A BUSINESS SYSTEM BASED ON AGENT TECHNOLOGY DESIGN

Vasile MAZILESCU
vasile.mazilescu@ugal.ro
Dunarea de Jos University of Galati, Romania

The work in hand compares Business Intelligence Systems (BIS) in relation to Intelligent Business Systems (IBS). We highlight the ways to shift from an E-business system to an IBS system, provide a solution for an Intelligent Business System based on Production Rules (IBSPR). We use this solution in developing a four levels Web application to solve a problem of planning, in compliance with all of the developing phases of an expert system, by using UML-Agent methodology in the analysis of the application and by using .Net technology.

Keywords: Model, Production rules, Rete algorithm, BIS, IBSPR

1. Introduction

Dynamic nature of economic processes and phenomena, their complexity and diversity, the prospect of economic globalization and decentralization, have determined the decision makers to focus on continually improving the methods and techniques aimed to support them, both at the microeconomic and the macroeconomic level. The relevant characteristics of economic processes and phenomena necessarily require approaching them systemically. The current trend is using on-line intelligent information systems as a support for developing new methodologies to improve the organization and the management of the economic systems. Intelligent e-business Systems, also known as IBS differ from BIS through the use of knowledge models to represent the constraints specific to a business and by including reasoning mechanisms to solve problems specific to Artificial Intelligence. This paper describes a model for an IBS that keeps the basic features of a qualitative cybernetic model, but which is extended through the concepts, the reasoning mechanisms, the knowledge representation models and the development methodologies of an expert system. Are also identified the features specific to the model-based approach of systems. To get to a modern business system we identify the processes of transition from a conventional system to an IBS as well as and the connections of such a subsystem with integrated systems for ERP and Business Intelligence. For modelling an IBS we use a cybernetic frame of the modern enterprise and we extend it by including, among the basic concepts, the elements of a rules-based production system as a variation of Rete algorithm. An IBS should be functionally integrating into a complete business architecture, and for this purpose we describe the models for such an architecture and explain the methodology for developing an IBS through modern projects and methods of analysis. From the practical perspective, this paper presents an IBSPR which is using the components identified in the structure of an IBS and provides a solution for a problem of planning.

Section 2 provides a comparison between BIS versus IBS. The aim of section 3 is to identify and describe a model for an IBS. For this purpose are highlighted the features, the phases and the benefits of the model-based approach. It is also described the architecture of an intelligent enterprise and are identified the directions for shifting from an E-Business model to an I-Business model. Are analysed the development phases specific to an IBSPR for representing the business constraints. For this purpose, we present an evolution of the methodologies for developing an expert system and we identify the development phases of an IBS also following the directions imposed by the project management methodologies. We define the business problem (automatic management of questionnaires for a car leasing financial institution), identify the relationship between the IBS and the credit analysis subsystem, we draw an equivalence between an Artificial Intelligence Planning problem and the business problem of questions' management, and we present a model of business intelligence application by including the developed components into the IBS structure, highlighting the relations between subsystems.

Next we identify the mechanism of formalizing through production rules, we transpose these into business rules and we present a methodology for the use and the development through UML-ORM constraints which can be used in defining a graphic representation language through XML structures. Structurally the application is built on 4 levels: the database level (which contains the stored tables and procedures), the middleware level of connecting to the database, the business components and the web forms. System testing is performed in accordance with the development methodology for expert systems based on verification and validation. Section 4 presents the conclusions of the paper.

2. Characteristics of BIS and IBS

A business model describes the logic of a business system to capture the value chains hidden in the actual processes. A business model is a description of the value that a company offers to a specific segment of customers and includes business architecture, the network of partners to create and distribute this value and the business relations capital, aiming to obtain profit and sustainable income flows. Business models (or system mock-ups) can be regarded as the connecting part between strategy and business processes: in strategy are defined and conveyed the objectives and purposes and the business processes should be able to understand information flows and to implement them. For a rigorous integration we can use an approach based on components, generating thus a commune and shared representation of the system that enables understanding the domain of discourse and communication between agents and distributed heterogeneous applications. The advantages of using a conceptual business model based on components are: the process of business modelling helps to identify and understand relevant elements within a domain as well as the relations between them; the use of standard and rigorous models helps managers to communicate and to share their own understanding of the economic world with other decision makers horizontally, but also vertically, with operational centres; are used work diagrams open and shared on the model, through which multiple agents can make changes on the model; the consistency rules ensure the continuous updating between all the business partners; a formalized business model helps to identify various measures for a business, such as evaluation diagrams or performance indicators; a model helps in the simulation and learning of the business processes; such risk-free experiments may be carried out without jeopardizing the safety of the organization. Online business (e-business) is obtaining economic value by using information infrastructure and the Internet network in a virtual frame of digital economy. The multiple opportunities and technological developments of the e-business environment toward an integrated intelligent model have been included in the basic features of the online business environment: (i) connectivity : globalization of businesses and communication has allowed changing from a linear chain of value - dependent on space and time - to a distributed network environment, independent of space and time but dependent on the Internet; (ii) interactivity: collaboration between customers and vendors has become highly specific, customized, were invented new business models (B2B, B2C, B2E) and it has been passed from the physical value transfer to the informational one; (iii) context : the interconnection of heterogeneous environments generates complexity and consequently the need for robust information systems, which work with rules and exceptions, that have embedded behaviour models for multiple business cases generated in a context shared between partners. These three characteristics define three plans of interaction between the logic levels of business: (i) technical logic (includes all the technical instruments of e-business and the technology used, and it's quantified through the virtual technological competence of the enterprise); (ii) market logic (i.e., customers' need to buy the business' on-line offer, the degree of innovation - customization, to exploit opportunities and to expand the classical business through means of electronic business); (iii) business logic (how to interconnect the business roles (customer, supplier, intermediate), the formation of structures for on-line communication and collaboration (that allows minimizing the costs) and the generic business architecture in which flows of value are oriented to services and operation). It can be noted that complete integration of all these sub-systems and business flows can be achieved only by unifying the value chains in a safe, reactive environment which allows adjustments and which works rigorously in compliance with the business rules. Such an environment should include intelligence in customer management and understanding of consumer's theory, intelligence in managing channels of activities and specific costs of e-business, intelligence that can be formalized, expressed and shared between the business partners. Since an BIS is an extension of the e-business systems and the integration with BI environments, business portals, modules for the management of the business rules, it needs to meet the following requirements, according to [1]: the existence of 1-1 type marketing applications which customize the information, the applications and the services that are related to third parties; applications for content analysis, work sessions analysis, interaction with client-systems analysis [2]; applications for measuring, analysing the communication channels and for evaluating the results of the interaction through an interface customized for each business component but associated with same trade policy; application for the analysis of supply chain, modules which manage supply in relation with present and future demand for a flexible use of the stocks [3].

Intelligent Business represents a change on the data warehouse and BI (Business Intelligence) systems and represents a complete systemic approach of the business flows: operational systems' fusion with the multidimensional analytical systems. Therefore, traditional data systems remain the centrepiece of storage, analysis and reporting and operational applications (part of the business processes) and portals can query the data sources. Hence, i-Business means BI integrated into operational processes of e-business. Intelligent businesses are based on a system reactive to an event, i.e. there is an automatic monitoring of all business activities in order to respond to requests in real time and to determine whether it is necessary the integration with other operational data before delivering data to applications. This concept removes the separation between operational and analytical systems and provides an approach in which the two systems communicate permanently: any business process collects BI to guide the operations to strategic business objectives; this constant call to BI resources is done in a transparent manner, being integrated in applications of business operations via BI web services that allow a dynamic interconnection. Automatic capture of events in business operations that trigger the integration with command data queries is done by a component of control for monitoring the business activities: BAM (Business Activity Monitoring). BI components are used as intelligent resources for command processing in the context of an operational task within a business process. Thus, process activities must be connected with the business objectives within the component of corporate performance management (CPM); this way the business processes will comply with the business objectives. One can associate to BI mechanisms various business activities in order for BI elements to be used to achieve a strategic objective. This is why the integration between CPM and BI is considered to be "critical". Furthermore, in this system structure the number of queries to BI (analytical instruments) will be exceeded by the number of queries coming from operational applications which are frequently used by employees, partners (customers, suppliers) [4,5].

3. THE MODEL FOR A BIS

To define a model of business intelligence we use the cybernetic theory concerning the elements of systemic modelling, and identify the classic elements of this theory with the business rules formalism (which are defined in the process of transition to a BIS in previous sub-section). A BIS is an iBusiness system oriented toward business and uses rules to define and control business interactions and activities. The transition from the generic model (systemic) to the business model (application) implies that in the general model of a BIS to be used a business model based on components to represent the particular business universe and the specific rules. Condition of identification between the levels of resolution (system-application) is carried out by approach at the model level. The term model is frequently associated with qualitative or quantitative description of a real system, and let S be a system defined as a set of subsystems Si, i∈N, and by the relationships rij between these subsystems. In the set S can be included subsystem S0 (environment). Real economic systems are perceived and described by people (observers, H), which have a certain level of knowledge, enabling them to develop ideas related to that system. The set of these ideas forms the universe of possible representations (U). To issue these ideas observers use not only knowledge but also instruments, defined as modelling techniques T. In addition to observers H, in the modelling process can occur also a beneficiary, the user (D) providing a certain finality to the model. We can say that model M is the result of quintet (S,H,U,T,D). The operational definition for the concept of model is: M is a model for S if and only if - conditions found in work [14]: M, S – they have elements, connections and achieve goals; for any element Si∈S, there is at most one sub model Mi∈M; for any relationship rij between the elements of S, there is no more than one corresponding relationship mij between the corresponding sub models from M; for any set of elements or sub models connected through a relationship mij∈M, it is true that there is a corresponding set of elements in S linked by a corresponding relationship rij. It can be observed that: M has at most the same number of elements as S; M is the most simple with respect to the relations between elements; M must be useful, i.e. what is said about M must be true also about S. To describe the structure of a systemic model of business intelligence we identify each member of the quintet with entities specific to a field of the business:

S - business system;

H - the observers from the classical definition are represented by the hierarchical structure of information flow in an enterprise: operational, executive and informational level. Thus, the decisions which are defined on these levels guide the system in its dynamics, and contribute to defining the purpose (problem) to be fulfilled by the business system;

U – the universe of possible representations shall be identified with the variation parameters determined by S system's interactions with external environment. Variation of these parameters determines and characterizes at each moment the operationalization context by generating the facts base (FB);

T - the modelling techniques which, in the classic model are attached to the observers, become the business rules specific to company's system, the business conduct that defines the rules base (RB); the relatively stable component that gives the system a logical character, and consistent in relation to the business directions. The business rules are represented by production rules and allow the definition of reasonings in the business universe analysed;

D - the beneficiary from the classic model, i.e. system's finality, is identified with the internal parameters of S, those factors for the evaluation and quantification of the results obtained by attaching reasoning to the business knowledge model analysed. The control parameters of the system are determined intrinsically by the relationships between the 5 subsystems of the dynamic enterprise structure.

Thus, BIS, as a subsystem of the intelligent enterprise, is defined starting from the general cybernetic system of the enterprise, from the identification of the components specific to a model-based approach (in order to enable the identification specific to a business model based on components) and from adding of elements specific to expert systems, in order to allow the use of knowledge representation and processing mechanisms of the intelligent systems.

The observer H considers that $M \subset U$, is a model of the real system S if and only if the information IM is the equivalent in a certain sense with information IS, which objectively exists in the real system. Extracting this information is done by using a modelling technique from the set T. The model obtained shall not meet only the requirements of observer H but also those of the beneficiary D which has some information about the system noted I_{S1}. The information customizes a model; the model of a real system can have different shapes but its information content remains the same. The information is what is extracted from the model in the process of analysis of the latter and corresponds in a certain way with the modelled system.

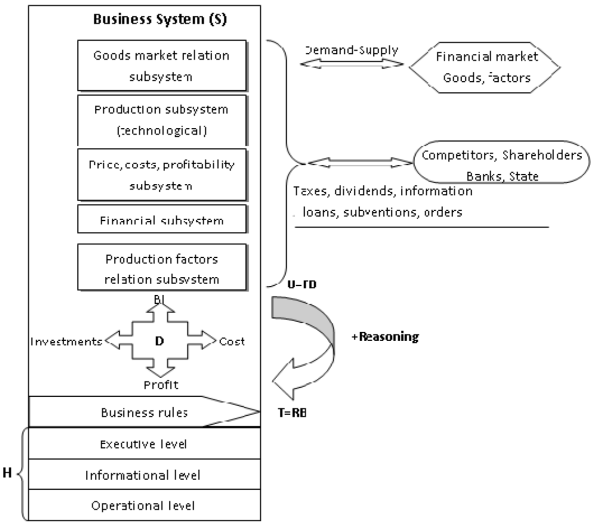

Figure 1. IBS structure

Cyber modelling appears as the result of an influence mediated by the observer between the real system S and the model M. On this information has been built the global diagram of the cybernetic modelling process: M occurs when the real system S has a corresponding influence on the ideal system S* studied by the observer H. M is built based on the informational content I_S by using a modelling technique T. Informational content of system S* is obtained through a reverse conversion. A qualitative model is obtained when I_S is approximately equal to I_M. Since a BIS can be considered as a system S, and an AIA is also a system (S*), what is to be obtained by modelling system S on a particular area of business (being therefore a subsystem with a certain business resolution) by using rules and structures specific to an ES, and an AIA is a model based on components for an i-business system, similar to e-BCM model (a model based on components for an e-business system) and since for the representation of the universe of discourse specific to the internal modelling of a ES, is generated a correspondence between the business system and the knowledge base, it can be considered that an AIA should be a qualitative model for a BIS.

Defining the Problem Specific to BIS

To describe a business problem (granting of car leasing) we consider a concrete case of market, identify a situation of the system that can be improved and we make an analogy with a generic problem of artificial intelligence: planning. Let a financial company which specialises in refinancing and in the analysis of applications for car leasing, with over 200 centres widespread throughout the country and located in different social-economic environments. Each office is responsible for requesting financial information from customers and in charge of carrying out a relevant questionnaire for the evaluation of credit scoring. Since offices are located in different socio-economic areas, the relevant questions are quite different and the information collected is not always consistent with the specific problems of financial loans. Initially the company created a form which contained each question possible and employees filled out only the information that was specific to the customer. This system has not worked since the forms were very long, time consuming and most times the desired information was not on the form. A web application was developed in order to separate forms depending on the centre, and the subsidiary manager chose only the questions that were considered relevant. If employees wanted to add a new question, they had to send an e-mail to the database administrator, which had to introduce the question and to confirm. This process used to take usually 1-2 days, and sometimes data was not collected or it became irrelevant.

BIS solution: To improve application's efficiency, have been created rules based on the business functionality. If an employee wishes to introduce a new question, the application checks its existence, and if it doesn't exist then are activated business rules that determine the place and the way in which will be performed this process of adding, with a view to facilitating the management of questionnaires for customers assessment. Each time an employee wishes to add a new question, the changes are saved in the log tables, and the system is periodically evaluated using these logs. If a field was not changed properly the administrator must adjust the rules.

Generic planning problem: Is defined a system S, an initial state x_0, a set of restrictions R valid for system S, and given a set of resources Res, the planning problem (resources allocation) is to determine the existence of a finite string of states x_0, x_1, ..., x_f so that, in the presence of restrictions R, the set of resources to be used for the purpose of obtaining a final state (it is not required to a priori know the final state). A state of the system is determined through the distribution of system's resources, distribution which is valid if system's restrictions are met. Thus, starting from the classical definition of the problem of planning we make an analogy for the case met in the above business context, and define the main functionality of the system: Automatic management of questionnaires for a car leasing financial institution. The business intelligence system that will implement this function is a CRM type sub-component, which gathers, uses, and transmits information related to customers from the generic architecture of intelligent enterprise as defined in figure 2.

Being defined:

A financial system for the leasing with the following components relationally structured: The business component - C1, expressing the information required for the management of the financial leasing business; questionnaires component - C2, expressing the structure of automatic management of the knowledge pieces (questions) dynamically forming questionnaires for assessing the customers; business rules component - C3, expressing constraints of the process in the form of production rules;

A set Q = {f/f-questions, f = 1, n} where n represents the questions existing in the system. Set Q has a correspondent set of answers A = {f/f-answer, f = 1, n} structured in c > 0 disjunctive categories, Categ = {Pc(A) where 0 ≤ c ≤ n}, the function funct: Q → Categ, funct(f) = ci where f∈Q and 0 ≤ ci ≤ c, is bijective (i.e., for each question there is only one answer and there is no answer without a question),

The set of constraints specific to the automatic management of evaluation questionnaires for leasing: identity constraints - ↨ , for any 2 questions i, j with i ≠ j and i, j = 1, n must be valid f_i↨f_j : ↨ means that the 2 questions are not identical from the semantic point of view, i.e. do not require the same answer; for any 2 categories Categ(s), Categ(j) with i ≠ j and i, j = 1, n must be valid, Categ(s)↨Categ(J), i.e. the 2 categories do not group answers from the same semantic field - the degree of matching between categories is variable and can be changed by the user by a percentage 'Ω.

Structure constraints [Categ] = c complies with a normalized relational structure;

Communication constraints: any change in the structure of Categ(c) involves a communication message between system's business and questionnaires components, C2 - C1;

Let a new question f_x be introduced by a user (agent) of the system; to be determined if can be added - (fx↨fi)'Ω, i = 0,n and fi∈Q - respecting the constraints (identity, structure, communication), question fx in set Q so that funct(fx) = cf where 0 ≤ cf ≤ c+1.

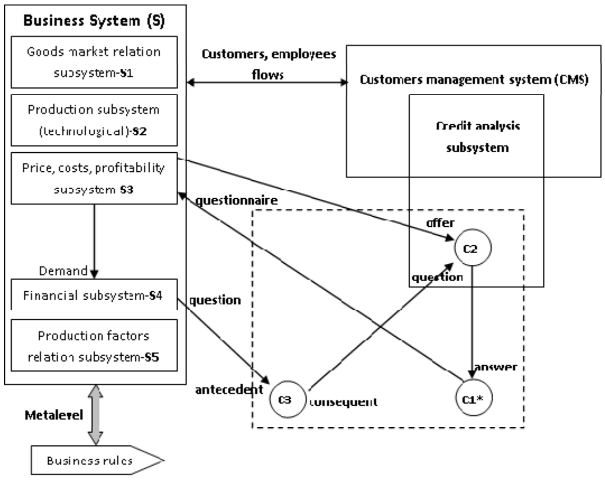

Figure 2. Schematic interconnection between S-GSC-BIS

Identifying the Problem

S = the financial system of car leasing that uses a database in which are found the 3 components of the system. S is the context of the problem, in which the three components of the system are modelled relationally: (i) questionnaires component C1: stores the base of answers on all questions of all the questionnaires **-** i.e. are stored results, effects of a questionnaire and not the structure itself; this component has a dynamic structure because it changes every time a new question is added to C2 and obviously will be added also a new answer to C1; (ii) business component C2: allows modelling the financing business in the sense of the interdependence with ERP system's databases (customers, employees), stores elements of connection with C1 through the base of answers and stores thus the structure of the questionnaires; (iii) business rules component C3: it is of type Rule-TypeRule, the elements from antecedent are correlated with the questions in C2 and the elements from consequent must be correlated with structures from C1; the effect of activating a rule is storing the question in the questions base of C2 and indirectly creating the area of response in C1.

R = business constraints, translated into directions of action for managing the questions base in such a way to ensure the consistency (relational) of the model and the automatic management of the questions-answers base, to maintain the normalization of the database, to minimize the semantic redundancy of questionnaires, to allow the communication through messages with users (agents); these constraints depend on the dynamic of the business and on the organization of financial institution's processes, and for the case of the automatic management of questionnaires for a system of leasing we have the 3 types of constraints: identity, structure and communication.

Res = **Q**, questions, i.e. the resources to automatically manage questionnaires are represented by the knowledge pieces which dynamically compose these questionnaires. This resource generates internally the set of answers A which represents the component that will complement the management of the questions within the system, providing the support for storing their effect.

The types of rules express the business actions, template factorized rules expressing the 3 types of constraints of the system S, and represent directions of action for the business frame defined:

– **identity rules:**

IF QUESTION CONTAINS @Phrase
THENS QUESTION BELONGS TO @ParamCategory
Case: @Phrase= {car, motor vehicle} → @ParamCategory= {TableAnswerAuto}

– **structure rules:**

IF QUESTION CONTAINS @Phrase
THENS FIELDTYPE SA @ParamType
Case: @Phrase= {how many, how many times} → @ParamType= {integer}

– **communication rules:**

IF QUESTION ENTERED BY @Employee
THEN EMAIL TO SA @Email
Case: @Employee= {Employee1} → @ParamType= {Employee1@domain.com}

Each type of rule contains more than one instance of business rules corresponding to the type of constraint represented. For each of these rules a priority is attached both to antecedent and to consequent, values which will be used in the selection process. For each type of category will be a default rule that will be activated in the case that for the constraint in question there is no instance in rules.

$\mathbf{x_0}$ = the initial distribution of the n answers within the system's base; i.e. their semantic ranking in categ(c) where 0 ≤ [Categ(c)] ≤ n;

$\mathbf{x_f}$ = the final distribution of the answers in categ(c_f), where n ≤ [Categ(c_f)].

Using the UML designing methodology (Unified Modelling Language) is carried out the conceptual modelling of a BIS, through static diagrams (case diagram and class diagram) as well as through behaviour diagrams (state, activities, chain and collaboration).

OCL-ORM formalizing of the business rules component

The concept of rule has different meanings depending on the scope of application, either specific to business rules approach, or specific to the traditional approach of expert systems. These approaches have in common the concept of inference rule, i.e. a production rule involved in at least one inference cycle within the inference process which takes place in a rules-based system. Production rules are, according to [14], IF/THEN statements, also called cause-effect relationships, precondition-conclusion, hypothesis-action, condition-action, test-result, and carry out a representation of knowledge in the form of:

IF <precondition> THEN <action1 >. The production rules differ from the IF-THEN-ELSE commands from the imperative programming languages, because they are relatively independent of each other and are based on heuristics, on the experience- originated reasoning and not on algorithms. They may even incorporate uncertainty, a numerical value assigned by an expert, resulting from his experience in solving the problems.

To succeed in modelling the inference rules by using the UML methodology through OCL, must carry out an analysis from an object oriented perspective. Thus, business rules, in an object based approach, find their significance in UML, as follows: Structural constructions - classes and relations between classes; Simple constraints expressed in the graphical mode of UML notations; annotations on the elements; Formal constraints expressed in OCL in the form of pre/post conditions and invariants. OCL is a language that allows the use of expressions and constraints on object-oriented models. An expression is an indication or a specification of a value defined on a context. A constraint is a restriction to the values of a model or system. OCL is a standard query system, an analysis and design component specific to UML. There are 4 types of constraints:

Invariant: is a condition to be fulfilled by all instances of a class or interface; it is described by an expression that returns True if invariant is complied;

Precondition: is a restriction on an operation that must be true when running an operation; are used to represent obligations;

Postcondition: is a restriction that must be satisfied at the end of execution of an operation;

Guard condition, must be satisfied before triggering an event of transition between states.

Most business rules work at system-level as constraints on the business process, for example transactional rules which prevent online transactions or updates of the data. Numerous development methodologies based on components convey the fact that the rules specific to a business express, first of all, constraints. However, implementing the inference rules is not connected with defining transactional constraints. An inference rule cannot be implemented as such a restriction because a restriction prevents a situation, when a production rule defines a derivation, and assumes that a specific situation is true; a production rule develops, does not restrict. The production rules are usually rules in an expert system, where they are executed by the inference engine. This engine provides a model of reasoning and a control strategy (with forward chaining or data-oriented/ backward, or goal-oriented) to execute unordered expressions from the state space by discovering connections, in order to solve a problem specific to artificial intelligence domain.

4. Conclusions

OCL expressions must not have side effects because the constraints cannot invoke actions, and inference rules' purpose is to change the state of the system, i.e. aiming to produce effects. Though appears to be an incompatibility in the representation of the rules through OCL expressions, it is noted that at transactional level, the constraints prevent actual physical events and therefore are affected possible instantiations of the model. At the modelling level (conceptual), rules can be modelled through constraints, i.e. expressing model's instantiations which they prevent. Thus, a template for the representation of rules through OCL must contain: An UML pattern for modelling the structures involved in the rule; a condition of safety, an invariant, which expresses the fact that a structure cannot break certain conditions; a "survival" condition, or a mean to ensure that a change of state will be produced. These conditions are still constraints, but which express hypotheses under which rules generate actions.

From the object-oriented point of view, inference rules can be implemented as: rules of rank 0 (compel the behaviour of an individual element, such as a class or an instance, these rules are valid for a class); rules of rank 1, also known as pattern matching rules, operating on sets of instances by processing an action on each instance of a set or on each t-uple of the instance if the matching expands to more classes. Pattern matching rules are usually executed by a RETE algorithm. When a system is developed through prototyping, each phase of the process can be checked and validated without waiting to complete the system, as is the case in conventional information systems. Testing process is complex and is aimed at verifying and validating system's intelligence in relation to human intelligence. In all cases the knowledge engineer must define a search strategy to use a minimum set of cases in order to detect errors. Such a strategy can be exhaustive testing, i.e. the strategy which involves all possible combinations of values for the inputs in the inference processes. This strategy is an expensive one, time consuming and impractical when comparing the total cost with the marginal benefit. Problems occurring, specific to testing stage may be: (i) the subjective nature of intelligence testing: solutions to many problems involve subjective decisions which cannot be considered as good or bad. Experts continuously update their knowledge, skills and judgments. Systems' answers are often heuristic; (ii) lack of reliable, secure specifications: inconsistent specifications determine also an arbitrary and unreliable verification /validation. Must be specified also the situations in which the system may not work; (iii) problems in establishing the consistency and correctness: the engineer must check both the knowledge base and the inferential processes. Checking the correctness and consistency of solutions is problematic because the system can continue to operate correctly even if a part of the inferential process is incorrect; (iv) exact determination of what is indeed an error: this can be achieved only by expert's systematic participation at testing; (v) the sources of test cases: cases can be generated by a cognition, by an expert or by generating programs. Regardless of the source, the results shall be evaluated; (vi) danger of negligence in testing: this comes either from the difficulties in generating test cases, or if the system, by its very nature, is difficult to be tested and then verification and validation can be carried out superficially; (vii) inappropriate instruments for self- verification: when the system has many parameters with multiple potential values, there is a danger of increasing the number of rules and implicitly the difficulties in generating the test cases. For decision tables there are tools for automatic check of conclusions' consistency, redundancy and adequacy. They contribute to achieving significant time savings in the testing work; (viii) the complexity of user interfaces: the situation is valid especially when some generators provide inputs difficult to regenerate reason for which solutions could be unsafe.

REFERENCES

[1]. D. Merritt: Best Practices for Rule Based Application Development, Microsoft Journal 2004
[2]. Amine Platform: http://amine-platform.sourceforge.net
[3]. Conceptual Graph Standard: http://users.bestweb.net /~sowa/cg/cgstandw.htm
[4]. Conceptual Graphs Integrated Tools allowing Nested Typed Graphs: http://cogitant.sourceforge.net
[5]. Synergy Platform: http://www.insea.ac.ma/CGTools/ CGTools.htm
[6]. K. Wilson, I. Maung: UML Patterns for Modeling Inference Rules Using OCL, Computer Associates Intl., 2006
[7]. J. Sowa: Ontology, Metadata and Semiotics, ICCS'2000 in Darmstadt, Germany (http://users.bestweb.net /~sowa/peirce/ontometa.htm)
[8]. Business Rules Community website – http://www. BRCommunity.com
[9]. R. Sara: Building Intelligent .NET Applications: Agents, Data Mining, Rule-Based Systems and Speech Processing, Addison Wesley, 2005
[10]. OCL Presentation: http://www.klasse.nl/ocl/ocl-introduction.htm