graphics and Parallel Processing,Vice Chancellor, Director, Banasthali University, Banasthali, INDIA

Ranjit BISWAS,
Associate Director, MIRU, Faridabad, Published about 100 research papers in International journals/bulletins of USA/Europe, out of which more than 40 papers are independently published papers and the rest are published jointly with other authors (with Ph.D. scholars). Haryana, INDIA

Integrating XML Technology with Object-Relational Databases into Decision Support Systems

[1]Iuliana BOTHA, [2]Adela BÂRA, [3]Simona-Vasilica OPREA, [4]Ticiano COSTA JORDÃO
[1,2]Academy of Economic Studies, Bucharest
[3]Transelectrica Company, Bucharest
[4]University of Pardubice, Czech Republic
iuliana.botha@ie.ase.ro, bara.adela@ie.ase.ro, simona.oprea@transelectrica.ro, ticiano.costajordao@upce.cz

This paper presents some informatics technologies, like XML and object-relational databases, and the main motivations for implementing them in Decision Support Systems (DSS). Also, it is proposed and analyzed a conceptual model of a DSS prototype, which can be applied in the uncertain and unpredictable environments, like the production and the prediction of the wind energy.
Keywords: *Object-Relational Databases, XML, Decision Support Systems, Wind Energy, Power Plants*

1 Introduction

Early this century was marked by a series of major changes in the field of information and communication technology (ICT), but also in the field of economics, leading to the creation and development of a new type of economy, generally named most often, the new economy.

The role of informatics in the new economy, as producer and consumer of information, is crucial. All economic activity is strongly influenced by the evolution of the informatics context and vice versa.

Under these facts, it is appropriate the research of decision-making solutions development, built with advanced technologies, such as relational, object oriented and XML.

Decision support systems (DSS) form a distinct class of computer systems. It integrates specific IT tools to assist decision and general purpose tools to form a constituent part of the organization's overall information system. Any DSS is designed to provide IT support required to minimize the effects of restrictions (cognitive, communication, time, etc.) with which is facing the human decision maker during his activities. Decision support systems have become a real presence in the IT world. Web technologies, advanced databases and methods for accessing them, flexible user interfaces, are all pressing needs of IT technology. Therefore, today's decision support systems, in order to be efficient, integrate many of these technologies. The DSS proposed in this paper will use spatial, multimedia and XML data stored in object-relational databases.

2. Considerations about object-relational databases

The attempts to model the complex elements of the economic activities have caused the reorientation to the object-oriented technology. Thus, in the late 80s has appeared the third generation of databases, namely the object-oriented ones. As stated in [2], the development of object-oriented data model was due to limitations of the relational model to handle efficiently massive amounts of data, with high complexity, found in new types of applications (multimedia, Internet, Geographic Information Systems, etc.).

Although object-oriented databases appear to meet the needs of applications required by the new economy, the market they use remains relatively low, often the invoked reason being the difficulty to query data

and the high consumption of computing resources [10].

Currently, the solution that combines the advantages of relational and object-oriented data models, comes with a hybrid data model, the object-relational one, which involves object-oriented facilities (especially the fundamental characteristics of objects: encapsulation, inheritance, polymorphism) as extensions of relational model.

The object-relational data model was designed as result of the research in the 90s, by extending the concepts of relational databases with object properties. In the center of the new model architecture, the researchers decided to keep the declarative query language based on predicate calculation.

Thereby, the object-relational databases (ORDB) are the result of applying features of object-oriented technology in data storage and retrieval. Unlike object-oriented databases, the object-relational ones provide the ability to represent complex data structures through objects, while retaining the advantages of relational databases.

Broadly, we can define an object-relational database as a set of tables of objects, linked logically, organized in the external memory and accessible to multiple users in a timely manner.

The main features of object-relational databases are given by the flexibility and the structure of the used data model and refer mainly to the following:

- Treating consistently the entities, by modeling both the properties of objects and their behavior;
- Providing communication between data and among programs;
- Simplifying the data structure that provides ease of use and high portability of the resulting systems;
- Modeling and flexible storing the real world objects, which helps to address the complex areas and to use different types of data;
- Integrating the structural and behavioral description of the entities, which gives a dynamic aspect to the object;
- Inheriting the properties of object types, which provide the opportunity to realize complex operations and processing over the data.

Concepts used in object-relational databases are borrowed from the object-oriented terminology, but also from the relational one, and they were presented at defining the structure of the object-relational data model. Figure 1 illustrates the combination of these concepts in order to develop the object-relational databases.

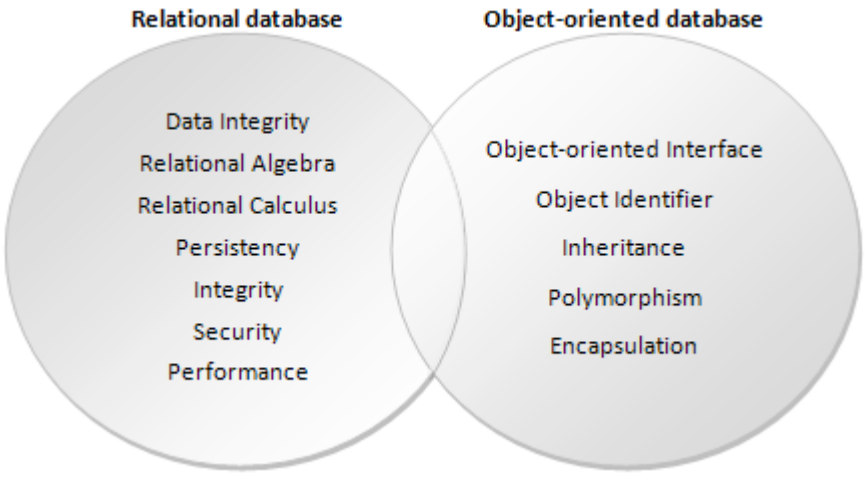

Fig. 1. The main concepts of the object-relational databases
(Source: Authors)

Defining the type of objects represents the mechanism for specifying the database schema. Type definitions include inheritance relationships (which generate supertypes and subtypes) and structural relationships between them.

A complete schema of the object-relational database consists of relational tables and object tables. The object tables can present inheritance relationships between types of objects that were the basis for creating these tables.

Can be identified two types of changes in an object-relational database schema. Thus, we can discuss about changes on how to define an object type, which can lead to changes in the specification of attributes and methods defined (e.g. changing the domain or the name of an attribute; adding or deleting an attribute or a method). Updates in the schema can also appear in the case of some changes in the hierarchy of types or in the supertype/subtype relationships. However, this version of modifying the schema is limited, depending on manufacturer.

According to [11], a fully object-relational database must necessarily meet the following four characteristics:

- Allowing the creation of extensions from basic types. Refers to the ability to create extensions of data types by defining abstract, simple or complex, data types;
- Enabling management of complex objects. There is a major difference between the ability to manipulate complex data using an object-oriented database and using relational database. The set of basic types of relational databases is very poor compared to the existing set of object-oriented databases. In this respect, [11] states that a database, to be considered object-relational should support a rich collection of complex data types, including mandatory: complex types, SET constructors, RECORD constructors, REF constructors;
- Accepting the inheritance property. In an object-relational database, a table is seen as a container used to hold instances of a type that can use the inheritance feature. This can refer both to the data and methods inheritance. Data inheritance applies only to the data types, but the property also can be propagated upon the tables built over these data types. Instead, methods inheritance apply to the user-defined functions and methods included in the types of objects. Also, according to [11], a fully object-relational database must support multiple inheritance, required in many applications;
- Using a system of rules. In a fully object-relational database should not miss a system of rules and triggers, necessary especially to ensure consistency of the database.

In conclusion, based on the object-relational data model, we can define an object-relational database as a collection of tables, relational or object, persistent, stored in the external memory through a system of rules, consistently organized, ordered in hierarchies and providing shared access to competing users. As a hybrid, the object-relational databases combine flexibility, scalability and security in using existing relational systems with object-oriented properties such as: abstraction, encapsulation, inheritance and polymorphism.

3. Integration of XML technology into object-relational databases

eXtensible Markup Language (XML) was designed as a standard for information exchange over the Internet. In [13], W3C offers an abbreviated definition for XML: „XML is a markup language for structuring arbitrary data. XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's

storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure."

Databases and XML offer complementary functionality for storing data. While databases store data, ensuring their rapid retrieval, XML enables easy exchange of information, which allows interoperability between applications due to data encapsulation with metadata.

XML allows designers to create their own tags, allowing the definition, transmission, validation, and interpretation of data between applications and between organizations. In Romania there are several systems based on XML standard (e.g. Labour Inspection, Bucharest Stock Exchange) [9].

Can be identified two approaches in order to store XML data: we can use native XML databases, or we can map XML data and queries into a relational or object-relational database [6]. The advantage of using object-relational databases is that we can get the benefits of both relational and object-oriented technologies, while the disadvantage translates into lower performance due to XML data mapping to the relational data, which can produce a database schema with many relations.

When used in object-relational databases, XML data must be mapped into relations. In order to transfer the data between XML documents and object-relational structures are used specific mapping methods. The study [8] makes a basic classification of these mapping methods, as follows:

- *Generic methods* – that are not using any schema of stored XML documents;
- *Schema-driven methods* - that are based on existing schema of stored XML documents;
- *User-defined methods* - that are based on user-defined mapping.

The mapping process is represented in Figure 2 and detailed in [6]:

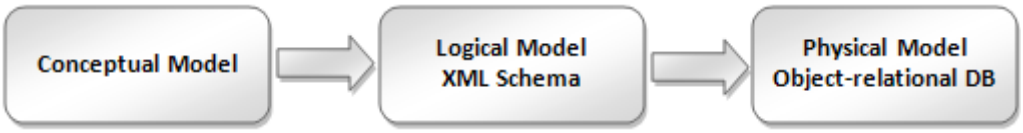

Fig. 2. The mapping process
Source: [6]

An algorithm for mapping XML schema to object-relational database schema is proposed in [7]. Also, in [5] are presented mapping algorithms and correspondences between object-relational databases and XML schema for the description and validation of the data. The correspondence is observed in reverse and includes in addition to abstract data types (those defined by the user, which corresponds to XML schema elements), persistent objects, specific to databases, in general.

Starting from [5], in table 1 we have completed the set of mapping rules, as follows:

Table 1. Mapping rules: ORDB – XML Schema
Source: adapted from [5] and completed

Object-relational database element	XML Schema element
Table	Complex type whose sequence contains elements corresponding to the table columns, including the constraints

Simple attribute, without constraints	Element of the complex type sequence, which is represented through a simple type
Attribute with primary key constraint	Attribute of the complex type having the property *use="required"*
Attribute with not null constraint	Element of the complex type sequence having the property *nullable="false"*
Attribute with uniqueness constraint	Element of the complex type sequence, with the attribute *xs:unique*
Attribute with referential constraint	Element of the complex type sequence, with the attribute *xs:keyRef*
User defined object type	Complex type having the elements corresponding to the attributes of the object type
REF type attribute	Reference specified directly as a property of the XML schema element *ref="pointer"*
ROW type attribute	Complex type whose elements represent attributes of type ROW
Collection type attribute	Complex type which contains a sequence of the element associated with the collection type, that has the property *maxOccurs* equal with the collection dimension
Inheritance	Complex type having complex content and the type *extension*

The DSS that we propose will use XML documents containing data which may influence wind park performances, like meteorological data. These XML data need to be stored in object-relational databases, in order to realize further analysis, necessary for decision making.

4. Conceptual model for a decision support system with object-relational databases implementation

Organization management involves different types of activities and therefore requires different types of information. For better manage, transform, process and analyze this information, it is needed a decision support system (DSS), seen as "an information system that draws on transaction processing systems and interacts with the other parts of the overall information system to support the decision-making activities of managers and other knowledge workers in organizations" [4].

For an efficient management of resources it is necessary to be able to achieve a prediction of energy produced with minimal errors.

Currently, as stated in [10], both nationally and internationally, there are various types of frameworks, architectures, solutions and systems to provide economic assistance for decision making processes and production environments with a relatively high degree of certainty.

However, as shown in [1], the existence of such a system to assist decision making at national level is necessary because currently there are no solutions to offer efficient predictions and with minimal errors for the environments with low predictability, such as wind energy, which depends exclusively by natural factors.

As shown in [12], in Europe there are some countries that have been developed decision support systems in the energy system for efficient management of wind resources (e.g. Germany, Spain), but the costs of building these systems is very high and the particularities of national energy potential makes difficult the application of these methods in Romania.

These issues concerning the impossibility of prediction accuracy, the precarious data integration from various local equipment and systems and the inefficient analysis of energy resources, lead to the need to develop solutions for a better prediction of energy produced, but also an integrated system, with a component for assisting decision making in this area.

The data sources provided for the decision-making process must be as varied and require integration. Some of the data collected within the operational activity will be stored in the object-relational database as: spatial data, multimedia data, Large Objects, user-defined object types. The spatial data are used in order to representing them on interactive maps monitoring the production of energy, multimedia data and LOB for using and manipulating large objects.

Also, since data integration from various sources is mandatory, we will use XML as a standard for integration and data warehouses for centralize data from heterogeneous data sources.

The decision support system will use data that can influence wind park performances, like meteorological data or data resulted from national energy production regular monitoring. The data will be received like XML documents and after validating them with XML Schemas, they will be stored in object-relational databases, in order to realize further analysis, necessary for decision making.

In the following paragraphs, we propose an architectural model for such a decision support system based on the typical architecture of a DSS, but with specific components of information requirements identified in the National Energy System (NES).

In [3] are identified four components of a decision support system: 1) interface; 2) database system; 3) system of analytical, mathematical and statistical models; 4) component to ensure communication. This architecture is implemented in most Business Intelligence solutions made by major manufacturers.

Starting from these standard DSS architecture and also based on the information requirements identified by the National Energy System, we have proposed the conceptual model of the informatics system. Thus, as stated in [10], the conceptual model contains the four levels presented above (Figure 3).

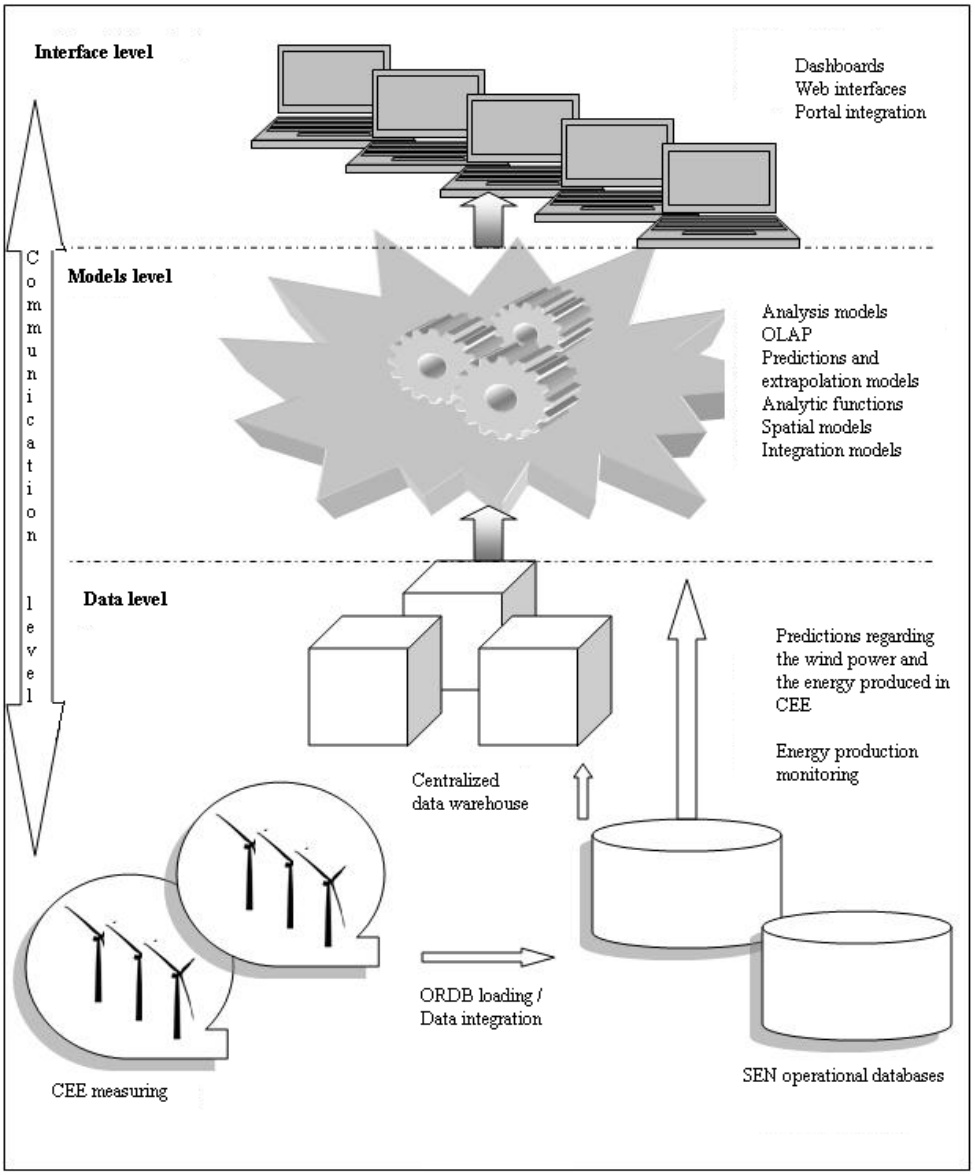

Fig. 3. The architecture of the proposed DSS
Source: [10]

The proposed DSS architecture can be seen in terms of achievement levels, from the bottom to the top of the pyramid, on three levels: the bottom-tier, middle tier and top-tier, being interconnected at the level of telecommunications.

1) The data level will consist of operational data sources, as well as spatial data and the data warehouse to be built to facilitate multi-dimensional analysis.

Some of the data collected on operational activity will be stored as data in a spatial DBMS (Database Management System Data), in order to represent them on maps for interactive monitoring of energy and resources.

The data warehouse will be a centralized repository, organizational, composed of data marts for each type of activity which is analyzed: operational (energy production) and financial. To load data warehouse from data sources is necessary to undertake a process of Extract, Transform and Load (ETL). This process will automatically run at regular intervals depending on the technical requirements of implementation.

2) The models level will include models for the current financial analysis and

forecasting of these activities, models for spatial data representation, and simulation models, extrapolation, prediction and analysis of energy production from wind sources. To achieve models will use new technologies for Business Intelligence (BI), such as OLAP, data mining, predictive analytic functions and algorithms.

3) The interface level will contain the data presentation elements and specific analysis and reporting for decision support systems. The system will be accessible via a Web interface so users can access reports without having to install client applications. It will choose to integrate all elements into a portal interface that allows a single, uniform type of authentication: Single Sign - On. The portal will integrate elements of the subsystems and previous levels, such as OLAP analysis, prediction and monitoring modules, reporting.

4) The level of telecommunications will provide support for access the decision support system both within and outside NES, using mobile devices. For this level will use the existing network components in the NES, where the access is authorized according to the role and access rights of each user.

Following analysis of the data sources provided by the wind power plant units, we found data heterogeneity, mainly due to the diversity of wind turbines and measuring instruments, and computer applications to process data. For this reason it is essential to use at data level techniques of data migration and data integration. In this manner, data can be loaded in a consistent manner in a centralized operational database which can be later used for prediction, simulation and analysis.

Conclusions and future work

The integration of the special data (like multimedia, LOB, XML, spatial data) into object-relational databases is a necessary characteristic for today's enterprises that maintain informatics systems in the unpredictable environments.

The paper provides a synthetic look at object-relational database features and possibilities for integration with XML technology. Also we have presented an implementation perspective of a DSS, which requires decision making in the wind energy prediction and production.

Our research will continue with the development and implementation of the conceptual model proposed in this paper. For this purpose, will be used specific technologies for each level of the model architecture. Thus, the data level will use solutions for organizing and integrating data, the models level will use solutions of multidimensional analysis, forecasting models, simulation and extrapolation, and for the interface level will be used solutions for data analysis and dynamic presentation of data.

Acknowledgment

This paper presents some results of the research project PN II, TE Program, Code 332: "Informatics Solutions for decision making support in the uncertain and unpredictable environments in order to integrate them within a Grid network", financed within the framework of People research program.

References

[1] Adela Bâra, Ion Lungu, Manole Velicanu, Simona Vasilica Oprea, "Intelligent Systems for Predicting and Analyzing Data in Power Grid Companies", *TheProceedings of the IEEE International Conf. on Information Society* (i-Society 2010) London.

[2] Iuliana Botha, AndaVelicanu, Adela Bâra, "Integrating Spatial Data with Object Relational-Databases", *Journal of Database Systems*, nr.1/2011, pp. 33-42, ISSN: 2069 – 3230

[3] Daniel Power, "Decision Support Systems: Concepts and Resources", Cedar Falls, 2000: DSSresources.com, http://dssresources.com/dssbook

[4] Ralph H. Sprague, Eric Carlson, "Building Effective Decision Support Systems", Prentice-Hall, 1982, ISBN: 978-013-086-215-0

[5] Laila Alami Kasri, Noureddine Chenfour, "Model of Storage XML Database based on the Relational-Object Model", *International Journal of Engineering Science and Technology*, Vol. 2(11), 2010, ISSN 0975–5462

[6] Eric Pardede, J.Wenny Rahayu, David Taniar, "On using Collection for Aggregation and Association Relationships in XML Object-Relational Storage", *ACM Symposium on Applied Computing*, Cyprus, 2004, ISBN 1-58113-812-1

[7] Irena Mlynkova, Jaroslav Pokorny, "From XML Schema to Object-Relational Database – an XML Schema-driven mapping Algorithm", *Proceedings of the 3rd IADIS International Conference WWW/Internet*, Madrid, Spain, 2004, pp 115 - 122, ISBN 972-99353-0-0.

[8] Irena Mlynkova, Jaroslav Pokorny, "XML in the World of (Object-) Relational Database Systems", *Information Systems Development: Advances in Theory, Practice, and Education*, Vilnius, Lithuania, 2004, pp. 63 - 76, ISBN 978-0-387-25026-7.

[9] Vlad Diaconița, Iuliana Botha, "The Level of System Integration in Romania", *Informatica Economica Journal*, Vol. XI, no. 3/2007, pp. 96-99, ISSN 1453-1305

[10] Iuliana Botha, Adela Bâra, Simona Vasilica Oprea, "An implementation perspective on object-relational databases in decision support systems", *Proceedings of the Tenth International Conference on Informatics in Economy*, 2011, ISSN: 2247-1480

[11] Michael Stonebraker, Dorothy Moore, "Object-Relational DBMS - The Next Great Wave", Morgan-Kaufmann, 1996, ISBN: 155-860-397-2

[12] Adela Bâra, Simona Oprea, Anda Velicanu, Vlad Diaconița, "Decision support systems - solutions for predicting, monitoring and integrating wind energy resources into the national grid companies", *Development, Energy, Environment, Economics*, pp. 55-60, 2010, ISBN: 978-960-474-253-0

[13] http://www.w3.org/TR/xml/

Iuliana BOTHA is an Assistant Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2006 and the Databases for Business Support master program organized by the Academy of Economic Studies of Bucharest in 2008. Currently, she is a PhD student in the field of Economic Informatics at the Academy of Economic Studies. She is co-author of 5 books, 12 published articles (2 articles ISI indexed and the other 10 included in international databases), 16 scientific papers published in conferences proceedings (among which 6 paper ISI indexed). She participated as team member in 4 research projects that have been financed from national research programs. From 2007 she is the scientific secretary of the master program *Databases for Business Support* and she is also a member of INFOREC professional association. Her scientific fields of interest include: Databases, Database Management Systems, Design of Economic Information Systems, Grid Computing, e-Learning Technologies.

Adela BÂRA is a Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Economic Cybernetics in 2002, holds a PhD diploma in Economics from 2007. She is the author of 7 books in the domain of economic informatics, over 40 published scientific papers and articles (among which over 20 articles are indexed in international databases, ISI proceedings, SCOPUS and 10 of them are ISI indexed). She participated as team member in 3 research projects and has gained as project manager one research contract, financed from national research programs. She is a member of INFOREC professional association. From May 2009, she is the director of the Oracle Excellence Centre in the university, responsible for the implementation of the Oracle Academy Initiative program. Domains of competence: Database systems, Data warehouses, OLAP and Business Intelligence, Executive Information Systems, Decision Support Systems, Data Mining.

Simona Vasilica OPREA is a Senior Engineer and Project Manager at Transelectrica National Power Grid Company. She has graduated the Polytechnic University in 2001, holds a Master Diploma in Infrastructure Management Program, Yokohama National University, Japan in 2007 and a PhD diploma from 2009. She is the author of over 20 articles, from which 3 ISI Web of Science indexed and 2 included in SCOPUS international database. Domains of competence: wind farm, investment opportunity analysis, studies of prognosis, stationary and dynamic regimes, short circuit calculations.

Ticiano COSTA JORDÃO is a Full Lecturer at the Faculty of Economics and Administration of the University of Pardubice (Czech Republic), director and co-founder of the Sustainability Crusade Observatory (CRUSUS), vice-president of the Czech-Brazilian Chamber of Commerce, and a member of the editorial board of the International Journal *Sustainable Development and Corporate Social Responsibility Review* (*SD&CSR*).

In order to know more about his professional work, please consult: www.crusus.org.

Integration of Information Technologies in Enterprise Application Development

Iulia SURUGIU
TotalSoft, Bucharest, Romania
isurugiu@totalsoft.ro, iulia.surugiu@yahoo.com

Healthcare enterprises are disconnected. In the era of integrated information systems and Internet explosion, the necessity of information systems integration reside from business process evolution, on the one hand, and from information technology tendencies, on the other hand. In order to become more efficient and adaptive to change, healthcare organizations are tremendously preoccupied of business process automation, flexibility and complexity. The need of information systems integration arise from these goals, explaining, at the same time, the special interest in EAI. Extensible software integration architectures and business orientation of process modeling and information systems functionalities, the same as open-connectivity, accessibility and virtualization lead to most suitable integration solutions: SOA and BPM architectural styles in a cloud computing environment.
Keywords: *EAI (Enterprise Application Integration), BPM (Business Process Management), SOA (Service Oriented Architecture), ESB (Enterprise Service Bus), WOA (Web Oriented Architecture)*

1 Introduction

The importance and the actual state of knowledge regarding the integration of informatic technologies in enterprise application development, rise from the current research on information processes, concepts and technological approaches. The essential mechanisms regarding enterprise software platforms and integration requirements of heterogeneous platforms at different levels (data, functionalities, services, processes), depend on business specific and existing software products. Integration processes should be oriented on capturing the development stages and aspects of enterprise applications on the one hand, and information technology integration processes, on the other.
Basic EAI principles and concepts should be highlighting issues in defining complete and coherent solutions for enterprise application integration.

2. Analysis of enterprise software platforms in relation with business requirements

EAI technologies are emerging, not yet reached a consensus on the ideal approach that businesses should adopt the correct formula and complete integration. Moreover, the processes of definition of original and revolutionary EAI solutions are based on the existing integration solutions, that are ussually optimized and extended in order to meet new requirements and approaches in the field of integration technologies.
The best known commercial EAI solutions are: IBM WebSphere Message Broker, Jonah Technologies, Microsoft BizTalk Server, Oracle BPEL Process Manager, Pervasive Software, SAP Exchange Infrastructure, Sterling Commerce, TIBCO Software, webMethods, Sun Microsystems. EAI open source solutions, often preferred by software integrators, are: OpenESB, Virtuoso Universal Server, Jitterbit Integration Server, JBossESB, Mule.
Table below presents a comparison between different EAI platforms, given some basic criteria: price, implementation, software solution maturity and support.

EAI solution/Criteria	Mule (MuleSource)	BizTalk (Microsoft)	WebSphere (IBM)	OpenESB (SUN)
Price	-	44,228 $ EE[1] 10,138 $ SE[2]	85,000 $ EE 25,000 $ SE	-
Implementation	ESB Messaging[3]	ESB	Messaging Middleware	ESB Messaging
Releases	often	rare	rare	rarely
Training & Support	yes	yes	yes	LogiCoy

Fig. 1. EAI solutions comparison

Concluding on the table, open source EAI solutions provide the same functionality as commercial solutions, also offered support developers is supported by communities or ad-hoc solutions for open source to commercial, which is purchased at cost. In terms of planning versions, open-source solutions tend to be updated more often. My choise in term of integration platform for this research is Mule, an open source solution that offers all the benefits of a mature development integration platform and a proper work environment.

EAI implementations and modularisation of software applications, regardless of business processes addressed, are based on applying some architectural principles in SOA approaches: applications are transformed in independent service bus, ESB architectures are used for application integration, XML is used for data sharing, operational architecture and logical database structure are redesigned, according modularization rules, services can be consumed independently, which allows more flexibility in deployment and software configuration capabilities, compared to solutions for stand-alone applications. Different versions of the integrated system are simplified by EAI modularization schemas, as an effect to a weak binary coupling, while versioning data and interfaces will be compliant to the overrall integrated solution. I will briefly identify some general characteristics of services and processes integration, focusing on the advantages and disadvantages of integrating information technologies in enterprise application development.

Advantages:

- natural and proper software solutions;
- development projects must follow the standards, procedures and working methodologies at all levels;
- independence between services and processes, and also business functionalities;
- project development and production rollout are efficient and easy to handle;
- systems are scalable and availability is maximum, restricted to user types, business processes and business reasons;
- project development life cycle is much reduced, while adapting to new requirements of business process is faster and easier.

Disadvantages:

- hard work on business analysis and technical analysis in the development of projects;
- consistent initial efforts to modify and adapt existing solutions and to implement the new system;
- additional and detailed management activities;
- not all software configuration options are possible for rollouts;

- some business processes that require interoperation may induce a response latency of communication between integrated subsystems.

3. Study of compatibility between different software platforms for enterprise application development

The latest technology approaches in enterprise application integration are SOA, WOA and cloud computing architectures.

Two current concepts and frequently used as an approach to integrate computer technology, cloud computing and technologies based on XML Web Services. Cloud architectures dominate distributed computing solutions that require access to hardware, data and functionalities. Architectures based on XML Web Services technology is the main trend in heterogeneous information integration technologies, the integration of functional capabilities and operational business process flows, the definition of complex models and consistent level of service integration and process. XML Web Services architecture is a starting point for integration of information technology-based processes, as programmable atomic entities underlying the business logic for definition of workflows and business processes.

Established enterprise application development platforms, Java and. NET allow the natural operation of SOA architecture and Web services consumption by implementing specific invokers. Both in Java and .NET platform, XML Web services use, in the context of client applications, is done through remote procedure calls, implemented in Java by invoking RPC on EJBs servers or Servlets and in .NET through listener implementation of type SOAP listener or .NET Remoting. Regardless of the technological approach and operating system being heterogenious, service-oriented architectures offer support for interoperability and integration, encouraging the development of ESB components in various programming languages designed for bussiness application connectors.

Integration architecture based on web services has led to aggregation of strong integrated solutions across business processes and enable design of complex models of business processes, the starting point for new technologies such as grid computing or WOA. In the context of orientation towards web technologies, the topic of cloud computing is the most powerfull approach in software integration, presenting series of capabilities and facilities of such software and hardware architecture models. I will define a comparative study between SOA and WOA architectures, highlighting specific factors of interest.

Table 1. Comparison between SOA and WOA

Criteria/ Architecture	**SOA**	**WOA**
Approach	Focus on simple functionality, but properly implemented.	Focus on simple and natural information technology and architectural concepts.
Reusability	Encourages use of Web services to standardize interoperability.	Opt for simple web technologies at the expense of complex protocols.
Software architecture	SOA services functionality aligns the business processes by mapping and modeling operations specific business logic.	The emphasis is on gradual implementation of functionality for business process modeling as their need arises.
Information technologies	Leverages declarative interfaces and insists on standardization of implementation tools	It promotes the implementation to use standard development tools, but encourages the use of information technology as the best suited typology

		applications tools

The main conclusion is that integration technologies such SOA is one of the most natural EAI solution to implement, while providing the possibility of defining reliable, stable, decoupled, complete architectures and, not least coherent and interoperable, beyond any limitations related software platforms or complex data structures. Exposing SOA integration solution through web, in a cloud computing environment, for instance, lead to extreme accesibility and virtualization power of business processes modeling business functionalities.

4. Development of a methodology for design, implementation and integration of information systems based on heterogeneous software platforms

Following standard EAI methodologies analysis and undertaken research, with the starting point of such process oriented methodology Waterfall TBI, I decided to use an iterative, process based methodology for developing the solution for applications integration. Each stage of software solution development will be functional validated and quality and experimental tested so that the work for the new phase to have the previous results as a starting point. By the very nature of the research project and the subject matter, iterative approach has proven to be an appropriate choice.

I consider that the most appropriate way to develop an integration project would be based on an Agile type EAI methodology, enhanced with RUP (Rational Unified Process) influences, MSF (Microsoft Solution Framework) and XP (eXtreme Programming) techniques. As iterative and incremental process model with object-oriented architecture, I propose RUP influences as a starting point, being suitable to use for a product with a high complexity and dynamism, with complex activities, continuously changing and developing.

Generally, choosing a methodology for developing a software system will be made according to each project specificity and to the nature of the business model. Methodologies and procedures, working rules will be not simply applied step by step, but they will intervene during each phase as required by the current project, as guidelines. Standard methodologies provide only starting points and directions to a complete and perfect methodological approach.

As in this case I combined three software methodologies and I used established templates to define work phases that will optimize the process of software integration solution, in a similar way, in any project, according to a whole work context and also according to factors outlined above, we will choose to use our own and specific methodology, which will be the most appropriate option to meet requirements and project necessities.

In terms of practical usage for application integration solution, I have used the proposed development methodology for an experimental prototype, as following: analysis and design phase has been extended in time so that the bussiness requirements on one hand, and objectives of the integrated system - on the other hand, is completely defined. After defining the technical solution and starting the development and implementation phases, I repeated, in iterative cycles, functional validation and acceptance testing of the modules, so that after completion of one iteration to get to know the real state of the project and to re-estimate total work effort, as well as refactoring / adapting the solution according to the feedback and validation of intermediate versions at the end of iteration. This incrementally work model contributed to greater predictability in project completion and to a smaller refactoring effort as compared to the requirement

changes that would have occurred at the end of implementation, if there were no partial validation, along the way, for atomic features.

Proposals to improve standard software integration methodologies:

- increased understanding of the business area to identify key business processes and the least important processes;
- give greater importance to the implementation of logical flows, described in terms of functional, operational and integrated business flows;
- understanding of data structures - many EAI projects are integrated only at the data level; it is very important to correctly identify data structures, descriptive data schemas and to accordingly apply business principles to determine which data streams coexist in an integrated system, where and why to use these data in order to make data mappings and transformation;
- understanding of the processes is essential to obtain a correct and functional model of the integrated system; project definition is dependent on how business processes are perceived, but also on the accuracy of the data model definition;
- increased attention to developing methods and technologies, rather than work processes;
- identification of user interfaces has great importance for understanding the integration points between disparate applications;
- understanding of events is a key factor for the functional model accuracy of the integrated system; events can be synchronous and, moreover, asynchronous calls in the automated EAI solutions;
- business process modeling using traditional techniques such as UML will refine them as to achieve the process, according to the specific integration project;
- „continous integration" process, especially automated testing in order to validate system quality;
- appropriate use of information technologies in the integration solutions development; it is a basic need to select the best technologies in order to develop integrated solutions, otherwise any mistake is costly in time – an appropriate option would be to build a pilot project;
- integrated system performance and scalability is a major concern with integration projects;
- importance of the prototype installation of intermediate versions of the integrated system, before running the final installation and launching it into production environments;
- maintenance effort and support are considerable in software integration solutions and should be treated separately.

I also made a research on the risks faced by software integration projects, by highlighting some key issues and implementing a risk analysis example for e-Health prototype. I highlighted the major risk factors and proposed a risk management plan, as a measure to reduce the negative impact on implementation processes of an integrated system.

Risk	Probability (1–low, 5-high)	Impact (1–low,5-high)	Exposure degree	Action
Data updating gap	1	3	3	Acceptance
Misuse of the system	2	4	8	Avoidance
Data non-compliance	1	2	2	Prevention
Incomplete business requirements	3	3	9	Contingency plan
System unavailability	1	4	4	Transfer

Fig. 2. Risk assesment

For each of these risks I have probably, given the appropriate points of occurrence and expected impact, causing a degree of exposure of project risks. Prioritizing, then, risks after exposure, we see that the most dangerous risk for the integration project is the lack and incompletion of requirements analysis. For each risk noted in the figure above, I have shortly proposed actions to take in order to minimize negative impact and effects, actions that should be the most appropriate strategy to consider. Extending risk analysis, we can discuss here also about positive risks, events that can bring benefits and new capabilities to project success.

In other words, the proposed methodology was applied naturally in the e-Health prototype, proved to be both a way of control and predictability, offering in the same time operational support for carrying out a complex integration project.

5. Implementation of a cross-platform software system

The proposed solution for enterprise applications integration is located at the middleware level and aims to integrate functionalities and heterogeneous data sources, by interoperation of heterogeneous information technology, data sources of different origins and different business functionalities. Also, in the practical experiment I will emphasize the use of uniform policies for communication between applications, by implementing software development over established protocols and standards. Case study performed and presented in this paper starts from the goal of systems integration process, the complexity of its component phases, meaning the creation of an integrated information system, having at least the following attributes: comprehensive, coherent, independent of geographical, social, national or business processes logic limitations.

I will describe in this section a platform for software systems integration, developed in Mule, which aims to highlight aspects of the application integration, such as: integration at data sources level, service based integration, business processes integration, through implementation of components focused on the integration of data, services and processes. I choose Mule platform for software prototype development because it is an efficient, complex and frequently used development environment for integration of information technologies. To illustrate the theoretical and practical concepts presented in previous sections, according to the implementation which is further described in detail, I present the prototype of an integrated system for e-Health. From functional point of view, the experiment represents an interoperability model between a medical system and financial system. In terms of working context, the integration prototype combines development technologies / platforms like Java, .NET, Microsoft SQL Server, MySQL, Mule, Apache, web-service based architecture (SOA, WOA), service and process based,

ODRA, ORM. The project aims to achieve an integrated sistem and consistency at the level of data, services and processes, by combining and integrating heterogeneous information technologies.

Besides service level integration prototype and specific calls that orchestrate the communication between the two heterogeneous systems, the next level of complexity is demonstrating a non-trivial mechanism of communication through messages. I will present architectural and functional model underlying messaging communication between software systems located on both sides of the enterprise web service bus.

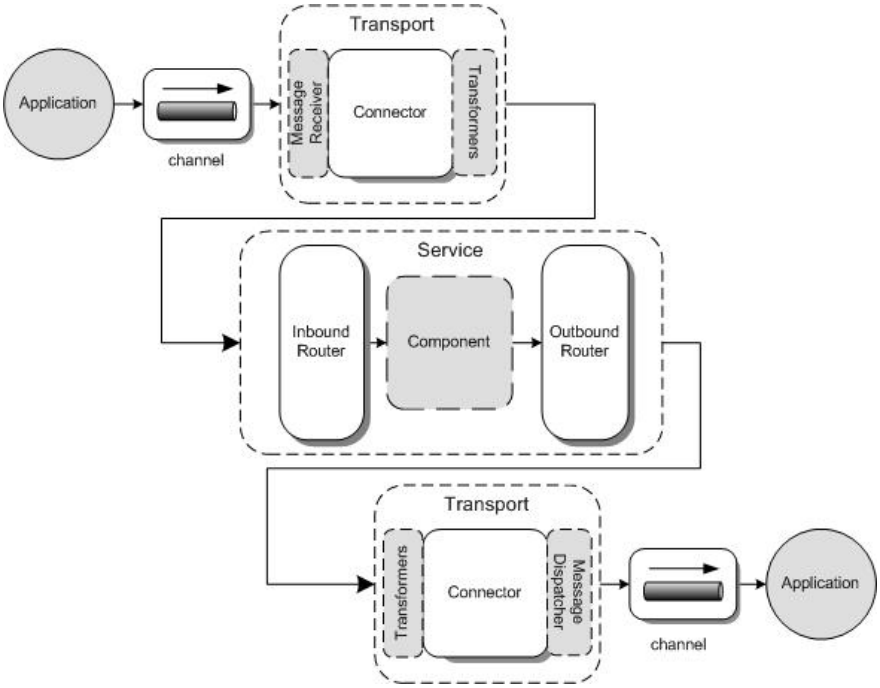

Fig. 3. Mule architecture

Mule supports transport protocols used in distributed programming, WS-* standards, JMS, File, SMTP. Client applications publish client requests on pre-defined communication channels. Requests are processed by the integration server application through services. Client applications access the services of ESB through secure communication channels and the flow transfer is done by specific invocations of connectors which implements the connection between client and server calls integration. Processing activities that the integration server performs in order to operate on incoming messages, for communication reasons with other client applications, are a result of transformation calls that specialized components perform inside ESB. ESB messages arriving through protocols / mechanisms of transport services are configured in Mule and processed on the application server side. A service component includes two processing end-points: an entry point and one exit point. The entry point is an endpoint that specifies the communication channel can be used, what messages can be transmitted and how to transfer such calls to Web services through a .NET proxy or a through a JMS message queue.

As a case study based on theoretical elements on integration of information technology and XML web services already mentioned, I propose a prototype implementation of an integrated system through web services technologies and Web 2.0 standards. This practical experiment

highlights ways to integrate enterprise applications based on integration of information technologies, in addition to a complex architecture based on an integration server. In this respect, I followed techniques and integration between Java and .NET platforms at the level of database, functionality and user interface applications. Working context is given by an architecture based on XML Web services implemented using Java platform. Data sources are heterogeneous, relational DBMS, MySQL and Microsoft SQL Server and will be integrated into a global catalog on Microsoft SQL Server. The purpose of the prototype is to demonstrate how we can integrate up to the level of services, applications built on heterogeneous data sources.

Coming back to the e-Health integration prototype, the business flows architecture can be described as following: in the medical system we have dictionaries of patients, delivered medical services under financial articles mapping and information on financial status of the services performed by physicians. On the other hand, the financial system have invoiceable items – corresponding to medical services from healthcare application – along with prices and financial attributes, customers - corresponding to patients from healthcare application, financial transactions - the equivalent of medical consultations, financial form and generated financial documents - invoices and details.

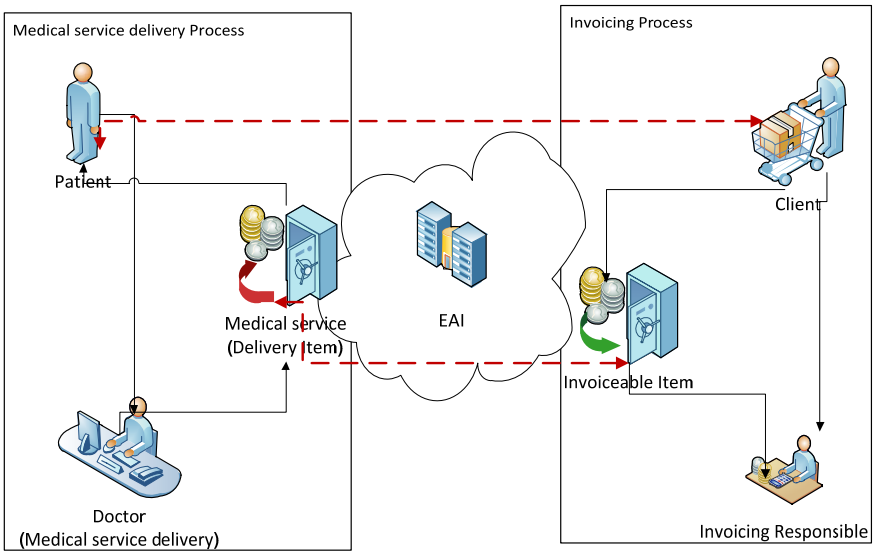

Fig. 4. Business flow design

Medical system manages entities like patients and consultations and receives from the financial system billable items, deliverable items/services, notification about patient billing / payment. At the other endpoint of the enterprise service bus there are integration services and business processes of the financial system, which manages items and bills and communicate both with the medical system for receiving patients as the equivalent of customers and for intercepting transaction execution as the deliverance of medical services in the consultations.

Using this functional model, I have designed an integrated system implemented in Mule, using some design principles to work in the following integration schema: declarative addressing and XML configuration for service invoker, SOAP formatting messages sent inside the service bus and web proxy service calls to end-point. I used the .NET environment developer tools and Visual Studio 2010 to generate references to proxy for Web Services.

After generating web service brokers, client applications generate in the form of .NET proxy, on the one hand, and within ESB, Apache proxy, on the other hand, the distributed communication architecture is the following: a client application, i.e. the medical system, calls a local method to process entity "delivered service into consultation"; local method calls a web service exposed by Mule integration server, it makes medical entity to be passed internally within the service bus, which will be given to a web service component type to the other endpoint of the integration server. This service will call an internal method of the other client application, i.e. the financial system, that will process information from another business perspective: invoicing patient medical services. Between the exchanges of messages in the web service bus, working entities are subject to transformations and serialized/de-serialized. Generic objects are serialized in SOAP messages, then transformed into XML data, SOAP and retransmitted as serializable objects to client applications, to be deserialized for internal usage purposes.

Software architecture used for exposing functionalities of business processes as web services is based on a SOA virtualization engine, which allows the definition of services that can be accessed via FTP, XFB, EJB, JMS, SOAP / HTTP. The access to web services is done through UDDI or graphical interfaces provided by the development environment tools, the main purpose of SOA being service re-usability. UDDI registry allow the attachment of metadata objects that virtualize access to services, such as status or version identifiers for each called service instance.

The conclusion of the project of integrating information technologies Java and .NET using the Mule platform, is that I have proposed an efficient and interesting way of building a reliable, scalable and extensible prototype for enterprise application integration. The project can be easily extrapolated and adapted to complex business processes, in order to achieve a fully integrated IT solution. Developing this prototype I reached to demonstrate an original formula for a solution of integrating information technologies for enterprise application development.

6. Business process modelling and optimization based on integrated software systems and workflow analysis

The need for enterprise application integration result from the need for networking and communication between different types of applications. Relative to disconnected systems in terms of costs versus benefits, there are two solutions to meet the need to integrate existing software products: rewriting them for intercommunication or software integration for their existing IT solutions - that means integration through coupling / decoupling mechanisms governed by EAI principles.

Various aspects of business processes modelling and optimization in terms of information systems, start from the use of existing information systems, restructured/refactored systems or integrated systems.
Based on the comparative approach in terms of cost and efficiency, depending on circumstances and context of business (designed business area, dynamism of the business domain, scale and complexity of software product, architectural design of the underlying software systems, the degree of use and efficiency) will be chosen one of these options: to rewrite existing software products, according to new requirements, or to define a solution for enterprise application integration, starting fro legacy applications.
Whatever the choice made, system development related activities require additional resources and maintenance efforts. As the level of abstraction of a system is higher, the more simple the process of reengineering becomes and there will be available more information about program and data structure, models of business entities, design models for data and control flow, UML class diagrams, states, actions, workflows.
The basic conclusion is that systems require additional efforts and extensible, automated architectures, in order to maintain and simplify the structure and to deal with the dynamism of business environment. Any chosen solution must be optimal no matter the considerents taken into account. In particular, integrated software solutions are subject to software reengineering, in order to optimize business processes and to better exploit the new system.
I will define, based on personal experience and information systems efficiency theories, an own assessment of the effectiveness system for an enterprise application integration, in the medical field. The main advantages of integrated information system usage are:

- removes difficulties and low speed in the operation of data, restricting users to be proficient;
- offers real time data acces and operation;
- removes multiple data operations and the need of greater allocation in order to check data consistency;
- eliminates gaps in information system, information redundancy and long time development of various internal processes (billing, contracting, commissioning, monthly reporting);
- reduces cumbersome tracking business costs distributed on cost center and budget lines, allocated to business size.

Conclusions

Research done in integrating enterprise applications, as described in this paper, captures more fundamental aspects, theoretical and practical, existing and new concepts of EAI. Based on existing theories, but also practical and methodological aspects of the existing implementations of EAI, I defined a solution for enterprise application integration on multiple descriptive levels: software product development methodology, technical implementation and functional analysis. I also stressed the context of these approaches, methods of optimization of business processes through an integrated approach, options for achieving a fully integrated information system respecting the principle of maximizing efficiency. Based on personal experience, but also on standard theories in software integration area, I outlined the advantages and disadvantages of various solutions and I have highlighted generally accepted principles in case of dynamic software systems over a certain level of complexity. Software prototype for enterprise applications integration for e-Health was designed and developed with the latest information technologies and it is a functional system that demonstrates the practical and experimental utility, of an integrated information system, built with a personal vision.

References

[1] Iulia Surugiu, *Integration of Information Technologies in Enterprise Application Development*, PhD. Thesis, 2012

[2] Faisal Aburub, Mohammed Odeh and Ian Beeson, *Modelling non-functional requirements of business processes*, Information and Software Technology, Issue 49, 2007, pp. 1162–1171, ISSN: 0950-5849

[3] Joseph Barjis, *The importance of business process modeling in software systems design*, Science of Computer Programming, Issue 71, 2008, pp. 73–87, ISSN: 0167-6423

[4] Bih J., *Service Oriented Architecture (SOA) - A New Paradigm to Implement Dynamic E-business Solutions*, ACM Ubiquity, Vol. 7, Issue 30, 2006, ISSN: 1530-2180

[5] 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, Issue 3, pp. 647-655, 2006, ISSN 0018-8670

[6] David Dossot, John D'Emic, *Mule in action,* Manning Publications, Greenwich, 2009, ISBN 978-1-933988-96-2

[7] Kevin B. Hendricks, Vinod R. Singhal, Jeff K. Stratman, *The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations*, Journal of Operations Management, Issue 25, 2007, pp. 65–82, ISSN: 0272-6963

[8] Xin Jin, *Research on the Model of Enterprise Application Integration with Web Services*, Proceedings of the 3rd WSEAS International Conference on COMPUTER ENGINEERING and APPLICATIONS (CEA 2009), ISSN: 1790-5117, ISBN: 978-960-474-41-3

[9] Shahed Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice)*, O'Reilly Media, 2009, ISBN:0596802765.

Iulia SURUGIU has graduated The Bucharest Academy of Economic Studies (Romania), Faculty of Cybernetics, Statistics and Economic Informatics in 2007. She is a Ph.D. Candidate in Economic Informatics with the Doctor's Degree Thesis: "Integration of Information Technologies in Enterprise Applications Development". Her research activity can be observed in scientific papers and proceedings achievements, among which:

- "Information Technology Standards – A Viable Solution to Reach the Performance", The 12th WSEAS International Conference on Automation & Information (ICAI '11), April 2011, Brasov, Romania;
- "Information Systems Integration, A New Trend In Business", The 10th WSEAS International Conference on Applications of Computer Engineering (ACE'11), March 2011, Spain;
- "Ways to Increase the Efficiency of Information Systems", The 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Databases (AIKED'11), February 2011, Cambridge, UK;
- "Advanced e-Government Information Service Bus (eGov-Bus)", The 9th International Conference on Informatics in Economy, May 2009, Bucharest, Romania;

- "Academic Research Information System", Journal: JAQM (Journal Of Applied Quantitative Methods), issue 1, vol. 3, 2008;
- "Medical Virtual Public Services", Journal: Informatics Economics, issue 1(45), 2008.

SOA - An Architecture Which Creates a Flexible Link between Business Processes and IT

Radu Stefan MOLEAVIN
Bucharest Academy of Economic Studies
ROMANIA
ramos_cisco2003@yahoo.com

To be viable, a company must be adapted continously to the market's requirements. The adaption of a company to the market's requirements means also changes in the business processes of the firm. Till several years ago, in any enterprise there was a strong link between business processes and IT applications. Any changes in the business processes lead to a change in the IT applications. This means that any change in the business side involves time, human resources and material resources consumption.This fact can be translated in a slow adapting to the market changes and a cost increasing for changing IT applications side necessary for their adaptation to new business market. Today the market changes quickly and it has a great development; this means that the enterprise adaptation to the market's requirements must be done quickly and with low consumption of either human or material resources. Also, the market changes which cause changes in the business processes should not cause major changes in the IT area. This goal can be achieved using an architecture services-based (SOA) for the enterprise. This architecture allows us, like in a puzzle game, to use the same pieces of material ("services") to build different models ("business processes").

***Keywords**: SOA, Web Services, Business Processes, ESB*

1. The need of integration. In every enterprise, focus of business environment is concentred to relationships with their partners, customers and suppliers, as well as new compliance regulations and compliance with applicable laws. In point of view IT, in major part of enterprises was developped over time, many applications as:Supply Chain Management), CRM(Customer Relathionship Management), Human Resources),ERP(Enterprise Ressources Planning) and other applications as Health and Safety Managment,Communication Management etc,).These systems use: different platforms, operating system and program languages.As is defined in [1] "*...old method, technology, computer system, or application program that continues to be used, typically because it still functions for the users' needs, even though newer technology or more efficient methods of performing a task are now available....*" this are legacy systems.These legacy systems was built to support one business process or part of business processes and comprise computers, programs, databases, networking and so on.This IT systems was develloped in decentralized manner. Every IT system was built for one department as own application, without taking into consideration applications built by other departments within the company.So these IT systems cannot comunicate each others.For this reason when a business processes is changed, all IT system which support those business processes have to be modified.This lot of IT systems has become a brake for business process enhancing. For this reason IT systems and Business processes have to be linked into an integrated system, which can ensure operational performance and competitive advantages.It should be better as every department of enterprise to share accurate and reliable data , to integrate their IT systems , in order to offer a better support to the decision making process concerning the overall organization's activity.

2. SOA-Scope,Goals and Benefits.

In some words it have to find a solution which have to solve some challanges [2] from Chapter 1: Alignment of IT with business goal, Responding quickly to new business opportunities,Getting the competitive advantage,Handling the launch of new products/services , while managing existing ones an Handling regulatory and government requirements with ease.

The solutions at problems above can be solved using SOA which is defined in [3]:

„SOA is an architectural paradigm for dealing with business processes distributed over a large landscape of existing and new heterogeneous systems that are under the control of different owners".

SOA is essentially a distributed architecture, with systems that include computing platforms, data sources, and technologies. A distributed architecture requires integration.

Integration software provides the bridge between the legacy systems and SOA,SOA represents a software architecture that is business-oriented and integrates the business tasks as a set of interconnected and reusable services, which communicate with one another.

„SOA establishes an architectural model that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in support of the realization of the strategic goals associated with service-oriented computing". [4].

Strategic goals and benefits are:

-Increased Intrinsic Interoperability- Interoperability refers to the sharing of data and more interoperable programs exchange easier information.In this case it can say that integration is a process that enables interoperability.

-Increased Federation- A federated IT environment is one which resources and applications are united and have individual autonomy and self-governance.This is succeded through deployment of standardized and composable services each representing a piece of the enterprise and expresses it in a consistent manner.

- Increased Vendor Diversification Options. Vendor diversification refers to the ability an organization to choose best technology innovations and use them within one enterprise. when is required.This help enterpise to change quicly business processes to at changes of market.

- Increased Business and Technology Alignment

-Increased ROI – Services have increased reuse potential that can be realized by allowing them to be repeatedly assembled into different compositions.Once created a service can be used as simple service or as composed service .In this way cost for design services is decreasing

-Increased Organizational Agility- Enterprise agility, on refers to efficiency with which an organization can respond to change. Because services have developed as reusable IT assets, they can be repeatedly composed into different configurations. As a result, the time and effort required to automate new or changed business processes is reduced.

- Reduced IT Burden- Applying service-orientation approach in an IT enterprise it can reduce redundancy, size and operational cost, and overhead associated with its governance and future evolution.In this case IT enterprise is agile department which contribute at increases in efficiency ,cost-effectiveness and to realisation of strategic goals.

3.Services

In the context of SOA, a *service* is a function performed by an application. A function is coded only once and then reused wherever it is needed. "Basically, [5] service is a self-contained (offers different functionalities related to one business or technical area/sub-area), cohesive (all related functionalities are placed together), black box (consumers know nothing about its internals, and underlying technologies) software

component that encapsulates a high level business/technical concept that can cover specific area of whole system".

In point of view of granularity, a service can be designed to *be fine-grained* or *coarse-grained.* A fine-grained service can handle small and specific functionality (e.g. log service, database access service, fees calculation service). Fine-grained services are reusable. A coarse-grained service is more powerful service that usually is composed of many fine-grained services and handle more complex and wide range of related functionalities (e.g. loan management service, purchase order service, etc). Coarse-grained services are hard to reuse.

A service might be *stateful* or *stateless.* *S*tateful service retains and manages service states during its execution, between different requests (e.g. a service that is responsible for handling loans must be stateful in order to keep state information about the loan being processed for client). A stateless service does not retain its states between different requests. This type of service is the most used type of services in SOA, because it allows loose-coupling between requestors and offered services.

Also, a service can be *short-lived* or *long-lived.* The execution of transactions offered by a short-lived service can take sub-seconds or seconds to finish working and returning back the generated results (e.g. getting or updating a database record such as customer information). Execution of long-lived service may take some minutes for starting and will take some days or even months to reach the final state (e.g. a loan management service may take months to grant a new loan to one customer).

Technically, a service is composed of three parts shown as

Fig:

• **Contract:** It provides both *formal* and *informal* specifications of service. Formal specifications use description languages WSDL to describe information related to technical areas of service such as underlying programming language(s), middleware(s), network protocol(s), and other runtime aspects.

Informal specifications are textually presented to provide general information such as the purpose, functionality, constraints, usage of exposed service, and expected response time. A contract [3] is the complete specification of a service between a specific provider and a specific consumer. From a consumer's point of view, it defines "everything you have to know when using this service," so that (ideally) no doubts remain.

•**Interface:** It provides technical representation of service operations (contain information about public operations, parameters and return types) that are available to be invoked by clients.

•**Implementation:** It contains logic of service that might be related to accessing data, business logic, etc. This implementation logic could be encapsulated internally within service itself or it may be provided by other external artifacts (e.g. other programs, code libraries, components, legacy systems, etc). Implementation of Service as Web service can be written in any language and in any platform. The technologies of Web service makes this implementation logic accessible using standard Web technologies, such as HTTP and Web browser, and result in a faster and more dynamic communication for connected applications. Web services architecture is service-oriented and is shown in Figure 2.A consumer searches(using message SOAP- Simple Object Access Protocol specific format for exchanging Web Services data over HTTP) for an

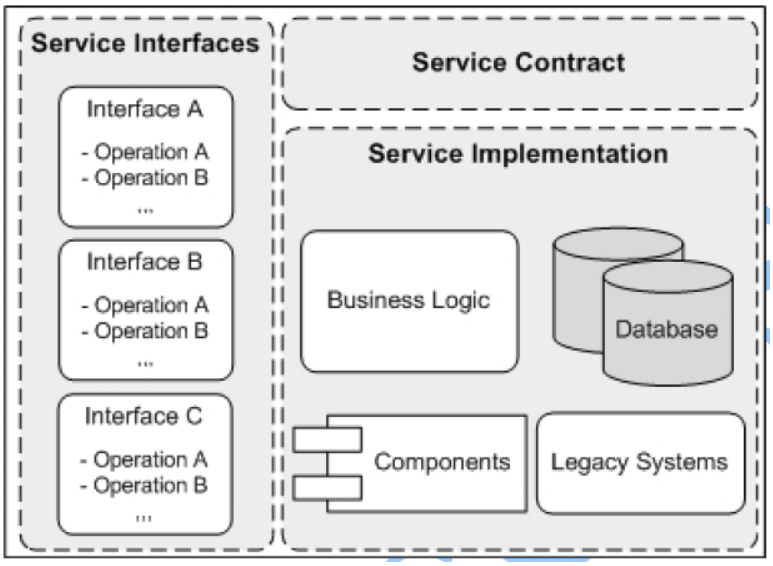

Fig.1. Essential Service Elements [5]

available provider of a service to execute a certain function. A service registry(UDDI- Universal Description, Discovery, and Integration specification can be used by the service providers to advertise the existence of their services and by requesters to search and discover already registered services) contains published services and when there is a request for a service, it returns back the information which allows requestor to locate and invoke the service.

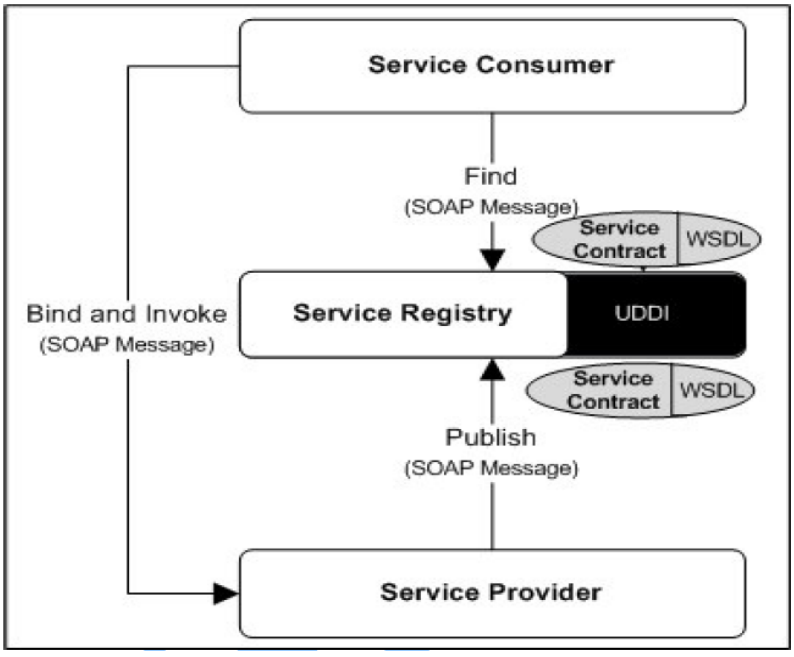

Fig.2. Web Service Arvhitecture

4. SOA Architecture

The structure of a Service-Oriented Architecture can be mapped with a layer model. SOA is an n-layer architectural model as is shown

Fi.

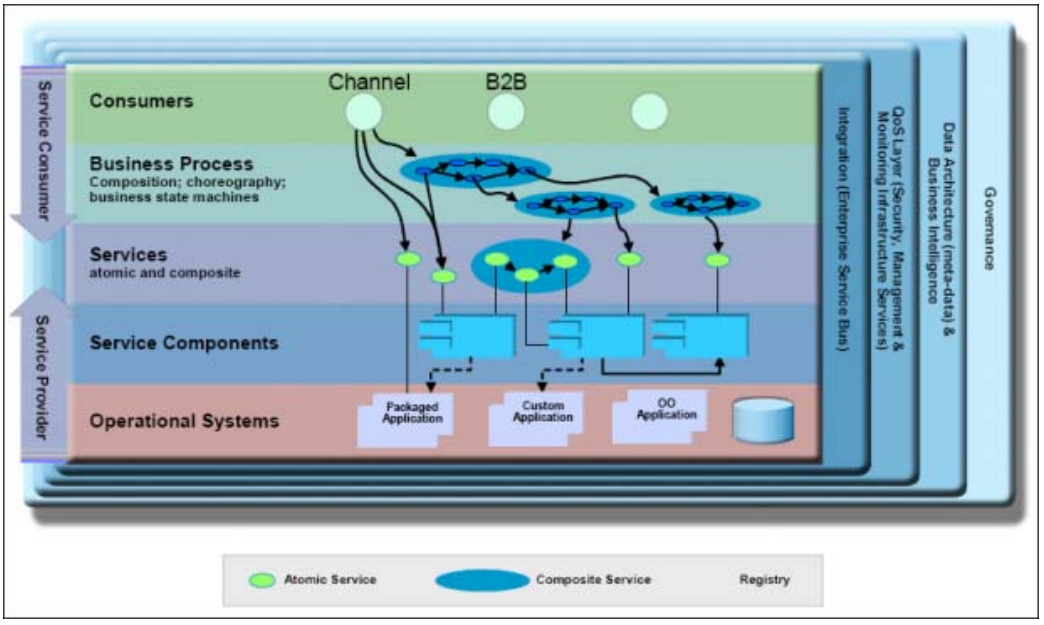

Fig.3. SOA Foundation Reference Architecture [6]

SOA layers are:

Operational Systems Layer: This layer contains existing applications of enterprise as object-oriented systems, business intelligence etc.These applications are known as Custom or Legacy application. These applications provide the background for services and each of them has its own proprietary structures, databases and other system resource access. This layer contain also technolocical layer which contain: Application Platform,Technology platform,Operating system.

Enterprise Components Layer: These are specialized components to provide certain functions and requirements for services. They are the business assets for service implementations, and other system necessities such as management, availability and load balancing of services.

Services Layer: This layer contains the actual services which can be discoverable and invoked by other applications to provide a specific business function for enterprises.Services are held in Enteprise Repository Services.Services are splited in three categories:Local Services(intenal developed in enterprise),Remote services(outsources services) and Enterprise Information system services(internal developed in enterprise using components of Enterprise Information system(legacy systems)).

Business Process Layer: The services can be composed into a single application through service *orchestration* or *choreography*, which supports specific use cases and business processes.

Presentation Layer: It provides user interfaces for:internal users,customers and providers of enterprise.They are consumers [6] of the processes and services.

Comunication between services and Business Processes is facilitated [6] by ESB(Enterprise Service Bus)-Fig

.

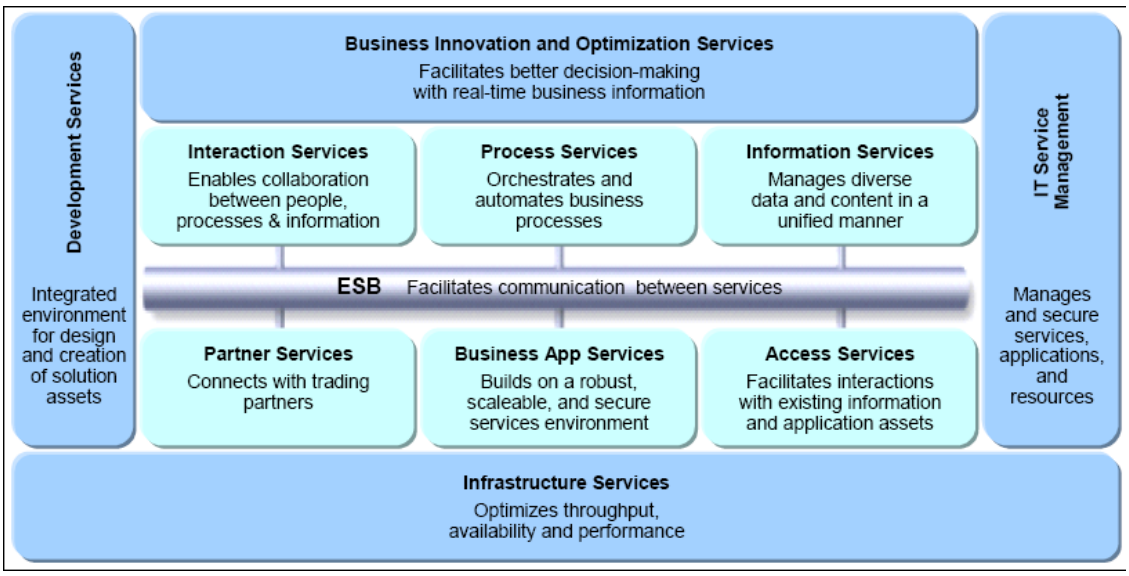

Fig.4. SOA Foundation Reference Architecture: Middleware Services view [6]

Core components of Middleware Services are:

- Interaction services- provide the capabilities that are required to deliver IT functions and data to users;
- Process services- provide the required control to manage the flow and interactions of services for implement business processes;
- Business application services- are used by service consumers for access to portal or business processes.
- Information services-Manage disparate data source in unified manner.
- Access services- provide a link between core applications of enterprise, enterprise data stores and the ESB to incorporate services delivered by these applications
- Partner services- Partner services provide capabilities for business processes which interact with outside partners and suppliers.

Part of SOA,an ESB is defined as [7]
„... an open standards, message based, distributed integration infrastructure that provides routing, invocation and mediation services to facilitate the interactions of disparate distributed applications and services in a secure and reliable manner."

Main properties of ESB are:
Invocation-Is the ability of an ESB to send requests and receive responses from services.That means that an ESB has to support the standards for web service communication including SOAP,UDDI and the WS-* family of standards.
Routing-is the ability of ESB to decide the destination of a message during its transport.
Mediation-refers to all transformations or translations between disparate resources including transport protocol, message format and message content.
Adapters- connect to the native APIs and data structures of legacy applications and present a standard interface, which makes it easy to reuse business logic and data.
Security-means for an ESB that is able to encrypt and decrypt the content of messages, handle authentication and access control for messaging endpoints .
Management-ESB has to provide: audit and logging facilities for monitoring infrastructure,a mechanism for configuration and administration of bus and also tools for usage metering.
Process Orchestration-ESB may include an engine to execute business processes described with the Web Services Business Process Execution Language (WS-BPEL) which coordinates the collaboration of the services connected to the bus.

Complex Event Processing-ESB may include mechanisms for event interpretation, event correlation and event pattern.

5. SOA-Flexible link between Business Processes and IT

A business process is a set of specific, tasks performed by people and systems and designed to achieve a predetermined outcome. Companies are trying continuu to improve their business processes , to adapt at market changes, using new technology. As tool for modelling processes, companies are adopting BPM (Business Process Management) which helps to align IT systems with business's strategic goals by creating well defined enterprise business processes, monitoring their performance, and optimizing for increase efficiencies. Each business process is modeled as a set of individual processing tasks. These tasks are typically implemented as *services* within the enterprise. BPELs are the languages for process management. Business Process Execution Languages (BPELs) are XML(eXtensible Markup Language) used for modeling business processes using orchestrating, coreography, and controlling web services.

As is shown in figure 3,4 and 5, SOA based system is composed of Service Consumer, Service Bus(Enterprise Service Bus), and Service Provider.Service Provider start at level Framework and application.System Applications, based on various implementations of language and platform, provide Application Services which use adapters to comunicate with ESBApplication services are exposed in Regpository of ESB. Here are exposed also Created services which are new services which don't use functionalities of legacy System Applications..These can be Internal services(create within enterprise) and External services(create outside enterprise).

Business processes are composed of sub-processes and activity.If activity is simple can be executed by one service, else is complex can be executed by a composed service from Service Orchestartion Layer of ESB.In generally Sub-processes are executed by composed services from Orchestration layer.

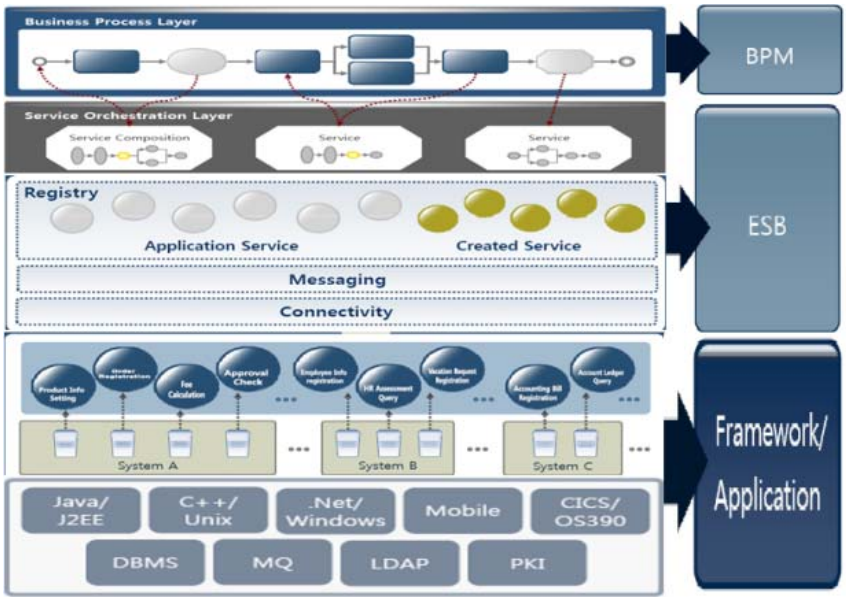

Fig.5. SOA Architecture [Adapted [8]]

6. Conclusions

BPM and SOA provide a good combination for enterprise. BPM provides a tool for defining businesses processes and also other important posibilities of monitoring and managing these processes.
Services provide the functions that support these processes.
SOA provides infrastructure(ESB) and the capabilities for services to be combined and to support and create an agile, flexible enterprise.

References

[1] Wikipedia. www.wikipedia.org. *http://en.wikipedia.org/wiki/Legacy_system.* [Online]

[2] G. Behara http://www.bptrends.com/publicationfiles/10-08-ART-SOA%20and%20Integration-Gopola-final.doc.pdf. *http://www.bptrends.com.* [Online] October 2008.

[3] N. Josuttis, *SOA in Practice.* s.l. : O'Reilly Media, 2007. ISBN-10: 0-596-52955-4;ISBN-13: 978-0-596-52955-0.

[4] T. Erl,. http://www.soaprinciples.com/. *http://www.soaprinciples.com/.* [Online]

[5] Q. Hassan http://arxiv.org/ftp/arxiv/papers/1202/1202.6623.pdf. *http://arxiv.org.* [Online] Faculty of Computers and Information Systems,Mansoura University, Mansoura, Egypt, 2009.

[6] M. Keen http://www.redbooks.ibm.com/redbooks/pdfs/sg247234.pdf. *http://www.redbooks.ibm.com.* [Online] August 2006.

[7] F. Menge http://kanagwa.com/assets/21/Esb1.pdf. *http://kanagwa.com.* [Online] 2007.

[8] P. S. Beom http://conference.kiec.or.kr/presentation/Sang-bum.pdf. *http://conference.kiec.or.kr.* [Online] October 17, 2008.

SAS Enterprise Data Integration Server - A Complete Solution Designed To Meet the Full Spectrum of Enterprise Data Integration Needs

[1]Silvia BOLOHAN, [2]Sebastian CIOBANU
[1]Marketing Manager, [2]Implementation Project Manager
SAS Analytical Solutions Romania

This paper is about why is Data Integration important for organisations around the world. Organizations struggle daily with the challenges of large distributed data volumes, inconsistently defined data across disparate systems and the high expectations of data consumers who depend on information to be correct, complete and available when they need it. SAS Enterprise Data Integration Server provides a comprehensive solution that enables organizations to solve these challenges in a timely, cost-effective manner with the ability to efficiently manage data integration projects on an enterprise scale, increasing overall productivity and reducing the total cost of ownership.
Keywords: *metadata, data cleansing, data integration, ETL, ELT*

1 Introduction

What does SAS® Enterprise Data Integration Server do?

SAS Enterprise Data Integration Server, featuring DataFlux® technology, is a powerful, configurable and comprehensive solution designed for managing big data. It can meet a wide variety of data integration requirements, from small tactical projects to strategic business initiatives.

- Access virtually all data sources.
- Extract, cleanse, transform, conform, aggregate, load and manage data.
- Support data warehousing, migration, synchronization, federation and provisioning initiatives.
- Support both batch-oriented and real-time master data management solutions.
- Create real-time, reusable data integration services in support of service-oriented architectures and data governance.

Why is SAS® Enterprise Data Integration Server important?

It enables organizations to efficiently manage data integration projects on an enterprise scale in a timely, cost-effective manner and meet the high data quality expectations of information consumers.

For who is SAS® Enterprise Data Integration Server designed?

It is designed for organizations in all industry sectors that are implementing one or more data integration projects, dealing with changing business landscapes and business-driven IT initiatives, trying to meet regulatory requirements, or implementing data governance.

2 SAS® Enterprise Data Integration Server

Product overview

SAS offers the only comprehensive enterprise data integration environment that is built from the ground up to meet the full spectrum of enterprise data integration needs. Instead of linking and managing technologies from different vendors, SAS Enterprise Data Integration Server provides a collaborative design environment promoting object reuse and sharing, administrative controls, wizard-driven design process workflow, and ease of use and maintenance. This flexible, reliable solution can access data from virtually any system in any form, transform and cleanse data even in real time, and handle data migration, synchronization and federation projects all through a versatile services environment that is easy to deploy and maintain.

Interactive data Integration Development Environment

A graphical user interface (GUI) simplifies and speeds projects with wizards, extensive built-in transformations and powerful productivity enhancements, all while providing a single point of control for managing complex enterprise data integration processes. SAS Data Integration Studio is easy to learn and use. It provides a collaborative environment that lets you build reusable processes to speed data integration development both now and in the future. It automatically captures and manages standardized metadata from any source, and enables you to easily display, visualize and understand enter-prise metadata and your data integration processes.

Connectivity and Data Access

Most organizations struggle with accessing the plethora of data sources (legacy, relational, flat files, XML, cloud data, text, etc.) that are necessary to support analytical systems. SAS Enterprise Data Integration Server provides connectivity to virtually all types of data sources and types, operating systems and hardware environments using both native access and open standards. It also supports the reading and writing of data from message queues and the sending and receiving of data to and from Web services.

Metadata Management

SAS provides a shared metadata environment that is both independent (for data integration) and part of SAS' comprehensive platform. Technical, business, process and administrative metadata is stored and managed in a way that leverages and facilitates reuse of existing table definitions, business rules and more. Navigational tools help users understand how the data was derived and where it is stored and used. Shared metadata provides a consistent definition across data sources to speed integration projects, simplify design and reduce maintenance costs.

Data Cleansing and Enrichment

There is an increased awareness, driven by compliance mandates and data breaches, of how data quality and data governance can directly affect the bottom line. This puts increasing pressure on IT to address potential data quality issues. SAS Enterprise Data Integration Server provides a single environment that seamlessly integrates data quality within the data integration process, taking users from profiling and rules creation through execution and monitoring of results. From data de-duplication (for example, within database marketing applications) to cleaning up data (for example, before storing in a data warehouse), SAS provides an enterprise approach that lets you develop and share a library of data rules and processes between projects and across the entire data integration solution. Organizations can transform and combine disparate data, remove inaccuracies, standardize on common values, parse values and cleanse dirty data to create consistent, reliable information.

Extraction, Transformation and Load (ETL) and Extraction, Load and Transformation (ELT)

Loading data warehouses and data marts within their allotted time windows, quickly building analytical marts for special projects, and creating extract files for reporting and analysis applications are tasks IT organizations face each day. SAS Enterprise Data Integration Server includes an intuitive point-and-click Design Editor window that allows developers to easily build logical process workflows, quickly identify the input and output data stores and create business rules in metadata, enabling the rapid generation of data warehouses, data marts and data streams. Users can also choose to have many transformations and processes take place inside a connected database, data warehouse or storage system. This is referred to as ELT, push-down or in-database processing, and can substantially speed up overall processing times by reducing unnecessary data movement. SAS

Enterprise Data Server uses visual SQL push-down to select the optimal processing approach.

Migration and Synchronization

Moving data from system to system is a constant activity in most organizations. Mergers and acquisitions result in multiple, overlapping systems containing information that often needs to be synchronized and ultimate migrated. Moving legacy data during upgrades and conversions is an on-going process, as is the movement of data into and out of ERP systems.

SAS Enterprise Data Integration Server provides the capability to migrate, synchronize and replicate data across different operational systems and data sources. The point-and-click process design editor makes it easy to document migration and synchronization processes in workflows that can be reused and modified for other projects. Powerful data transformations are available for altering, reformatting and consolidating information during these processes.

You also can build a library of reusable business rules ensuring that bad data is never spread from system to system. In this way, information delivered across all applications, systems, environments and geographies is up-to-date, consistent and accurate.

Data Federation

Organizations today have data stored and scattered in and across numerous data sources. Often what's needed is fast access to the most current operational data to support real-time analytics and reporting needs. SAS Enterprise Data Integration Server provides the ability to query and use data across multiple systems without the physical reconciliation or movement of source data. The logical semantic layer shields business users from the complexities of the underlying physical data. By avoiding unnecessary data replication and movement, it is possible to quickly and cost-effectively deliver up-to-date data that is consistent and accurate.

Master Data Management Support

SAS Enterprise Data Integration Server includes data-mastering capabilities that provide a basis for implementing master data management projects that enable you to identify, standardize and correct common master data such as customer and product data. Unsurpassed data access, profiling, enrichment, clustering and consolidation to clean, standardize and enhance data, and an intuitive development environment that is adaptable to each organization's technologies and standards, increase productivity and produce more rapid results.

SOA and Message Queue Integration

Organizations are challenged to improve operational efficiency, streamline processes and be more agile. Using a service-oriented architecture (SOA) approach helps IT ensure that various applications can communicate with each other to better meet changing business requirements. SAS Enterprise Data Integration Server delivers easily maintained data services that enable developers to build sophisticated data services once and deploy them across the enterprise for reuse.

Message queue integration is another way to reduce maintenance, integration costs and bridge new technologies. Using SAS Enterprise Data Integration Server, you can access message queues in batch or real time without the need for custom programming. Integration developers simply treat message queues as any other source and target. [1]

With SAS Enterprise Data Integration Server, you can define the propagation of information from table to table in your transformations. This shows an example of the default mapping rules being applied when mapping numeric to character columns, and character to numeric columns. This also shows the intelligence of the mapping display that uses color to indicate the presence of a transformational expression between source and target. Default mapping rules are predefined but can be extended to meet business-specific needs.

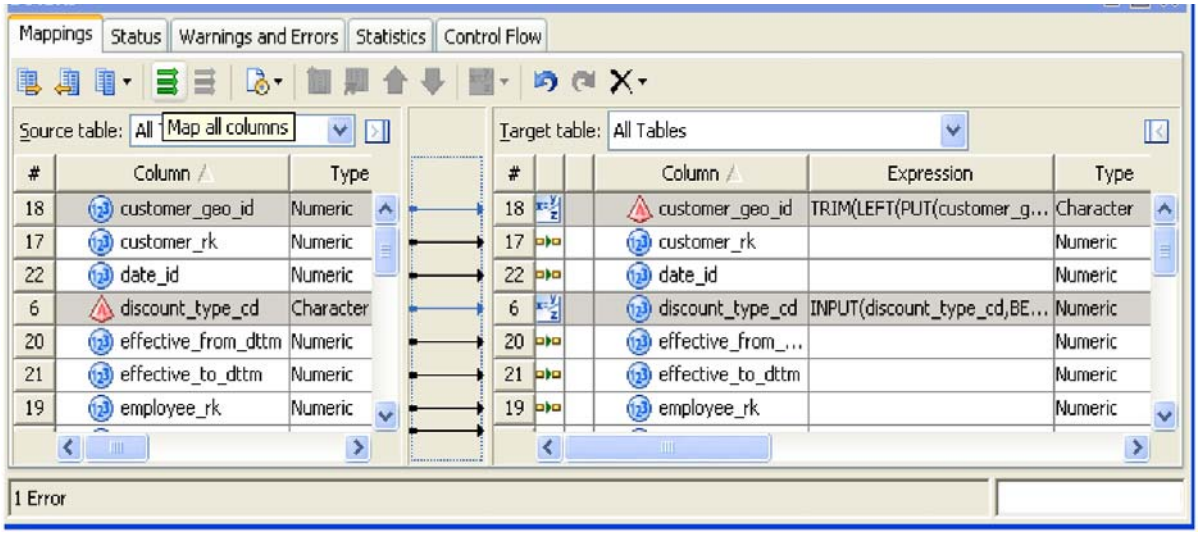

Fig. 1. Default mapping rules chart

3 Key Features

3.1. Interactive data integration development environment

• An easy-to-use, point-and-click GUI uses an intuitive set of configurable windows for managing data integration development processes.

- A visual, end-to-end process designer lets developers quickly build and edit processes.
- Drag-and-drop functionality eliminates programming.
- Wizards make it easy to access source systems, create target structures, import and export metadata, and build and execute ETL process flows.
- The multiple-user, multiple-level design environment supports collaboration on large, enterprise projects.
- Role-based permissions show users only what they are authorized to see.
- Customizable metadata tree views let users display, visualize and understand metadata.
- Dedicated GUI for profiling data to identify and repair source system issues while retaining the business rules for use in other ETL processes.
- Interactive debugging and testing of jobs during development and full access to logs.
- Check-in/check-out of jobs, related tables and objects; and job status viewing.
- Audit history lets designers see which jobs or tables were changed, when and by whom.
- Ability to distribute data integration tasks to nearly any platform and to connect virtually any source or target data store.
- Integration with third-party vendors Subversion and CVS provides enhanced version and source control features such as archiving, differencing and rollback.
- Job status and performance reports provide the ability to track metrics such as CPU use, memory, I/O, etc.
- Automated job deployment allows the use of common scripting languages to deploy SAS Data Integration Studio batch jobs in an automated manner.
- Enhanced SAS code import capabilities give current SAS users an easy way to import their SAS jobs and SAS code into SAS Data Integration Studio. Includes logging and error checking.
- Command-line job deployment for deploying single and multiple jobs.
- Enhanced data integration job orchestration (process flow).
- The ability to surface in-database scoring models within SAS Data Integration Studio.
- Enhanced connectivity to Aster Data, EMC Greenplum, Hadoop and Sybase IQ

databases with the ability to push down more processing to the databases. [2]

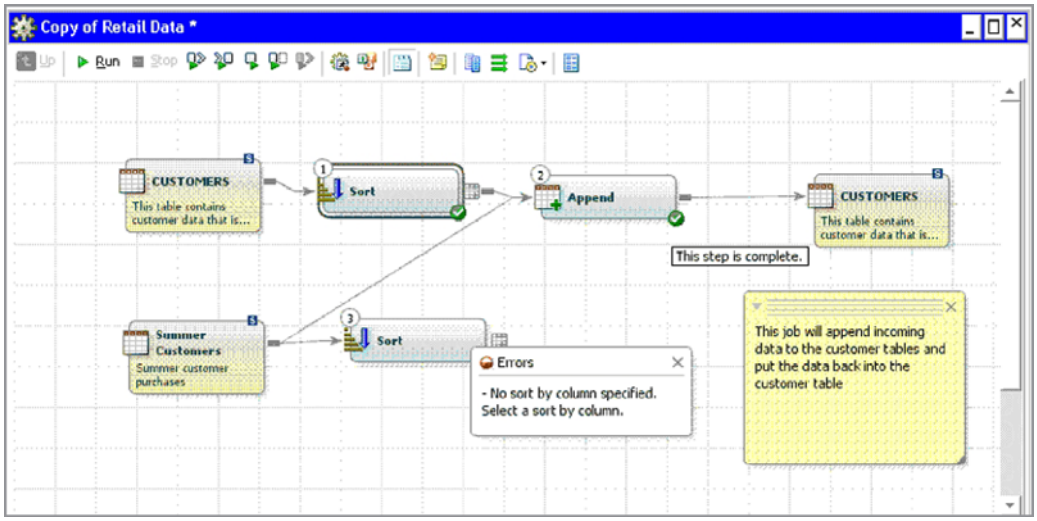

Fig.2. Retail Data ChartSAS Enterprise

Data Integration Server includes an easy-to-use and informative GUI. You build jobs by dragging and dropping data objects into the diagram area. You can add transformations such as sorts, joins and loads from a library and draw arrows to connect the objects together. Self-documentation is provided using annotated data, and yellow notes containing further information can be added by users.

3.2 Connectivity and data access

• Provides connectivity in batch or through message queues in real time to more data sources on more platforms than any other solution.

• Data access engines are available for enterprise applications, nonrelational databases, RDBMSs, data warehouse appliances, PC file formats and more.

• Specialized table loaders provide optimized bulk loading of Oracle, Teradata and DB2.

- File reader/writer available for Hadoop file system (HDFS).
- Support for Hadoop's MapReduce, Pig and Hive within flows.
- Data movement capabilities to and from Hadoop.
- A complete and shared metadata environment provides consistent data definition across all data sources.
- Native access methods deliver the best performance and reduce the need for custom coding.
- Support for message-oriented middleware, including WebSphere MQ from IBM, MSMQ from Microsoft, Java Message Service (JMS) and Tibco's Rendezvous.
- Support for unstructured and semi-structured data to parse and process files.
- Access to static and streaming data for sending and receiving via Web services.
- Expanded support for MPP databases: AsterData nCluster, EMC Greenplum and Sybase IQ, enabling more ELT pushdown and support for bulk-load utilities.
- Native support for SQL-based processing.

3.3 Metadata management

• Metadata is captured and documented throughout transformations and data integration processes, and is available for immediate reuse.

• Sophisticated metadata mapping column definitions from sources to targets, and for creating automated, intelligent table joins.
- New metadata search tool.
- Impact analysis for assessing the scope and impact of making changes to existing objects such as columns, tables and process jobs before they occur.
- Ability to determine the path, processes and transformations taken to produce the resulting information.

• Data lineage (reverse impact analysis), which is critical for both validating processes and building user confidence in data.
- Change analysis for metadata change technologies for quickly propagating discovery, comparison, analysis and selective propagation.
- Multiple-user collaboration support includes object check-in and check-out.
- Promotion and replication of metadata across development, test and production environments.
- Wizard-driven metadata import and export.
- Wizard for metadata column standardization.
- Metadata-driven deployment flexibility so that process jobs can be deployed for batch execution, as reusable stored processes or as Web services.

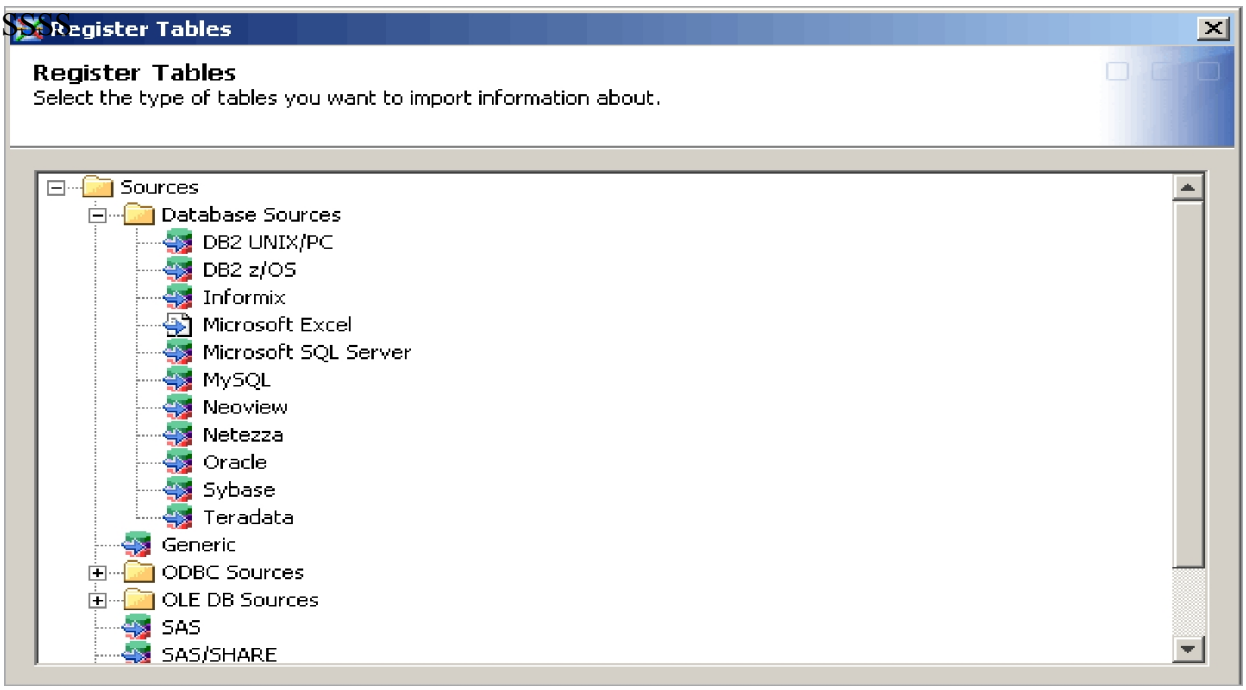

Fig.3. Register Tables Chart

The Register Tables wizard makes it easy to access data from many different systems, as well as read and manage metadata from external sources.

4. Conclusions

From legacy systems to ERP applications to data stored in Hadoop, data from virtually any hardware platform or operating system can be accessed, cleansed and processed. New source systems can easily be added and security is managed centrally. This saves time, shortens learning curves and gives decision makers the complete information they need.

Another conclusion is that a common repository enables the centralized storage, management and reuse of work based on user authorizations, reducing both

development and maintenance time. A GUI environment that is easy to use provides a standard interface for building and documenting work. Collaboration is encouraged and manual coding is available when needed. New team members can get up to speed quickly on work done by others, which is important when documentation is inadequate or missing.

A huge benefit of this tool is that you can manage security and administration at all levels. Reusable templates make it quick and easy to provide role-based authorizations and administrative privileges at the user, department or enterprise level.

SAS processes data fast! Organizations can take advantage of a grid-enabled load-balanced, multithreaded parallel processing architecture that can quickly trans-form and move data between different platforms and systems. SAS also supports zero data movement by using SQL pass-through into popular database appliances, including Oracle, DB2, Teradata, Netezza, SQL Server, AsterData and Hadoop.

Consistently getting correct data when and where it is needed provides increased confidence in the accuracy and timeliness of operational and business information. Data quality auditing tools monitor the quality of data in processes and source systems. Users can see where data originated and how it was transformed. Optional enrichment components can add value and ensure everyone receives the best possible data. Deliver consistent, trusted and verifiable information.

SAS offers the only comprehensive enterprise data integration solution that is built from the ground up to meet the full spectrum of data integration needs. It eliminates the piecemeal approach of linking and managing technologies from different vendors and provides lower overall cost, reduced risk and faster results.

References

[1] SAS Data Integration Studio 4.4: User's Guide, http://support.sas.com/documentation/cdl/en/etlug/65016/HTML/default/viewer.htm#titlepage.htm

[2] Base SAS 9.3 Procedures Guide, http://support.sas.com/documentation/cdl/en/proc/63079/HTML/default/viewer.htm#titlepage.htm

Silvia BOLOHAN is Marketing Manager at SAS Romania for 7 seven years. She leads and functions in the creation or production of marketing content for internal and external use in area of assignment. Silvia is responsible for developing and executing marketing strategy and/or programs for SAS products and services.

Silvia has given support for data analysis based projects such as customer segmentation, attrition modelling, customer lifetime value, etc. She contributes to the efforts of building and maintaining a comprehensive reporting and tracking strategy for campaign response.

Sebastian CIOBANU is SAS consultant for over 3 years in the Business Intelligence and Data Integration domain for the Banking sector. Projects he has worked for include Analytical CRM solutions, Data Mining and Sales Data marts. He has a BA in Economic Informatics and MsC on Databases from the Academy of Economic Studies of Bucharest. His areas of interest are: Databases, Data Modeling, Business Intelligence solutions and the Banking area.

Borderless Crime - Computer Fraud

Raluca Georgiana POPA
Department of Economic International Relations,
Bucharest Academy of Ecomomic Studies
ROMANIA
jpopa1961@yahoo.co.uk

Starting from the consideration that fighting cybercrime is a continuous process, the more the types of old crimes are committed today through modern means (computer fraud) at distances of thousands of kilometers, international cooperation is vital to combat this phenomenon.
In EU countries, still under financial crisis "the phrase", cybercrime has found a "positive environment" taking advantage of poor security management systems of these countries.
Factors that led criminal groups to switch "their activities" are related to so-called advantages of the "gains" obtained with relatively low risk.
In Romania, more than any of the EU member states criminal activities set as target financial institutions or foreign citizens, weakening confidence in financial systems and the security of communication networks in our country, people's confidence in electronic payment instruments and those available on the Internet.
Keywords: *Computer Crime, International Cooperation, Security, Computer Fraud, Computer Systems*

Expand information crimes in our country, is another aspect of the offense or another phase of its development to modern society.
We can say that some crimes have acquired a descendent character due to the used means that have evolved, but the same cannot be said about computer crimes that took a great extent, being in continuous growing
In Romania these types of crimes are on a lower step, reporting our country to other developed countries, because the nature of such offenses is favorable to any more or less developed country , the only difference consists of the mode of operation and other factors, but objective part is one and same, no matter where there is crime
Computer along with other equipment, is now the technical complex by means of which the criminals' illegal maneuvers are operated.
In the last fifty years, statistics show that computer crime has become a growing phenomenon
The computer could allow encryption of information, thus preventing access of investigators to them, and this is a possibility highly exploited by criminals, including terrorists.
The first laws against crimes committed by means of a computer, contained, in essence, provisions against acts of penetration database, of deceit and sabotage, and software piracy that is normally regulated by copyright law. It has also proved that drug trafficking, illegal arms trade, child pornography, online shopping, various forms of economic crimes and even some crimes on environmental protection, can be made by means of a computer.
Thus, at the end of 1986, the spectrum of these crimes has been extended to all offenses using automatic data processing as a tool for action, then it was also approached the term "Computer Aided Crime".
Cybercrime phenomenon has engulfed the economy, today no serious criminal organization cannot be imagined without computer support, and investigators must bear them in mind in their action research.
Evolution of organized crime in Romania in recent years is closely related to the evolution of cybercrime and the increasing use of ICT technology in committing crimes. Analyzes effected at the level of the

European bodies on Cybercrime trends, define organized crime as an important branch of organized crime in EU countries.
Following this development, in Romania too, a number of studies and assessments have been made, that identified some common characteristics of this type of crime taking place in our country or the perpetrators are Romanian citizens.
Thus, these activities aimed at obtaining financial products i.e. credit and payment systems offered by financial institutions and banking that members of these crime networks access fraudulently, causing significant damage.
Cybercrime has become a phenomenon both by the large number of cases recorded, the organization of those who commit such acts, in criminal groups, and by shifting groupings who are committing crimes such as international traffic of cars and people, to the crimes in the area of cybercrime.
Trend of organized crime, defines cybercrime as an important branch of organized crime in EU countries, according to analyzes conducted in the European organizations .

OECD expert group in 1993, defined computer actions such as: "any unethical or illegal behavior regarding unauthorized automatic data treatment and / or data transmission", definition whose utility is currently useful , even if spectacular development occurred on the design and use of computer .

Another definition is given by UNAFEI (United Nations Asia and Far East Institute for the Prevention of Crime and the Treatmet of Offenders), by computer crime, it is understood: ”any offense in which a computer or computer network is the subject of an offense, or a computer or computer network environment is the instrument of making an offense ". In the narrow sense, "any offense the offender interferes without authority with automatic data processing processes, represents a computer crime. [1]

According to ENISA (European Agency for Network and Information Security Agency), computer crimes are classified as:

- Computer fraud, computer forgery, damage to databases and computer programs, computer sabotage, unauthorized access to computers, unauthorized interception, reproductions of fraudulent schemes, no right to alter data or computer software, computer espionage, unauthorized use of a computer or computer programs protected by law .

The main factors for gang diversion to computer crimes are tied to large financial gain in a relatively short time and with relatively low risk. This phenomenon affects the image of Romania, as criminal activities that target financial institutions or foreign citizens, weakening confidence in financial systems and the security of communication networks in our country, people's confidence in electronic payment instruments and those available on the Internet. Examining criminal cases, involving a number of defendants prosecuted and aim at computer crimes, it appears that their commission is made by the same types of actions, namely: misleading persons in the circumstances of the conclusion of online contracts and introduction of computer data in a computer system. However all these having as a result in any losses in buyers’ property who have good faith and confidence in how to complete online business.

The prosecution bodies in Romania, (Service for Combating Organized Crime Valcea), have been reported by many foreigners on the fact that since 2005 they have been misled over the Internet by an organized criminal group , meaning that accessing sites E-bay or other companies specialized in selling products online, they have been in contact with them and negotiated by e-mail or telephone trading conditions of goods, especially agricultural machinery and mobile phones , and after they transferred the money to Romania, by Western Union or other payment method, representing an advance or the total price of

the contract object, the "so called" sellers have stopped all contact with victims. [2]

Finally, companies Ebay and Paypal have made complaints and have presented data and information from which content results that the organized criminal group held computer data ,by wise utilization were rigged users' legitimate accounts of the two companies, thus creating localized damage to their property .

In February 2005, the Singapore citizen Y.S., visited sites Ebay, in the idea of buying brand mobile phones Nokia9500, circumstance in which he established contact with the holder of the e-mail redbear925@hotmail.com who convinced him to transfer to Romania on behalf of defendant M.N., through Western Union Quick payment in two installments, Singapore's total of $ 18,860 . It is observed that the tenderer has induced the victim the idea that he represents a prosperous company, specialized in online transactions and also has imposed him his own rules for conducting the transaction, meaning that money would be transferred by Western Union, which was happened, and the good be sent by international delivery service, UPS, rules that have no relation to services Ebay si Paypal.

A brief analysis of the modalities of operation of criminal groups that operate online, thus, they, in 2005-2008, by unlawful accessing of accounts belonging to companies specialized in online sales, by creating trap sites and accounts in which content that falsely offered for sale various goods, mainly agricultural machinery and mobile phones, and by online launching of fraud auctions have damaged more foreigners, who were caused a very large aggregate loss .

The organizer and coordinator of the organized criminal group, caught in criminal activity, according to the "skills" of each, and who were raising sums of money that came from fraudulent electronic auctions .

Criminal activity consists in no right access to computer systems, no right possession of computer data, and e-mail used for cheating victims.

Ultra Electronics values at $ 50 billion/ year the global market of cyber security. "And this market is growing 10% a year, twice faster than the entire information technology sector," says director Cassidian Cyber Security Solution within EADS (European Aeronautic Defence AND Space Company N.V). [3]

The Ministry of Communications and Information Technology made an eFraud Portal and is currently managed by the Department of cybercrime in the Ministry of Interior and Prosecutor's Office specialized section of the High Court of Cassation and Justice. Portal gives everyone the opportunity to appeal, on-line, any chance of possible fraud or other illegal activities on the Internet, www.efrauda.ro.

Mission to Romania of the United States Agency for International Development (USAID) in cooperation with the Ministry of Communications and Information Technology has initiated since 2002, the project RITI dot-Gov, and currently has developed a "Quick Reference Guide" for the enforcement of legal provisions relating to computer crime and providing assistance to law enforcement authorities, and for all those involved in crime prevention information, implemented in Romania by Iternews Network Inc.,an American non-profit organisation.

The latest survey conducted by Computer Crime Institute and Federal Bureau of Investigation (FBI) in 2003 indicated losses of $ 201,797,340 for 538 U.S. companies and institutions surveyed.

In Romania, was made in 2003 by specialized research computer 200 offenses half of which were fraudulent electronic auctions, 30% of goods ordered on-line fraud, 10% unauthorised access and 10% referring to other crimes (Nigerian letters, viruses, child pornography, use of false

identities), transmission of viruses, child pornography, use of false identities.[4]

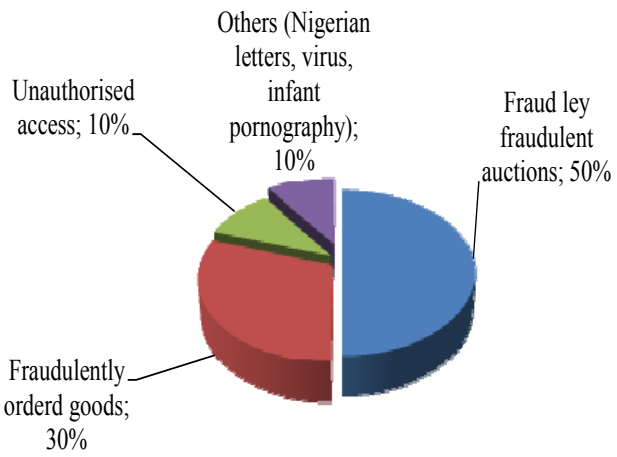

Fig.1. Computer fraud distribution. Source: data taken from www.securitatea-informatica.ro

As a result of this evolution in Romania have made a series of studies to identify the main causes generating:

- High performance technology used by criminals;
- Lack of preparation and training specialized personnel in the prosecution;
- Delayed reaction of the victims, this assuming minimal chances of identifying cybercrime and other measures to recover damages;
- The appearance of retention phenomenon , of the damaged, to report crimes to prosecutors;

EC (European Commission) Report supports the development by 2012, of the European emergency plan in case of cyber incidents, and supports national emergency plans and suggest the establishment of cyber attacks simulations, it also suggests the establishment of effective networks of intervention in emergency event of the global information. [5]

Recent events have shown that new and more sophisticated cyber attacks in terms of technology can disrupt or even destroy vital societal and economic functions. Examples include attacks on networks from the French Ministry before the G20 summit, the EU system of sale of emissions certificates and, recently, on the European Service for External Action and the Commission itself.

The Industry, Research and Energy Commission of the European Parliament, in the draft Report, "on critical information infrastructure, achievements and next steps: towards a global cyber security"-" All these developments in recent years unsolicited, which do not exhaust the efforts to enhance cyber security in the EU space, demonstrate that Internet security is a pertinent issue. It becomes obvious that the Internet is a critical infrastructure and the Internet disruption could lead to substantial losses and security risks, affecting a large number of European citizens and businesses. In addition, rapid technological developments require that prevent Internet attacks, repair and resistance reactions global network to be based on a comprehensive framework based on the flexible, innovative and long-term reaction. This framework should ensure effective interaction between governments, businesses, individuals and all other interested parties. Lastly but not least important, the increased strength of the Internet is possible only when there is an international system of cooperation and international rules." [6]

Border cybercrime materialized by cyber attacks causes great damage and it is possible a scenario in which a perpetrator able to launch attacks can interrupt the communication range. If only a small

percentage of important information hub sites are compromised, the chain reaction is likely to fragment the Internet. On the other hand, critical networks, such as those which are banking, have special equipment and protocols and are not connected to the Internet, so the only way to be compromised is the inside way. [7]

Moreover, in a modern society that the crisis becomes more acute as the use of the Internet to purchase goods, pay bills, or solve other everyday problems becomes a necessity, but such actions seriously jeopardize confidence people in the fairness of transactions online and generate a phobia in using electrical equipment. Another terminology used to define this type of crime is, cybercrime, which is crime performance using computer technology. Activities on economic and financial crime have exploded with unprecedented evolving practically together with the improvement in telecommunications and information technology assets, consisting of up almost means faster and "safer", of acceptance and transfer of money.

A computer operator can bring financial harm to a company by entering false data, the possibility of theft of money.

If one has to appreciate the extent of this type of crime, we can say that for them there is an impediment because computer technology is unevenly distributed in the world, with reference to the internet network and almost no existing means of protection .

But there are no borders for the "pirates ", they are forming organized criminal groups. Cybercrime is characterized mainly by committing crimes via computer, and they consist of:

- computers are used with predilection to commit financial crime, not necessarily as an instrument of crime, but chose to enter the database in order to extract and or enter information to initiate microchips for cell phones, to examine checks company, bonds and other negotiable instruments for counterfeiting [8]
- theft from distance of large sums of money from private accounts were the accounts of banks, commercial or material transfer tax;
- purchase of goods over the Internet that are paid illegal credit card numbers illegally using them without the owner's consent;
- offering for sale goods on the Internet at tempting prices, after opening a bank account and after deposit of amounts ,the account is abolished and goods are not dispatched
- entering the large database of companies, with the destruction or theft of database information;
- use computer for terrorism purposes;
- infringement of intellectual property;
- identity theft.

The context of the current crisis worsens the transformation of hacker activity in an industry organized crime, due to the exploit on personal information, which affects individual and socio-economic life.

IGPR (General Inspectorate of Romanian Police) and (BSA) Business Software Alliance signed a Protocol providing for prevention, and combating software piracy and the purpose is to establish cooperation between the two parties on the issues mentioned.

This Protocol, which is valid for two years, involves the initiation of projects, programs, campaigns and information activities, outreach to consumers and businesses, to reduce the risk of infringement. [9].

In 2011, specialized structures within IGPR, conducted over 180 inspections, claiming 46 cases to prosecute. [9].

IGPR, pursuant to provisions under Law no. 218/2002 on organization and functioning of the Romanian police with subsequent amendments and the Law. 8/1996, as amended, on copyright and related rights, carries out through its specialized structures, specific measures to:

- prevent and fight against financial and economic crime, in using information technology;
- declare offenses under the regime of copyright and related rights and conducting investigations on them.

For example, during 2011 as well, there were found 144 economic crimes, of which 40 were tax evasion and following the

checks carried out by the Fraud Investigation Department of the IGPR to protect business environment against illegal acts illegal affecting competition relationships and market economy mechanisms , were found 144 new crimes. [9]

Computer system, SIRENE (Supplementary Information Requested at the National Entries) appeared due to the need for data exchange and additional information from the Schengen and it is designated for storing information by the Romanian authorities and the European common data exchange, border monitoring, that is SIS, respectively (Schengen Information System).

With the lifting of internal border controls of Schengen Member states, there was a need in render Schengen Information System operationally, which contains minimum information about persons or property pursued by legal proceedings, and data and information obtained through the SIRENE Bureaux are established in each Schengen member state. [10]

In our country, to assess the exact scale of this type of crime, there were identified certain features, which as I described, go beyond borders and often, the perpetrators are Romanian citizens.

Among the features identified on cybercrime in Romania we can mention with the predilection for :

- financial condition (payment systems, credit and payment products);
- organization of criminal groups, with highly specialized young people;
- transnationality of these actions, using the other states systems;
- specialization, continuous improvement of operating systems;
- refocus criminal groups who commit computer fraud, on companies, thus committing large damaging , hard to recover due to operating modalities.

These "activities" aim at getting with predilection for financial products, credit, payment systems offered by financial and banking institutions, criminal networks that members of such fraudulent networks access, causing significant damage to both natural persons, companies, even affecting economies countries.

Cybercrime, aims however, at mostly economic and financial field, in order to obtain illicit revenues by some criminal groups. Economic and financial crime is more difficult to identify and proved, is much more complex than other traditional forms of crime, by the severity of damage caused , the number of people affected and by propagation in a long time as well.

"Cybercrime" is known as computer fraud, is part of computer-related crime leading technology, based on the rapid evolution of computer technology and communications, has created high possibility regarding criminal activity.

Cybercrime prevention is related to how information systems are protected, the use of permanent protection systems.

Also in this respect, the European Commission proposes a common policy to combat these types of fraud and developing a coherent European policy framework and public awareness about the costs and dangers of cybercrime.

Another approach to crime information, can be analyzed in terms of other definition, namely that "achieving direct or indirect, physical or logical, deliberate or non deliberate actions, aimed at changing one or more states of a system and subsystem information" legally and can be thus grouped :

- crimes for which committing were used information and computer technologies;
- crimes where computers and computer networks are targeted attack.

The need for international reporting mechanism for Internet Service Providers (ISP-Internet Service Provider) in respect of illegal material distributed via the Internet is reported more than 10 years.

Transition to the Information Society in Romania, as a relevant component development requires not only information and communication infrastructure, but also legislative gaps on computer activities, the

computerization process and, in general, the information technology.

Change, modernization and adaptation to new technologies represents our country's response to the conditions of acceptance of Romania into the European structures. [11]

Development of information technologies in recent decades due to the need for storage and rapid transmission of information with the least cost revolutionized global commerce, directly or retail trade, redefining the traditional principles of marketing, electronic commerce became synonymous with profit growth.

With this type of commerce, also occurred offenses on electronic commerce. The analysis on these offenses provided under the law no. 265/2002 referrers to crimes of:

- forgery of electronic payment instruments;
- possessing counterfeiting equipment specific to electronic payment instruments;
- conduct fraudulent financial transactions;
- acceptance of financial operations performed fraudulently.

For example, "no right to a network connection", addressing purely technical, simply connecting to Wi-Fi signal (radio waves) can be often automatically, depending on device settings used by the alleged offender without his knowledge, case that his guilt cannot be established.

Analyzing the many technical means that allow "access", meaning, "the program, run a program to intercept, to establish, communicate, store / archive / store or retrieve data from any other use of a sources provided by computers, including data or computer programs, computer systems, computer networks or databases" [12]

With the use of virtual worlds, criminals have found an ideal means to "launder money" that hide the origin and actual possession of their income from illegal activities.

Economy of virtual worlds like SecondLife has no fiscal rules in the real economy. Freedom of action is virtually unlimited and documents formalism nonexistent, guarantees on transactions are secured by the same methods that e-commerce already uses in the real world.

Law speaks in all criminal modes of money laundering, on goods in the ordinary meaning, they are physical, material, having a perceptible physical existence. To these it might be added intangible goods traded, and some virtual goods, which by their features already mentioned, although sensory perceptible, they do not have a physical existence, but, by real-virtual correspondence, the tangible features with physical existence .

Furthermore, by assigning a value, as are the spellings, these assets become property values, some made extremely important by design. [13]

As in case of theft of goods in cyberspace, the law does not provide conformance to features, nor associates the economic value to this new categories of goods .

Thus, it remains an area where the legislator should intervene and adapt the legal text, making it applicable to virtual space as well, by the simple and extensive definition of goods or indicating the extension of regulation on illicit activities in cyberspace.

Internet can be also used as a " comfortable umbrella " for transactions outside virtual worlds, for automating transfer of money, without possibility on the phenomenon to be monitored.

Within the national regulatory space, notions of "money laundering" by means of computer or "cyber terrorism", do not currently have an impact, the subject being studied and treated accordingly, losing itself in general usual concepts, giving substance to field cybercrime through legal practice.

So, by definition, cybercrimes (cybercrime), are all crimes committed by electronic means and investigating these types of crimes involves certain challenges:

- the opportunity to acquire a transnational or organized character and the effects arising for prosecution;
- virtual place of the offense;
- data theft;

- difficulties with intercepting communications (encryption communications interception servers);
- management of electronic evidence;
- organs of the criminal investigation are limited to legislation and national jurisdiction;

These issues generate new management tools, leading to fighting cybercrime:

- establishment of preventive measures current threats (SIRENE application);
- liability to be established since the early space;
- harmonization of national legislation;
- agreements with Internet service providers, financial institutions, credit, etc.
- specialized professional staff, with the evolution of new information technologies;
- use of international instruments of cooperation in the field (SIRENE, SIS).

As electronic media become more accessible to all, cybercrime is diversifying and intensive, far outstripping the traditional fraud and forgery.

In this context, a European strategy is not simply a desire, but an important necessity.

Thus, the Prevention and Combating Crime Service, in its Progress Report for 2011 states that, in 2011 increasing trend of crimes against systems / computer data or committed using information technology, is fully revealed by statistical indicators pursued , represented as the number of cases recorded, as well as the number of cases solved or remaining in work progress.

It is also noted that, besides false informatics (work of "phishing") and the actual computer fraud, deception committed through information technology represented by fictitious goods auction sites specialized in electronic commerce is a big part of notified facts, which are the subject of cases registered or resolved during 2011.

Undermining access accounts belonging to users of e-commerce sites, financial institutions or social networks, unauthorized access to systems followed by blackmail or misuse of confidential data obtained (electronic payment tools data) represents ,with credit card fraud (compromised ATM), capturing information on the magnetic strips of credit cards, forgery of electronic payment instruments), manifestation forms of cybercrime in obvious increase.

Transnational character, whether it is given by where the facts are committed , or it is about the location of victims represents an objective factor to increase in cybercrime, with recruitment by means of increasingly sophisticated "arrows", and their specialization as needed (opening bank accounts, transport of money or creating money laundering networks comprising Romanian and foreign citizens, as well.

The transboundary nature of computer crime, the specialized networks located abroad, respectively, still remains one of the difficult problems to overcome in operational solution on the files recorded .

Not least ,the effective investment in creating/ purchase of criminal schemes, investment in technology and development of counterintelligence activities, determines that the groups investigated present an increasingly higher danger.

Specific analysis of cases solved, for 2011 also confirms the trend observed in recent years on the migration of people from organized crime to ordinary computer crime, especially in the fraudulent operations with electronic payment instruments.

Such facts remains a significant source of revenue for prominent members, without the latter to become directly involved in crime enforcement.

” Regarding credit card fraud, counties Prahova, Vâlcea, Teleorman, which are known as the manifestation areas of the phenomenon of computer fraud.

Counties of Vâlcea, Arges, Constanta, Teleorman as well the City of Bucharest keep being areas with high criminal potential for computer crime, being equally noted facts both in the field of credit card fraud, and computer crimes area in ”stricto sensu”, as well .

In 2011, 873 cases were resolved, an increase of 62.87% compared to 2010, when 536 cases were solved".(14).
Cybercrime is an ascendant "process", involving numerous domestic and international organizations in this respect, all countries are "united front" against this scourge.
"Models of computer attacks:
(i) Access user-attack in this class requires access to system by users having certain privileges, as the following steps:
- obtaining information- a complete search is effected ,in order to get data to identify security vulnerabilities. This process is often like being authorized by instruments such as "nmap" or instruments or tools that target a specialized application such as a Web server. Vulnerabilities can exist in the system components , may result from errors on system administration ,or can be reflected in poor security policies of the system.
- operation- of a security breach is done to gain access or obtain system information for their use in future cues. In the early stages of an attack vulnerability may come from information such as computer name or names of user accounts
- damages -the desired effects as a result of an attack are ,for example, changing data, access to classified information or establishing permanent connections that can be used for ongoing access. The final step in this attack is changing logos files, so the attack cannot be not identified
ii) Access to components- an attack of this category does not require user access to the system. These attacks create denial of services by sending improper requests. In some instances, such a request may result in loss of certain weaker system components . In other cases, the extra time needed for processing such a request is sufficient to slow processing much down . Steps in this attack are:
- obtain information – is identified a component of the system and a communication port
- operation- messages are sent to the port;
- damage - loss or overloading a component application or network service;
(iii) The application content- these types of attacks send improper data application rather than the network components. In this situation traffic is properly formatted. The problem is related to the content of traffic and as in the case of attacks on access to network , these examples do not require an attacker to gain access to the user. The steps are as follows:
- Obtaining information – is done the identification on the target application. This can be either a network application such as a Web server or a browser or an application such as Microsoft Office, where emails are used to transmit data to applicants;
- Exploitation- the content is directly or indirectly sent to the target application;
- Damages – following such attacks user files are deleted, configuration changes a user's account or user files are exported. [15]

Conclusions

Thus, UNAFEI, ENISA, DIICOT, EA DS, FBI, USAID, SIRENE, jointly elaborated projects, guidelines for application of the law on computer crime, and providing assistance to law enforcement authorities and to all those involved in crime prevention information.
UE also, supports in 2012, an European emergency plan in case of cyber attacks and make recommendations for the establishment of effective networks for urgent intervention in the event of information worldwide.
IGPR and BSA, signed a protocol providing for the prevention and combating software piracy.
Dimensions of cybercrime, was exemplified by specific data provided by authorized institutions (DIICOT, IGPR), the transboundary nature may also be highlighted by the interest of many countries, for this, and targeting the economic and financial with predilection for
.

Thus, SIRENE, appeared due to the need for data exchange and additional information from the Schengen area and it is designated for storing information by the Romanian authorities and the European common data exchange, border monitoring, SIS ,respectively.

Several definitions have clarified the notion of computer fraud, which means because of the state known beyond the frontiers of the country, manifested globally, which required the need for a mechanism for "reporting" international internet service providers (ISP-Internet Service Provider).

Computer fraud is not just stealing goods from virtual space, criminals have found an ideal means to "launder money" and hide the origin and actual possession of their income from illegal activities.

One can thus conclude that any or all means would set, as the means information is developed, there is virtually impossible to have a "percet sync" in terms of prevention.

Practically taking advantage of this situation, computer fraud is growing and diversifying extensively, and establishing comprehensive strategies on international instruments of cooperation, (SIRENE, SIS), becomes necessary.

For examples, we can say that computer fraud in Romania concerns with predilection for financial institutions or foreign citizens, thus reducing confidence in financial systems on the network security.

Expansion of information technologies, in addition to positive effects on socio-economic and political life of the world, generated, as mentioned, and some behaviors outside the law, taking forms that did not exist previously. But in a society that supports economic and social repercussions daily cybercrime, daily use is made of computers in all areas of socio-economic life.

Computer fraud can have a very high price in economic terms, for all the world, everything depends on their extent, and the problem of defining computer crime may appear as a new concern for societies where access to new technology was delayed and where some states precautionary measures have not yet established.

References

[1] www.criminalitatea-informatica.ro
[2] *Data delivered by Service for Combating Organized Crime Vâlcea-Indictment*, 2011.
[3] www.europarl.eu/meetdocs/2009-2014
[4] www.securitatea-informatica.ro
[5] bursa.ro
[6] V. Patriciu .a, *Internet and Law*, Editura ALL BECK, Bucure ti, 1999, p 21.
[7] I. Vasiu, *Cybercrime*, Nemira, Bucure ti, 1988, p 121-122
[8] *Bulletin of Information and Documentation MI*, anul, XII, nr. 6 (53), 2002.pg 131-133 (Bulletin of information)
[9] EuroAvocat.ro/20/04/2011
[10] Directive 95/46/CE
[11] *Reason presentation for Bill NO. 365/2002, as to e-commerce.*
[12] T. Amza .a., *Cybercrime*, Ed. Lumina Lex, Bucure ti, 2003, p.14.
[13] https//marketplace.secondlife.com/p/
[14] www.diicot.ro/pdf - *Directorate for Organised Crime Crime and Terrorism* / no 4/CD/2012-Prelucrare, pg 39-45.
[15] http://www.ciminalitatea-informatică.ro

Raluca Georgiana POPA is a PhD student in the field of Economy and International Affairs at the Academy of Economic Studies. She graduated the Faculty of Law within the University of Bucharest and she has a master in economy. Currently, she works within the Management Authority of the General Programme 'Solidarity and Management of Migration Flows', programme which sets out to support the common policy on the management of the external borders of the European Union and to help implement the common policies on asylum and integration.

Perspectives on the Role of Business Rules in Database Design

Anca Ioana ANDREESCU, Marinela MIRCEA
Economic Informatics and Cybernetics Department, Academy of Economic Studies, Bucharest, ROMANIA
anca.andreescu@ase.ro, mmircea@ase.ro

Business rules are at the foundation of every information system as they drive and offer guidelines for managing and conducting all activities within an organizations. They are important both for operational systems and for analytical ones, as they can serve as data quality rules. Several business rules implementation solutions are presented and analyzed, according to some basic criteria that have to be taken into account when choosing a business rules implementation strategy. This paper also emphasizes how an explicit manipulation of business rules influence the database design.
Keywords: *database design, business rules, implementation strategies*

1 Introduction

Business rules approaches in software development are concentrated in finding ways and facilities that would support automatic propagation of business changes from business environment to software systems. This would help to bridge the gap between business and technology, as aligning information systems and business operation is one of the fundamental problems in all organizations [1]. To reach that goal, it has to be clear how business rules should be dealt with, as a special kind of software requirements, in each phase of the software development lifecycle. Accordingly, at least the following aspects have to be considered in this process: rules identification, business rules specification, rules implementation and rules management. A business rules implementation strategy assumes identifying both the place and the way of how to implement a business rule. The next sections of the paper will address several issues and challenges related to business rules, aiming to act as guidelines during business rules implementation. Particular emphasis will be given on the rules that govern the database model of an information system and various guidelines will be addressed in this regard.

2 Business rules overview

Over time, numerous studies and research projects aimed at the discovery, analysis, modeling, classification, formalizing and documenting business rules. It must be noted that in all researches, reference was made (explicit or not), on how the scientific approach relates to dimensions from the Zachman framework [2]. From this point of view, we have identified two research groups:

- which are based on a business vision, aiming at developing a business model, therefore on the activities conducted within the organization; remarkable here is the study Business Rules Team.
- which are based on a system vision, and therefore aim at modeling the information system; representative examples of such projects are : ESPRIT-I FIELD GUIDE Temporal or ESPRIT-II.

As part of a consortium of 17 organizations, known as "The Business Rules Team", Business Rules Group (BRG) participated in the proposal from the Object Management Group (OMG) to analyze the semantics of business rules from a business perspective. The answer of BRT, called Semantics of Business Vocabulary and Business Rules - SBVR, is a comprehensive study on the semantics of business vocabulary and business rules, which have considered the following aspects [3]:

- The focus on business, in that there's no reference to information systems or concepts of information systems - focuses exclusively on how business people think.
- Business rules are fully specified from business vocabulary.
- Semantics integrity rules should be based on predicate logic.
- People involved in the business (as "customers") communicate vocabulary and business rules to computer professionals (as "suppliers"), in the form of system requirements specifications.

Regardless the perspectives from which they are being analyzed, business rules always belong to an organization and must occur in such a way that is acceptable to the business [4].

Graham Witt [5] distinguishes two main types of rules: a) operative rules, which describes behavior and state what must or must not happen in particular circumstance. They include the so-called "if-then-else" rules or event-action rules; and b) definitional rules which constrain the definition of various constructs created by the organization. These may be simple business facts, derivation rules, complex decision table rules etc. Additional, the author mentions several ways in which rules may be applied in an organization:

- rules governing the business processes performed by the organization through their employees, customers, suppliers and partners;
- rules governing user interfaces and electronic messages;
- rules ensuring database integrity;
- rules governing human behavior other that business processes.

Because some of the business processes may be automated or not (or partially automated) we can conclude that rules related to user interfaces, electronic messages and databases are likely to be fully implemented in the information system.

3 Business rules implementation solutions

Last decade was marked by the development of a wide variety of technologies, instruments and software products that support the implementation and maintenance of business rules. This section presents the main conclusions of an ample study [6] regarding a large set of possibilities to translate business rules from the business domain to the technology domain. The considered solutions will be gradually presented, starting from those who offer a minimum support for business rules management and getting to those who offer the premises for an efficient rules management. Likewise, a wide area of technical solutions was covered.

Probably the most common way to implement a business rule is to mingle it inside de application source code. In this case, it is advisable to use some style conventions in order to mark the source code parts that implements business rules [7]. Subprograms and program statements separated by user defined limits are a first step towards emphasizing and encapsulating business rules.

The specific constructors of the Design by Contract (DbC) method (pre-conditions, post-conditions and invariants), advocated by Berthrand Meyer in [8], are suitable for implementing business rules, allowing rules identification in the source code and increasing software reliability. Starting from this remark, in [9] and [10] we have proposed and illustrated various ways of DbC-like rules specification and implementation using: a) formal languages (Object Constraint Language), b) programming languages that supports DbC (Eiffel, XC#) and c) a business rules extension of the „Contract" design pattern [11].

Aspect oriented programming intends to complete object oriented programming (or other paradigms), by extending the possibility to modularize applications. As

aspect allows separating and reusing certain software requirements, they are a good candidate for implementing business rules, as stated and illustrated in [12].

Another step forward in rules modularization is the separation of rules in files that contains source code written in a script language, like VB Script. Choosing this implementation solution will have to take into account the low application performance, due to the fact that, for interpreted languages the code runs rather slow.

The existence of software components as stand-alone entities represents a good premise for encapsulating decisions based on business rules into such components. Their main advantage is due to the possibility to modify a rule implementation without affecting the base applications that reference the component. Rule components are seldom provided from external sources, because they implement a part of the application business logic, and therefore, are not very general.

Business rule markup languages play an important role within the Web environment, as they allow the specification of business rules as modular and independent units, in a declarative manner, as well as rules publication and interchange between different systems or tools. The proposal for the standardization of the RuleML [13] language proves an intense work in amending rule markup languages, despite their lack of usage in practice.

Business rules engines are a software technology that guerdons the efforts to identify efficient business rules implementation solutions, as they were designed with the exclusive scope of rules administration. As opposed to business rules components, a rule engine do not solve a particular problem, but provides a set o generic capabilities to define, store and apply business rules. But, above all things, a rule engine must manage the dependences between rules, using inference mechanisms. It can be used as a standalone toll, without storing business rules in a dedicated repository (like Jess or Drools), or it can be part of a special type of software system, named Business Rules Management System (like IBM JRules).

Very often business rules are associated with the control of business behavior. Research and practical solutions in the field of business process management and automation, have dealt with the problem of business rules implementation. Therefore, most of the process engines are able to reference other components that implement rules, while the rules that are internally implemented within a process engine will be most probably represented in a proprietary format, as part of the product that automates the business processes (for example, Microsoft Windows Workflow Foundation [14].

Rules that influence or constraint business objects will be, most probably, included in the system's database, from where they have the fastest access to data. Database mechanisms are able to implement business rules as constraints, stored procedures and triggers.

Table 1 depicts a comparative analysis of the implementations solutions presented above, by emphasizing their main advantages and disadvantages with respect to the following criteria: (S) rules are separated by other implementation aspects, but they are still placed in the same file as the rest of the code. By dignifying rules position in the source code, they can be easily located. (E) rules are externalized and separated from the system's business logic. (L) rules are expressed in a high level specification language, suitable for the non-business people. (R) relations among rules can be determined, similar to inference mechanisms. (D) the solution is dependent of a software producer.

Table 1. Comparative analysis of the business rules implementation solutions

SOLUTION	**CRITERIA**					**ADVANTAGES**	**DISAVDANTAGES**
	S	E	L	R	D		
Subprograms and program statements	x					• ease of integration in code • high execution performances	• a rule change implies base code modification
Design by Contract constructions	x					• rules are separated by subprograms • testing facilities	• a rule modification implies recompiling the base application
Aspect Oriented Programming		x				• rules are placed in files external to the base code	• debugging problems
Script languages		x				• rules are placed in files external to the base code	• low execution performance
Rule components		x				• a rule modification does not imply recompiling the base application	• low generality, due to the method parameters from the interface
Business Rules engines		x		x	x	• rule reuse in multiple system points • allows the management of complex set of rules	• rules tend to concentrate in one place
Business Rule markup languages		x				• rules can be interchanged • allows platform interoperability	• few tool support and practical applications
Database mechanisms		x				• high application performance (Constraints and Store procedures) • insure consistence along all system modules	• low application performance (Triggers)
Business process engines		x		x	x	• separate business rules from business processes	• rule representation in a proprietary format

SOLUTION	CRITERIA					ADVANTAGES	DISAVDANTAGES
Business rules management systems		x	x	x	x	• increased rule management facilities • allow business people to define and modify rules	• significant financial investment • implies additional security issues

It is highly probably that, for a certain software system, a business rules implementation strategy will encompass a combination of different approaches discussed in this section.

The decision to choose one implementation technology is influenced by many factors among which the most important is the type of system which is being developed. Through a simple analysis, we have identified four main ways to implement rules, as described in Figure 1. Assuming that every software systems stores the information about business objects within a database, the hypothesis stating that some of the business rules will be implemented at the database level is plausible, mostly due to performance reasons, but also to force all the system's applications which use the database, to follow the same business rules. Starting from the above observations, in Figure 1, is presented a pattern that establish the link between the type of software system and the ways to implement rules.

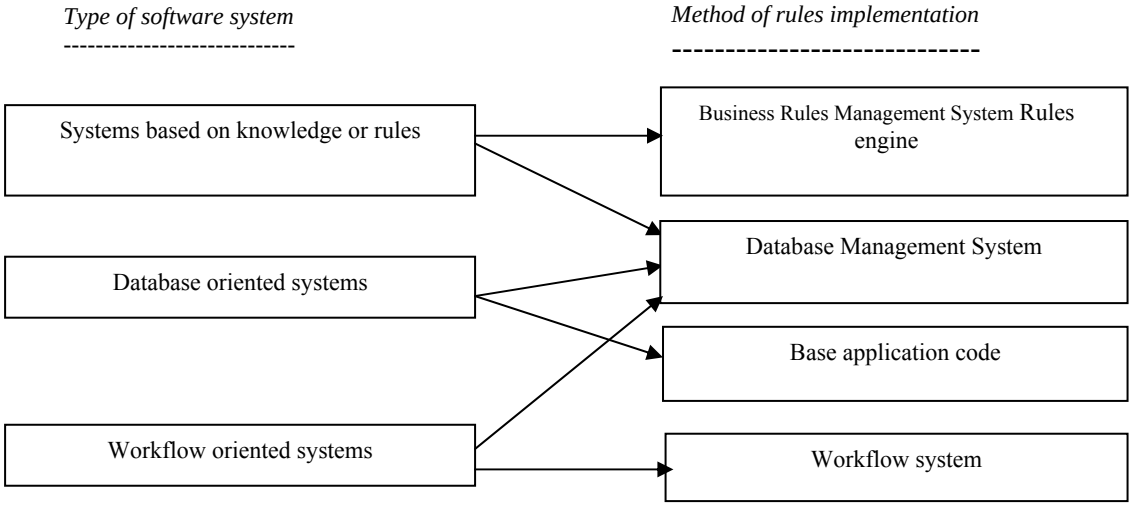

Fig.1. Correspondence between the type of software system and the business rules implementation methods

The great variety of implementation solutions of business rules leads to the necessity for a well founded and competent analysis of the system's needs. Therefore, at least three software quality characteristics must be explored: efficiency, maintainability and reliability. Efficiency will be evaluated with respect to the final user who demands a quick answer from the system in such a way that the business rules must be verified and executed efficiently. Maintainability resides in the degree of easiness necessary to update the application in order to continue using it even under modified conditions. From the business perspective, the system must provide facilities for: business rules specification, adding new rules, deleting those who cease to operate, modifying existing ones or restructuring relationships

between rules. Reliability has to be analyzed especially from the perspective of correctness, meaning the degree in which the system fulfills intended functions and imposes the business rules required by the beneficiary.

4 Business rules in databases

As mentioned before, besides rules that affect business processes, there are also rules that apply to the data. Rules that affect data will fit, most likely, in the database system, a place where they will be best positioned for access to data. Database technology that is currently most commonly used is the relational model, represented by products such as Oracle or Microsoft SQL Server, so we build the discussion around them. Relational Database Management Systems (RDMS) as the one referred to above, do more than store and collect information: it offers mechanisms capable of controlling the characteristics of application data [15].

In the following, we analyze three types of database features that can be used to implement rules: constraints, stored procedures and triggers.

A. Constraint is a generic term related to table columns which limits data values in order to preserve database integrity. Integrity of the relational model restrictions are structural and behavioral [16]. **Structural constraints** include:

1) Unique key restriction: a table should not have multiple tuples with the same value for all keys. This restriction is provided by a RDMS by not allowing duplicate values for a primary key field.

2) Entity restriction: for a table, the primary key attributes must not take NULL values. This restriction is provided by a RDMS also by not allowing null values for a primary key field.

3) Referential restriction: in a table t1that reference a table t2, foreign key values must include the primary key values of t2 or the value NULL (unspecified);

Database design should be influenced on the meaning that business people and business analysts give to certain objects or terms from the business vocabulary. And discussions regarding these aspects might be endless. If we consider a very simple e-commerce application, we must define who a client is: a) the one who visit the web site; b) the one who places an order not being authenticated; c) the one who places an order being authenticated; d) or the one who finalize an order being authenticated. Depending on these different interpretations, the database design could slightly differ. For the more restricted scenario of case c), in business rules modeling, such a situation is expressed in the form of a simple fact: A Client can place one or more Orders. One way to enforce this when entering a new order in the database is to specify the NOT NULL property for Client column. This means we have provided a value for the client to create an adequate record. An alternative to referential integrity is to allow NULL values for reference (case b) from the above example). This would allow an object Order to be created and initially disassociated and an association to be added later. Since both scenarios are plausible, their associated business rules will determine which of them is correct in a particular case.

Behavioural constraints are those which are defined by the behaviour of data and take account of existing values in relational database. By restricting the field, it can be checked whether the corresponding attribute in a table is between certain values or whether it conforms to a prescribed format. The relationship between columns of the same table can also be. Behaviour constraints being very general are managed either when data description (e.g. CHECK clause) or outside the model at runtime. We present two examples of constraints that apply to Customer and Order tables, using SQL Server specific syntax.

Rule 01: A customer identification code is defined as a four-digit integer.

```
ALTER TABLE Client
ADD CONSTRAINT BC_R01_ClientFormat
    CHECK
    (IDClient LIKE ' [0-9] [0-9] [0-9] [0-9]')
```

Rule 02: An order is, by definition, in one of the following states: Pending, Approved, Finalized and Delivered.

```
ALTER TABLE Order
ADD CONSTRAINT BC_R22_OrderState
    CHECK
(state = 'Pending' OR state = 'Approved' OR state = 'Finalized' OR state = 'Delivered')
```

Incorporation of control mechanisms in a database prevents the occurrence of any actions that contravene the rules, not only when the object is created, but also for any future attempts to change it. The major disadvantage of these constraints is that they simply can't be used for more complex conditions, involving several objects, resulting in association of multiple tables in a relational database.

As with other solutions that implement rules, it should not be forgotten that we will need to find a way to determine where the rule has been codified. For example, you can adopt a naming convention rules, as in the examples above, which can be used to track or to extract rules from the system [7].

B. As their name suggests, **stored procedures** are procedural code modules compiled and stored in the database (as any object manipulated by a RDMS) without having to be screened a second time to execute [17]. Compared to any other source code, it has some advantages. First, because they are so close to the data, can be more efficient to implement data-centric rules. Also, for distributed systems where there are several applications that share the same database, stored procedures can be a guarantee that the rules contained therein are used consistently within each application. The same is true for newly created applications that will not have to re-implement the rules that apply to the entire system.

C. A **trigger** is a special type of stored procedure that is not invoked explicitly, but is triggered automatically when a certain condition is detected. Triggers are attached to an event within a database table. The event can be an action to add, modify or delete entries in the table or a combination of these actions using logical OR operator.

Although is triggered by an event occurring in a table, a trigger is not limited to the characteristics of that table. Unlike simple constraints discussed above, a trigger can be used to implement constraints in a table, between the tables, between databases and even between servers. However, these advanced capabilities require taking some compromises. Triggers are slower than other approaches, such as simple constraints. It must also be taken into account the danger that inexperienced users can do when using these facilities ineffective, because overall system operation can have strong negative effects. Therefore, some authors even recommend avoiding the use of triggers, at least when we are not sure of the effects they can generate.

5 Conclusions

The construction of a rules repository and the specification of business rules in a structured and/or formalized manner constitute important steps towards raising the quality of a software system that treats rules as explicit requirements.

As guideline for developing a database model it is very important to use a standardize vocabulary because business stakeholders will comprehend the meaning of a rule more precisely if those specifications use business terminology rather than the names of database tables or columns. Experience shows that, despite the fact that all concepts seems to save a well-defined meaning inside the organization, many business terms turn out to have several meaning depending on the context and the person who uses them. In this regard, we

have presented in [18] a set of business rules patterns for rules specification.
Concerning efficiency and maintainability, these must not be analyzed separately, as between they exist certain interdependent relationships. The end result is that the evaluation of implementation strategies must balance the efficiency and maintainability requirements of the system.

Acknowledgements
This work was supported by CNCSIS-UEFISCSU, project PN II-RU (PD), "Modern Approaches in Business Intelligence Systems Development for Service-Oriented Organizations Management", code 654/2010, contract no. 12/03.08.2010.

References

[1] M. Bajek, M. Krisper, Issues and Challenges in Business Rule-Based Information Systems Development, Conference on Information Systems, 2005.
[2] J.A. Zachman, A framework for Information Systems Architecture, IBM Systems Journal, Vol. 26, No. 3, 1987.
[3] Semantics of Business Vocabulary and Business Rules, available at: http://www.omg.org/docs/dtc/06-03-02.pdf
[4] B. Von Halle, Business Rules Applied, John Wiley & Sons, 2002.
[5] G. Witt, Writing Effective Business Rules, Morgan Kaufmann, 2012.
[6] A. Andreescu, A. Uta, R. Mihalca, Business Rules Implementation Strategies, The Ninth International Conference on Informatics in Economy, Bucharest, ASE Publishing House, INFOREC Publishing House, 2009.
[7] T. Morgan, Business Rules and Information Systems: Aligning IT with Business Goals, Addison Wesley, 2002.
[8] B. Meyer, Object-Oriented Software Construction, Second Edition, Prentice Hall Professional Technical Reference, Santa Barbara,1997.
[9] A. Andronescu, How to Build Reliable Business Applications with Design by Contract, Seventh International Conference on Informatics in Economy, ASE Publishing House, INFOREC Publishing House, Bucharest, 2005.
[10] A. Andreescu, Dezvoltarea sistemelor software pentru managementul afacerilor, ASE Printing house, 2010.
[11] M. De Champlain, The Contract Pattern, PLoP '97 Conference, Monticello, Illinois, SUA, September 1997.
[12] M. A. Cibrán, Connecting High-Level Business Rules with Object-Oriented Applications: An approach using Aspect-Oriented Programming and Model-Driven Engineering, PhD Thesis, Vrije University, Brussel, June 2007.
[13] DATA H. Boley, The RuleML family of web rule languages, the fourth workshop "Principles and practice of semantic web reasoning", Budva, Montenegru, June 2006 .
[14] D. Chappell, Introducing Windows Workflow Foundation, Microsoft Corporation, 2007, available at: www.davidchappell.com/IntroducingWCFv1.2.1.pdf.
[15] C. Strîmbei, Modalități de organizare a datelor în sistemele informaționale economice, PhD Thesis, "Alexandru Ioan Cuza" University, Iaşi, 2005.
[16] I. Lungu, Baze de date oracle limbajul SQL, ASE Printing house, Bucureşti, 2005.
[17] I. Popescu, Prelucrarea avansată a informației.Oracle 8, Tehnică Pronting House, Bucureşti, 1999.
[18] A. Andreescu, M. Mircea, Managing Knowledge as Business Rules, Informatica Economica Journal, nr.4/2009.

Anca Ioana ANDREESCU is university lecturer in Economic Informatics and Cybernetics Department, Academy of Economic Studies of Bucharest. She published over 15 articles in journals and magazines in computer science, informatics and business management fields, over 20 papers presented at national and international conferences, symposiums and workshops and she was member in over nine research projects. In January 2009, she finished the doctoral stage, the title of her PhD thesis being: The Development of Software Systems for Business Management. Her interest domains related to computer science are: business rules approaches, business analytics, requirements engineering and software development methodologies.

Marinela MIRCEA received her degree on Informatics in Economy from the Academy of Economic Studies, Bucharest in 2003 and his doctoral degree in economics in 2009. Since 2003 she is teaching in Academy of Economic Studies from Bucharest, at Informatics in Economy Department. Her work focuses on the programming, information system, business management and Business Intelligence. She published over 25 articles in journals and magazines in computer science, informatics and business management fields, over 20 papers presented at national and international conferences, symposiums and workshops and she was member over 15 research projects. She is the author of one book and she is coauthor of four books. In February 2009, she finished the doctoral stage, and her PhD thesis has the title Business management in digital economy.