USING BUSINESS RULES IN BUSINESS INTELLIGENCE

Marinela MIRCEA[1]
PhD, University Assistant, Department of Computer Science in Economics, University of Economics, Bucharest, Romania

E-mail: mmircea@ase.ro

Anca ANDREESCU[2]
PhD, University Lecturer, Department of Computer Science in Economics, University of Economics, Bucharest, Romania

E-mail: anca.andreescu@ase.ro

Abstract: *Global recession brings new problems in nowadays business environment. In this period of economic recession, every organization must consider not only to survive, but also to improve its activity. Business Intelligence (BI) is one of the instruments that offer support in getting beyond crisis. If properly developed and implemented, BI can lead to improvements in decision making and to operational efficiency. This article is focused on the implementation of business rules, as an essential part in the development of BI systems, proper for the actual business climate and its underlying fluctuations.*

Key words: *business intelligence; business rules; metrics; patterns; public acquisitions*

1. Business Rules and Business Intelligence

Global recession brings new problems in nowadays business environment and many organizations must reduce costs and reorganize their business models in order to remain on the market. Business Intelligence (BI) instruments may provide organizations necessary information not only to survive recession, but also to achieve a good market position, thus being a real business differentiator. Likewise, BI systems may provide means to increase business value, answers to key questions, information about potential operational risks and financial metrics. If well implemented, BI systems can lead to improvements in decision making and to operational efficiency.

Business rules represent and essential part of any BI project. They allow automatic data interpretation, definition of performance key indicators, through important redline values and provide solutions for problem solving. BI experts use the term „business rule" in a variety of meanings and contexts. The definitions of this concept may be exclusively focused on a business perspective or an Information Technology (IT) perspective. Ronald G Ross offers a description of business rules, covering both views. From the business perspective,

„Business rules are literally the encoded knowledge of your business practices", and from an IT point of view *„ A business rule is an atomic piece of reusable business logic"* (Ross, 2003).

From an IT perspective, business rules are often encoded in either the ETL (Extract, Transform and Load) processes of a data warehouse or within BI tools during the design of specific reports or in analysis stages of business processes, situations that do not allow business users to remodel the implementation. A more efficient way to implement business logic is to independently describe business rules in a separate module. This software component is dedicated only to the implementation of business logic and offers four big advantages (Blasum, 2007):

- It is well designed and the business logic module is transparent to business users;
- It allows adaptation of business rules to frequent changes;
- It reduces duplicates, meaning that if the IT department decides to change an ETL or BI instrument, business rules implementation will not change;
- It allows inter-functionality, large scale IT usability and business rules management.

Business rules identification and analysis can be seen as a distinctive stage in the development process of a BI project (Figure 1). In order to integrate business rules into BI projects, the following aspects must be considered (Debevoise, 2007): ❶ identify rules that affect the metrics you are modeling; ❷ analyze the rule for redline values and incorporate these as dimensions in the data warehouse; ❸ extract numerical computations and add these as facts in the data warehouse. The development of a BI systems using the above guidelines will allow managers to evaluate the effects due to changes in redline values and calculations. All rules being organized in a centralized manner, it will be easier to locate a rule and to use it for decisions evaluation.

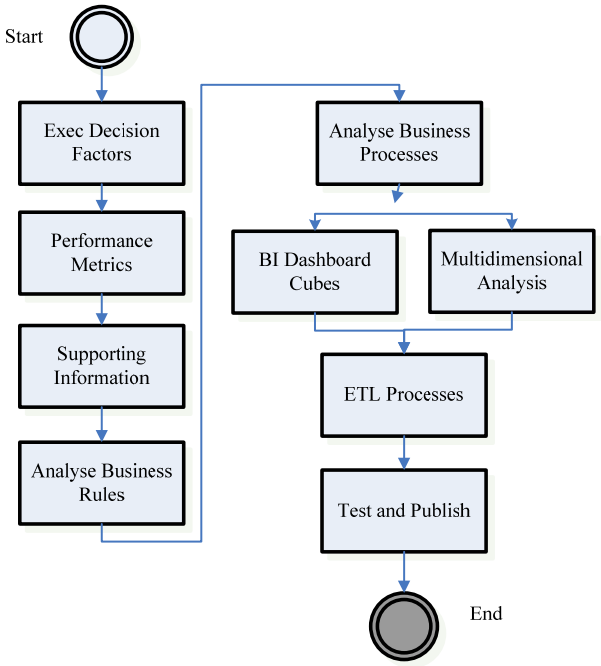

Figure 1. The basic process for developing BI for the project (Debevoise, 2007)

In order to be completely specified, each business rule must (Debevoise, 2007):

- *classify* a type, division or sort (for example, clients may be classified as preferred,

ordinary or not preferred, requests can be classified as approved, rejected, in progress or finalized); ● *calculate* formulas, to query data and statistics, to transform and associate values (constraints are often numerical, for example requests must not exceed a client's available budget); ● *compare* the calculation to the redline (redline is the key value that must be met or not exceeded or situated within a certain interval); ● *control* what is true or valid, correct or mistaken, and the messages that go with them. A business process for creating business rules consists of the following activities (figure 2):

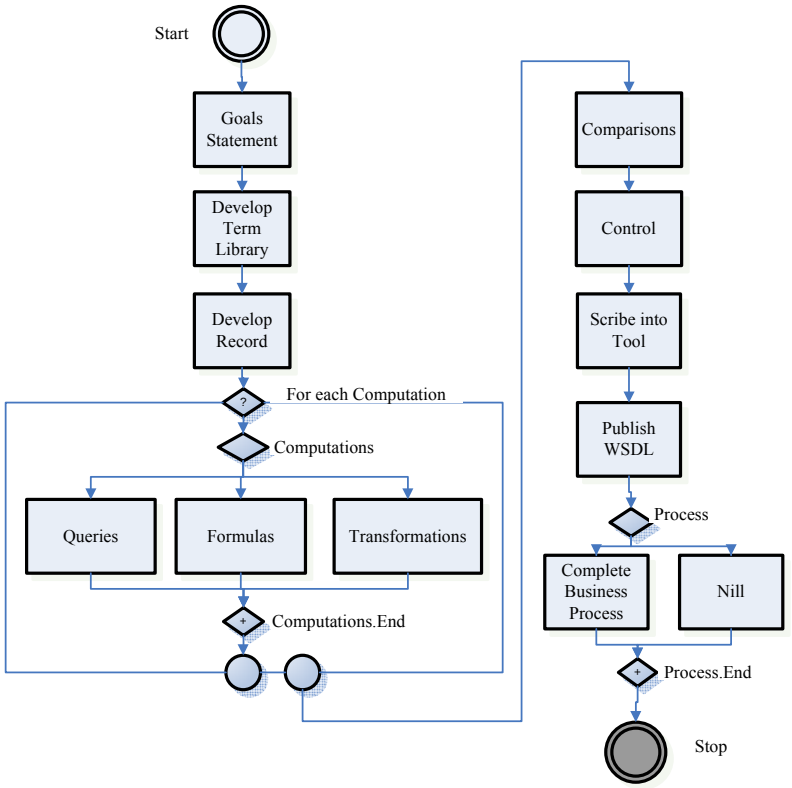

Figure 2. A process for developing business rules (Debevoise, 2007)

Creating a BI environment involves building an analytical data warehouse for managers. In many institutions, the most important decision metrics are calculated based on information obtained from various systems. For this reason, Business Process Modeling (BMP) is an important technique for collecting this information, and, together with the data warehouse, represents a method for integrating different sources of information. The analysis of processes and business rules (BR) offers support for the business analysis needed to create a BI solution, BR helping in defining the dimensions and metrics. One of the key factors of successful development is the use of BPM analysis and BPM/BR for improving the data warehouse schema.

The combination between business rules and web services offers an adequate approach for applications integration and sharing of distributed information. Business rules adoption, together with a service-oriented architecture, allows the integration of strategic corporate applications between multiple business units. For example, the same business logic that has been explicitly defined in a Business Rules Management System (BRMS), may be shared in a Service-Oriented Architecture (SOA) with other applications that need it. These applications communicate via XML with the Business Rules Services (figure 3) (Holden, 2007).

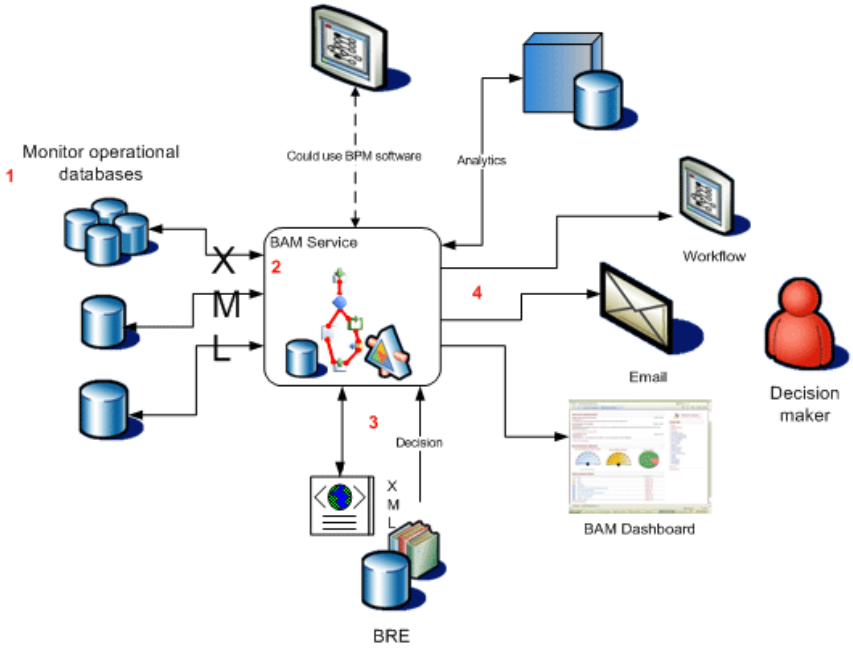

Figure 3. Business Intelligence General Architecture (Holden, 2007)

2. Business Rules Patterns

Currently there are several ways for textual specification of business rules, the main options being: informal (for example natural language or structured English), formal or semi-formal descriptions (for example OCL, UML, ORM, RuleML) or the use of a predefined business rules language, proper to all commercial BRMS. Business rules patterns are an alternative specification method, characterized by high generality and which offers a series of advantages, such as (REEDER, 2001):

- *Patterns are a first step in formalizing business rules statements*: often, the idea of introducing business rules is well received, but without carefully giving attention on how they are documented. By using a list of possible patterns, business rules identification becomes systematic and authentic.
- *Patterns encourage consistency in the presentation of the information captured by rules*, allowing those who describe rules to concentrate not on the presentation form, but on their content.
- *For business people, patterns eliminate ambiguity and enable validation*: unlike the diagrams that describe models of data or processes, the list of business rules may be reviewed by different people without any necessary explanations or additional documentation.
- *For developers, patterns appear to be Pseudo-Code*, especially since every term used to specify the rules can be mapped to an entity or attributed a value in the database model.

Starting from the categories of patterns described in (MORGAN, 2002), (VON HALLE, 2002) and (REEDER, 2001), this article proposes a potential list of rules patterns, suitable for describing rules during the requirements specification and analysis phases of the software development cycle. These patterns should be regarded as a suggestion for structuring the specifications and not as a mandatory requirement, with the possibility to

customize them according to the characteristics of the system. It must be noted that from the categories of patterns found in literature, those which offer guidelines were excluded, because they may create ambiguities and are not suitable for specifying business rules at the information system level.

A number of notation conventions will help to formalize the rules pattern, but will not appear in the rules statements. These are the following:

- round brackets () include a group of elements;
- brackets [] include optional items;
- vertical bar | separates alternative items;
- angular brackets < > include special items, which will be subsequently defined.

In order to build a pattern it is important to understand the precise meaning of its component elements. A list of variable and predefined patter elements is described in Table 1.

Table 1. Predefined pattern elements
(adapted from MORGAN, 2002; VON HALLE, 2002; REEDER, 2001)

ELEMENT	DESCRIPTION
<term>	A noun or noun phrase recognized as a business entity. To show more precisely the applicability of a rule, the entity can be qualified by other descriptors, such as its existence in a given geographical area.
<characteristic>	It refers to a behavior that the business must be conducted, or the imposing of a certain relationship.
<fact>	Emphasizes the combination of two or more terms, so it describes a relation between terms. When additional information is needed, the relationship between terms can be qualified by other descriptors.
<operator>	Help to establish the relationship between the two terms. The list of operators may include mathematical symbols or textual descriptions: =, <, >, <=,>= , <>, in, not, is not, has not, has at least n, has at most n, has exactly n, etc.
<facts-list>	A list containing elements of type <fact>.
<result>	Any value, not necessarily numeric, which has a specific meaning for the business. Often the result is a value of an attribute of business object.
<formula>	A definition of an algorithm based on which are determined the values of a result.
<x>, <y>	Numerical parameters.
<enumeration-list>	A list of mutually exclusive values that a term may have. An open list indicates that subsequent changes may incur as a result of changing requirements. A closed list suggests that changes to the list are anticipated. This distinction is intended to help choosing an appropriate implementation solution.
<condition>	Generic term that defines the antecedent of a simple rule, in the form of a combination of the following elements: <term>, <characteristic>, <operator> and <enumeration-list>.
<inferred-knowledge>	Knowledge arising from evaluation of one or more elements of type <condition>.
<action>	Actions arising from evaluation of one or more elements of type <condition>.
<value-term>	A value of a term for a particular instance.

As there are categories of business rules, there also is a variety of patterns for rules that to a certain point overlap the categories of rules. Rule-based development theory

requires that at the basis on any rules model, there should stand a model of facts, built upon the business terms. For this reason, the proposed list will include templates for the description of terms and facts. Likewise, there will be included patterns that describe classifications and enumerations, as they are very useful and helpful in assigning possible values for class attributes.

In the following there are presented the nine categories of proposed rule patterns, accompanied by examples regarding an ordering system.

Pattern 1 : Term

This template is intended to provide a widely accepted definition for a given business term.

<term> is defined as <textual description>

Examples:

Customer is defined as any person or company interested in purchasing the company's products.
Direct Customer is defined as any customer who buys directly from the store.
Online Customer is defined as any customer who purchases using the company's virtual store.
Discount is defined as a price reduction for a certain product under certain conditions.

Pattern 2 : Fact

This template depicts the relations between terms and may take several forms.

<term1> is a <termen2>
<term1> [may] verb <term2>
<term1> is composed of <term2>
<term1> is a role played by <term2>
<term1> ha a property of <term2>

Examples:

Customer places an **Order.**
Manager is a type of **Employee**.

Pattern 3: Classification

This pattern determines the truth value for an element of type **Term**. It is possible that such term has only a temporary use within the system. But if these are long time elements, then it would be preferable to define them through the **Term** template.

<term > is [not] defined as <value-term>
(if | only if) <fact>.

Example:

Rule 1	An order is defined as urgent if it has to be delivered within 12 hours.

Pattern 4: Enumeration

This template establishes a range of values which may be associated with a term or result.

<term>| <result > must be chosen from the following [open |closed] list :
<enumeration-list>.

Examples:

Rule 2	A customer category must be chosen from the following open list: Standard, Premium.
Rule 3	An order category must be chosen from the following closed list: Normal, Urgent.
Rule 4	Status of an order must be chosen from the following closed list : Placed, Accepted, Paid, Ongoing delivery, Delivered.

Pattern 5: Simple constraint

It is one of the most common rule patterns encountered in software systems requirements and aims to impose a constraint on the subject referred by rule.

<term> must [not] be| may be <value-term>
[(if | only if) <fact>].

Examples:

Rule 5	A normal order should be delivered no later than three working days.
Rule 6	An additional discount may be applied only if the product is not already included in a promotion.

Pattern 6: List constraint

Similar to the previous pattern, this describes a constraint imposed on the subject of a rule, the difference being that the imposing constraints will be taken from a list of facts.

<term> must [not] be| may be <value-term>
(if | only if) at least <x> [and not more than <y>] from the following conditions is | are true: <facts-list>.

Example:

Rule 7	An order must not be accepted if at least one of the following conditions is true: - Customer has placed five orders that are accepted. - The customer has not paid the last delivered order.

Pattern 7: Calculations

This pattern has the role to establish a link between business terms and to define the calculation formula of a term or result.

<term > | <result> is calculated as <formula>
<term > | <result> = <formula>

Example:

Rule 8	The total value for an order is calculated as the sum of the values of ordered products minus discount value plus transportation fees.

Pattern 8: Inference

This pattern contains a set of conditions which, when true, establish the truth value of a fact.

if <condition>
[AND/OR <condition>
AND/OR <condition...>]
then <inferred-knowledge> [is calculated as <formula>].

Example:

Rule 9	If a Premium customer has not paid orders within 30 days, then he is degraded to the Standard category.

Pattern 9: Reaction

This pattern checks certain conditions and, in case they are true, initiates an action or a suggestion for an action execution.

if <condition>
[AND/OR <condition>
AND/OR <condition...>]
then <action>

[<action> <action> ...]

Example:

Rule 10	If an online customer is registered, then display products which are recommended in accordance with his profile.

It's worth mentioning that the proposed patterns are not mutually exclusive, meaning that for a certain business rule there isn't a single type of suitable pattern. Specifications described in the form of Simple constraint and List constraint could be represented as Inference or Reaction patterns. The latter were introduced because, in terms of form, they are closer to the code that implements rules. Thus, depending on needs or preferences, developers may choose a specification style closer to natural language or programming languages.

3. Public Acquisitions Case Study

Increased organizational needs and quality standards, lack of management systems, as well as the current economic recession situation, require the need for integration of Business Intelligence solutions in public institutions. For monitoring progress and how the department adapts to legislative and economic changes, the manager and the staff within a Public Acquisitions Department (PAD) should monitor, control and report their performance fairly, consistently and directly. Lack of consistent and accurate data will aggravate the process of creating an environment within which performance will be properly assessed and improvements will be stimulated. Thus, at PAD level, a system of indicators should be created in order to provide a perspective on the performance of public acquisitions over time, on best practices in the field and on key performance areas to be improved. Next we will identify and specify some key business rules responsible for business decision-making in the development of a BI system for public acquisitions, through the following stage:

A. Identify Goals

a) The general Goals identified at the level of public institutions are:

- increase savings;
- enhance the quality of products/services offered;
- increase opportunity in the acquisition process, observing the laws and regulations in force.

For an acquisition process to be performance, efficient and effective, it should provide three main objectives: savings, quality and opportunity. One of the principles underlying the public acquisitions' contract awarding procedure is the efficiency of using funds. Thus, the main function that PAD must provide is to ensure that money is spent wisely. This will lead to savings on the purchase of goods and services that are needed by the institution.

B. Identify performance quantitative indicators for public Acquisitions

The process of creating a system of indicators is based on three steps: ❶ *identify the general objectives,* ❷ *identify key Acquisitions processes necessary to achieve the objectives and* ❸ *identify necessary organizational resources and imply the definition of indicators for all*

key performance aspects that have been identified (Ghilic-Micu et al., 2008). To achieve the general objective regarding the growth of savings, we propose the following set of indicators (table 2).

Table 2. The system of indicators related to the savings goal

Step	Specific indicators	Formula
❶	Savings (S)	$S = \sum_{i=1}^{n} Q_i (PE_i - PC_i)$, where n- number of products contracted over a period of time, Q_i – contracted quantity of product i over a period of time t; PE_i- estimated price for product i, PC_i – contracted price for product i
❷	The percentage of direct costs from total costs (I_{dc})	$I_{dc} = \frac{\sum_{i=1}^{n} PC_i \times Qda_i}{TV}$, where PC_i – contracted price for product i, Qda_i – quantity of the product i purchased through direct acquisitions; TV – the total value of acquisitions
❷	The percentage of costs by using electronic means (I_{em})	$I_{em} = \frac{\sum_{i=1}^{n} PC_i \times Qem_i}{TV}$, where PC_i – contracted price for product i, Qem_i – quantity of the product i bought using electronic means, TV – the total value of acquisitions
❷	The percentage of acquisitions procedures carried out online or in combination online & offline (I_{pe})	$I_{pe} = \frac{NP_e}{NP}$, where NP_e – the number of acquisitions procedures carried out using electronic means, NP – the total number of procedures performed
❸	Percentage of staff who attended training courses (I_{tc})	$I_{tc} = \frac{NP_{tc}}{NPers}$, where NP_{tc} – number of people who have attended at least one course for qualification in the field, completed by obtaining a recognized diploma, NPers – total number of persons
❸	The percentage of annulled procedures by the fault of the contracting authority (I_{pa})	$I_{pa} = \frac{NPA}{TP}$, where NPA – number of annulled procedures by the fault of the contracting authority, TP – total procedures performed

C. Describe each indicator as a completely specified business rule

For exemplification, we will specify in detail the business rule for calculating the "Savings(S)" indicator. *The objective of this business rule is the calculation of savings achieved by the contracting authority, as an aggregate indicator and divided for the following dimensions: time, buyer, procedure, product and supplier*. The facts in the data warehouse include estimated unit price, contracted unit price and quantity (figure 4).

Any public institution must measure the amount of savings depending on these dimensions, because this information is useful to managers in making timely decisions on how to reinvest and stimulate the acquisition process.

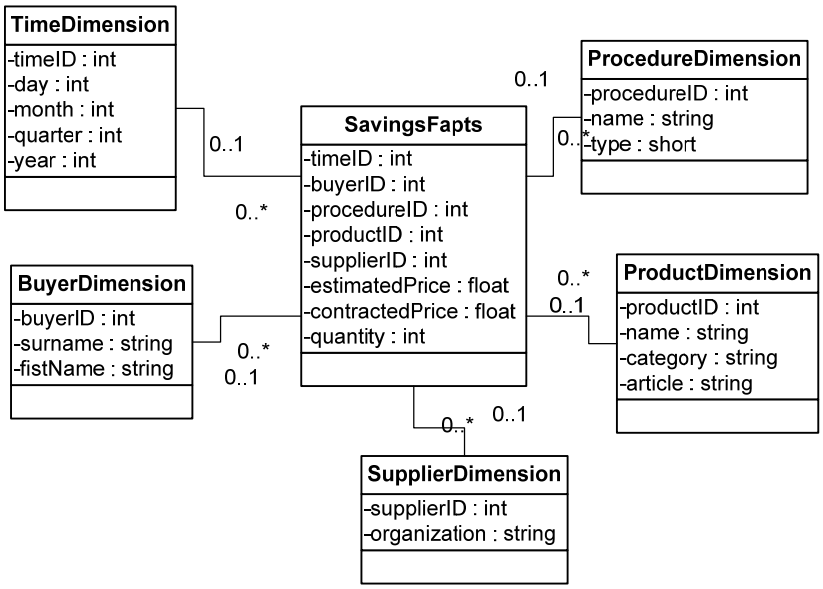

Figure 4. Dimensional model for savings

Savings is a calculation business rule, whose formula was presented in table 2 (❶) according to the business rules patterns previously proposed. In order for this rule to be completely specified and to be able to calculate Savings for each dimension, we have to provide the following additional information:

- *Savings Data Elements*: the important data elements are: product identifier, buyer identifier, supplier identifier, procedure identifier, time identifier.
- *Savings Classification*: the rule classifies the savings as related to a certain period of time, a type of procedure or product, a certain buyer or client.
- *Savings Computations*: For the aggregate indicator, within a period of time - using the product identifier, look up for the contracted price and contracted quantity within its corresponding contract row, then look up for the estimated price within its corresponding procedure row. For each dimension:
    - Procedure - using the procedure identifier look up for each productID and its corresponding estimated price, then look up for the product's contracted quantity and price in its corresponding contract row;
    - Product - using product category look up for each procedure identifier, look up for each productID and its corresponding estimated price, then look up for the product's contracted quantity and price in its corresponding contract row;
    - Supplier – using the supplier identifier look up for each contract identifier, then look up for the contracted price and quantity. Then look up for the product identifier in its corresponding procedure row and find its estimated price.
    - Buyer – using the buyer identifier look up for each procedure identifier, then look up for each productID and its corresponding estimated price and contracted quantity in its corresponding contract row.
- *Savings Comparisons*: compare contracted prices and estimated price and returns the quantity multiplied by the price difference.
- *Savings Control*: business rule will be applied in an aggregate form or for every dimension. Redline value for saving is zero.

In order to be able to calculate saving, the following related constraint business rules (specified using the 5th business rules pattern) must be evaluated as true in advance:

- A request on a product **must be** approved **only if** its estimated value is less then the available founds.
- The contracted value **must** be less or equal with the procedure's estimated value.

D. Publish business rules service

When the business process is ready to run the business rules, the necessary data must be provided. The business rule for calculating savings will need a simple type of registration which is expressed in a data type definition, such as::

```
<? Xml version="1.0" encoding = "UTF - 8"?>
<!ELEMENT SavingsRule (timeID, productID, buyerID, supplyerID, procedureID)>
<!ELEMENT timeID (#PCDATA)>
<!ELEMENT productID (#PCDATA)>
<!ELEMENT buyerID (#PCDATA)>
<!ELEMENT supplyerID (#PCDATA)>
<!ELEMENT procedureID (#PCDATA)>
```

When the business rule is exposed as Web service, business rules software will create the data type. Having the data type defined, the business process software automatically includes the data type in workflows.

4. Conclusions

Every organization operates according to a set of business rules. We can even speak of two major classes of business rules: external rules, coming from legal regulations that must be observed by all organizations operating in a particular area, or internal rules which define the organization's business politics and aim to ensure its competitive advantages in the market. Besides this, business rules are a critical component of any performance management system and any BI system. To be correctly managed, they must be treated as distinct artifacts in the software development process. A BI system typically includes a large set of indicators in order to measure the organization's performances. Frequently, these indicators act as calculations business rules, having several influence factors.

This article outlined the role of business rules within a BI project, especially in a SOA environment. It also proposes a language of business rules patterns, suitable for business rules specification during the requirements specification and analysis phases of the software development cycle. The results of our research were exemplified through a case study for public acquisitions.

References

1. Blasum, R. **Business Rules and Business Intelligence**, DM Review Magazine, April 2007, http://www.dmreview.com/issues/20070401/1079638-1.html
2. Debevoise, T. **Business Process Management, witch a business rules approac, Implementing the Service Oriented Architecture**, ISBN 978-4196-7368-9, 2007
3. Ghilic-Micu, B. Stoica, M. and Mircea, M. **Management of Acquisitions in Romanian Public Institutions**, Calitatea-acces la succes, nr. 93, Ed. Societatea Romana pentru Asigurarea Calitatii, 2008, pp. 344-350

4. Holden, G. **Reactive and Proactive Business Intelligence**, 5 September 2007, http://www.b-eye-network.co.uk/view-articles/5899 ,
5. Morgan, T. **Business Rules and Information Systems: Aligning IT with Business Goals**, Addison Wesley, 2002
6. Reeder, J. **Templates for Capturing Business Rules**, Business Rules Community, April 2001
7. Ross, R. **Principles of the Business Rule Approach**, Addison-Wesley, Boston, 2003 http://www.information-management.com/issues/20070401/1079638-1.html .
8. Von Halle, B., **Business Rules Applied**, John Wiley & Sons, Inc., 2002

---

[1] Marinela Mircea is an university assistant in Economic Informatics Department, Academy of Economic Studies of Bucharest. She published over 20 articles in journals and magazines in computer science, informatics and business management fields, over 15 papers presented at national and international conferences, symposiums and workshops and she was member over ten research projects. She is the author of one book (currently in issue) and she is coauthor of three books. In February 2009, she finished the doctoral stage, and her PhD thesis has the title *Business management in digital economy*. Her interest domains are: programming, information system and project management.

[2] Anca Ioana Andreescu is an university lecturer in Economic Informatics 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 over nine research projects. She is also coauthor of three books. 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, requirements engineering and software development methodologies.