

Business Rules Based Agile ERP Systems Development

Aidas SMAIZYS, Olegas VASILECAS

*Information Systems Research Laboratory, Vilnius Gediminas Technical University
Saulėtekio 11, LT-10223 Vilnius, Lithuania
e-mail: aidas@isl.vgtu.lt, olegas@isl.vgtu.lt*

Received: November 2008; accepted: May 2009

Abstract. Business rules are relatively new addition in the field of Enterprise Resource Planning (ERP) systems, which are kind of business information systems, development. Recently some relevant enhancements of existing business information systems engineering methods were introduced, although there are still open issues of how business rules may be used and improve qualitative and quantitative attributes of such kind of information systems. The paper discusses existing business information systems engineering issues arising out of using business rules approach. The paper also introduces several ways of business rule involvement aiming at ensuring ERP systems development agility based on running researches in the field also carried out by the authors.

Keywords: ERP systems, business information systems, business rules.

1. Introduction and Problem Statement

The issues of Software Systems (SS) development and implementation into the Business Information Systems (BIS) in context of rapid changes of business environment are an important part of nowadays agile systems, including Enterprise Resource Planning (ERP) systems, development methodology and automated decisions making based on Business Rules (BR) approach. BIS is usually defined as a set of business practices, procedures and processes that are implemented into the software systems. A set of such applications can be combined into the Management Information Systems (MIS) used for a planned system of collecting, processing, storing and disseminating data in the form of information needed to carry out the functions of management, while ERP system incorporates a set of applications that are not necessarily focused on decision support.

According to Chung (2003) ERP system extends Manufacturing Resource Planning (MRP II) system, which refers to the planning of manufacturing schedules taking into account the time-phased material availabilities, ordering, and the capacities of production facilities, to include engineering, finance, human resources, and other activities in the enterprise. ERP system intends to integrate and automate various back-office functions of management and control activities such as finance, human resources, marketing and operations. Such integration of former separate software systems into one modular system means great effort for developers, because of needs of business process standardisation

and use of same object and data model across the enterprise. ERP system is a packaged software solution that addresses the enterprise needs of an organization by tightly integrating the various functions of an organization using a process view of the organization. ERP systems main goal is to integrate data and processes from all areas of an organization and unify it for easy access and work flow. This is the main reason which drives fast ERP systems expansion into former areas of business systems not dealing with ERP systems.

In general a business process is a partially ordered set of linked activities that create value by transforming an input into a more valuable output. Both input and output can be artefacts and/or information and the transformation can be performed by human actors, machines, or both (Ciuksys and Caplinskas, 2007). From the historical perspective it is important to highlight expansion of evolving ERP systems to support business processes from operational through tactical to strategic levels. Such expansion enables implementation of more complex decision and logical derivation processes and requires the use of new technologies and methods to support more intelligent behaviour of ERP systems. ERP systems are as a core component, which expands through the different business levels (operational, tactical and strategic), integrates and includes several formerly separate Information Systems (IS). This leads ERP systems becoming a synonymous of IS (though ERP systems like BIS, MIS and Executive Information Systems (EIS) are specialization of general IS) and such a border between ERP systems and other kind of IS is going to be heavily distinguished.

On the other hand most ERP systems in enterprises already became huge and complex structures which are difficult to maintain, change or develop new functionality needed to provide artificial intelligence features needed for further decision automation. Growing maintenance, development costs, security issues, globalisation and business distribution across the different countries with different business rules leads to the opposite process of ERP systems disintegration, when separate business processes for individual functional areas are developed in fragmented manner. To stop such disintegration, development of new more sophisticated and intelligent ERP systems development techniques and methods are needed. That also might be provided by involvement of business rules approach combined with agile ERP systems development process.

BR are relatively new addition in the field of BIS (and ERP systems as well) engineering (Bocij *et al.*, 2005). It has been accepted since year 1988 that BR are an important element of all type of IS (Van Assche *et al.*, 1988; Zachman and Sowa, 1992; Loucopoulous and Katsouli, 1992) and BIS as kind of IS therein. Object Management Group (OMG) document "Semantics of business vocabulary and business rules specification" (OMG, 2008) defines a BR as "a rule that is under business jurisdiction". BR from database engineering perspective are somewhat similar to integrity constraints (Date, 2000) or various programming language constructions used for description of structured logics for making decisions in software systems. Some authors discuss BR similarity to requirements (BRG, 2000; Hay, 2003) or state that BR can be used to represent both user requirements and conditions to which the system should conform (Valatkaite and Vasilecas, 2005).

In general a BR was defined in Guide project (Hay *et al.*, 1997) as "a statement that defines or constraints some aspect of the business". According to the OMG the BR resides

at the borderline between business engineering and software engineering. The Business Rules Group (BRG) gives another definition using business and information system perspectives: “From the business perspective a business rule is a directive, which is intended to influence or guide business behaviour, in support of business policy that is formulated in response to an opportunity or threat. From the information system perspective a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behaviour of the business” (BRG, 2003).

In particular business systems are functioning according to BR approved in specific business domain. Due to dynamic nature of business environment and accordingly BR, they are changing frequently reacting to internal and external influences such as changes in law, new competition, etc. (von Hale, 2001). Such changes challenge further inconsistency of existing BIS and already automated decisions or decision support in MIS and ERP systems with new business situation are needed to be adapted to the new business needs and changed BR – a subset of the enormous number of BR enforced by BIS. From a business point of view, nontrivial BR may be context dependent in space (e.g., different law in different countries) and time (e.g., law may be changed). Therefore, the rules have to be maintained during the life-cycles of rule-based applications (Herbst *et al.*, 1994). According to the traditional IS, and BIS in specific, engineering methodology changes in BR should be propagated through all the reengineering lifecycle stages starting from problem domain analysis, requirement specification, design, etc. (Fig. 1), where BR are involved in requirements and implemented into the BIS software code as new or modified executable instructions or kind of production rules used for automated decisions later on.

Such traditional BIS engineering methods do not allow storage of the same rules in one place and spreading them across the applications code servicing automated decisions according to the same business logic. It makes further modification of software systems very complicated and resource consuming, but enterprises cannot allow business system to function according to inadequate rules and need achievement of fast BIS adaptation, moreover with minimum costs.

Another problem that should be solved in case of using traditional methods is that changes of BR usually influence not only one isolated component, but few of them and changes in business logic should be propagated through all the isolated and integrated software applications of BIS servicing business processes of business systems that are affected by changed BR, when separated applications in the same enterprise are acting according to slightly different rules. This leads to incompleteness and redundancy of these rules, which may be specified in different ways in several software applications or may be implemented using different technologies, or get lost in redundant pieces of code.

At least one more problem also exists. BR of particular problem domain can be similar in several enterprises, but they are rarely applied in the same way, even if they depend on the same legal regulations. It is an extremely actual problem, when enterprises implement separate software systems customising standard versions of business processes and BR already built into the standard installation package. Such implementations together with integrated software provide unexpected results. This leads to the need of business

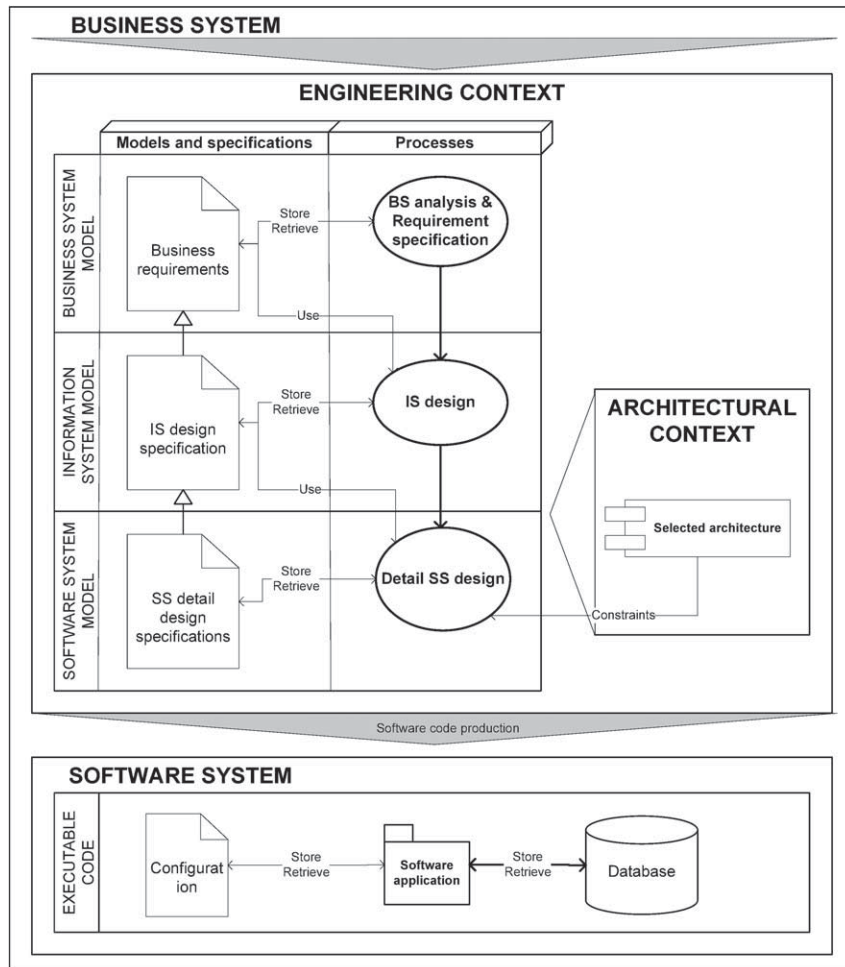


Fig. 1. Traditional information system engineering process.

logic exchange not only between software of BIS in one business system, but the need of business logic exchange between several business systems of same enterprise as well.

The current use of separation of system components into the application and data layers does not seem to be sufficient nowadays. The paper discusses agile ERP systems development and automated decision making using separation of business logic represented in BR and logical reasoning components out of the ERP system components developed in a traditional way. The aim of this paper is to analyse commonly used methods and introduce some relevant enhancements with respect to their expressiveness regarding BR. Finally a comparison and discussion on the analysed and several suggested ways of BR involvement is presented.

2. Business Rules in ERP Systems

At the end of the 1960's, a new idea to create an integrated data model within a single organization arose. Further evolution of Enterprise Resource Planning – abbreviation ERP – systems closely followed the spectacular developments in the field of computer hardware and software systems (Hossain *et al.*, 2002). Nowadays ERP systems in some context also called BIS are not only supplying information for all departments within an enterprise to support decisions, but are used together with other software systems to automate business processes. More and more functionality is implemented into the integrated ERP systems and now it is difficult to define exact boundaries of such systems. However all the computerised systems in one particular enterprise must or at least should operate according to the same set of BR. In this section we will discuss using of BR in enterprises and its IS with main focus on the ERP systems.

In the year 1987 J. Zachman defined IS architecture, first “by creating a descriptive framework from disciplines quite independent of information systems, then specifying information systems architecture based upon the neutral, objective framework and view of each participant in the IS engineering lifecycle (Zachman, 1987). This framework has become known as the Zachman framework by the name of its author. The initial Zachman framework consisted of three columns (for data, process, and network aspects descriptions, respectively) and five rows, making a 15-cell grid structure. In the year 1992 J. Zachman with J. Sowa extended the framework adding three additional columns: people, time and motivation. This was the first time when BR were explicitly introduced in the motivation column and found its place in process of IS engineering. The sixth – motivation – column of the framework represents BR aspect in business systems and IS of the enterprise according to the different views: scope, business model, system model, technology model, components and describing the final implementation of BR in a newly functioning enterprise, starting from vision, mission, goals, business policy and BR until the final implementation in particular software system process executed with respect to the BR, which either prescribe a certain action or constrain a set of possible actions in the final software system. Relations between motivation and other columns of Zachman framework are explicitly presented in Hay (2003).

All kinds of known BR have found its place in nowadays ERP systems, especially due to the emerged new multidimensional data analysis tools and software based on warehousing techniques and On-Line Analytical Processing (OLAP) technologies (Mahajan, 1997). Warehouse data based business analysis tools are widely used in enterprises for collecting and analysing of large data sets using several Data Bases (DB) of such systems as ERP, Data Mining, Expert and Decision Support, Customer Relationship Management (CRM), Production and other systems.

Structural BR in current ERP systems represented, e.g., in OCL (Object Constraint Language) can be used for data model design, constraining structure of entities in data model and implemented as integrity constraints in Active Database Management System (ADBMS). While the operational rules are usually expressed as the requirements in the traditional IS engineering process (Fig. 1) and implemented further into the process models of IS and its software system code as various case scenarios or methods of the objects

for implementation and execution of business logic in the application layer of the software. By emerging ADBMS the part of operational rules can be implemented directly as triggers (Valatkaite and Vasilecas, 2002) or stored procedures for execution of business logic in data layer of BIS applications.

von Hale (2001) proposes to separate the process of business rule discovery into two parts: business rule representation and relation with initial requirement specifications and specification of business rule itself. While initial requirements are discovered by analysing processes and events in business process model or use cases. After such definition of relations between events and actions it is possible to specify decisions making process according to captured and represented BR. This is a typical scenario of business logic representation using ECA (Event–Condition–Action) rules and their further implementation in the executable code, e.g., by triggers (Avdejenkov *et al.*, 2008).

This approach is the base for nowadays agile methods of the ERP system development dealing with separation of business logic out from executable software system code and maximum elimination of human factor in the way of BR implementation from BR management at business systems level until execution in the software applications. Such separation of BR provide some new possibilities for management of business logic directly by business users at business system level and future implementation using transformations and messages sent through dedicated interfaces as it will be discussed in the following sections.

3. Business Rule Based ERP System Development

The importance of BR in BIS development is motivated by many authors: Ross (2003), Barbara von Halle (2001), Herbst *et al.* (1994), Morgan (2002), Chisholm (2002, 2004), Date (2000), Bajec and Krisper (2005) and well-known organisations like Object Management Group (OMG, 2008) and BRG (2005).

Usually the traditional IS development process is vertical – from abstract business needs specifications to the detail software data model and data processing specifications and the finally to executable code and database scheme of software system (Fig. 1). This leads to ERP systems, where business processes usually are implemented in black box way, when business logic of the processes is hidden from the business user. But in our opinion it would be great advantage when some of the services could be rules-based and expose rules to the users with opportunity for them to manage rules directly and make system transparent.

The business rule approach makes IS development process not only vertical but also extend horizontal dimension. This happens because in addition to business system and information system level specifications we intend to produce separate executable model or new software system components that will operate on higher level of more abstract information processing and will be integrated with traditionally developed data processing components. The typical example of such horizontal development is the engineering process according to the service oriented architecture (SOA) presented in Smaizys and Vasilecas (2009).

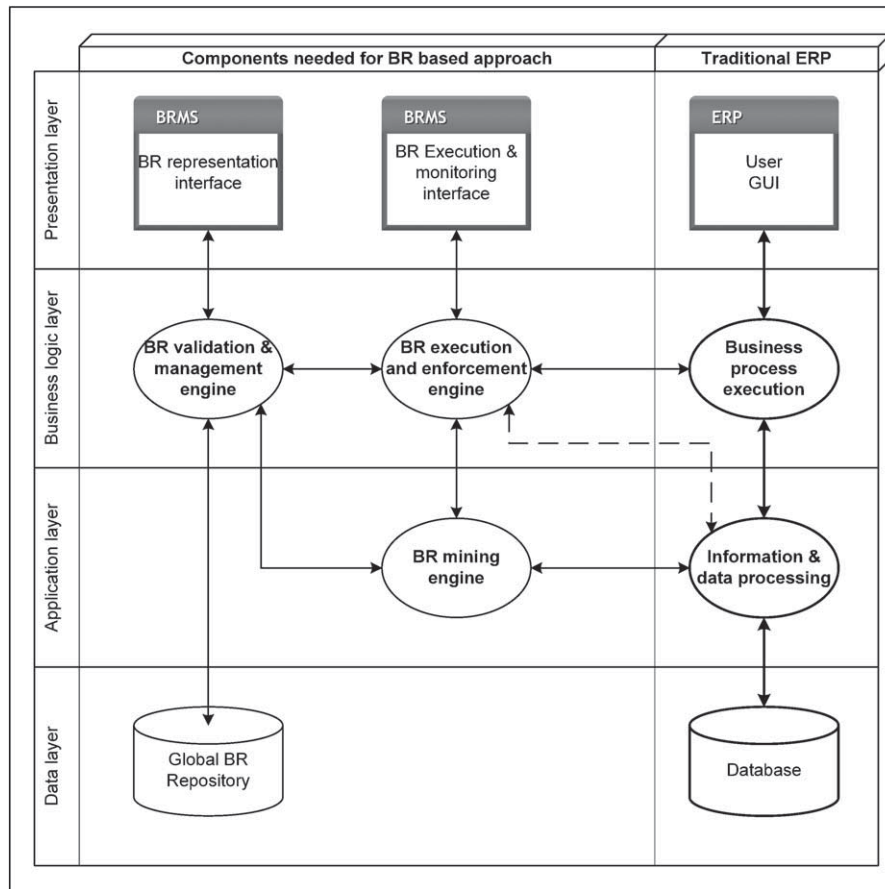


Fig. 2. Additional components of architecture needed for ERP engineering using BR based approach.

BR approach to the ERP system development first of all means separation of business logic, development of new architectural components servicing BR and their integration with components developed in a traditional way. Contrary to the traditional systems BR based systems can be viewed as not consisting of two, but of three major layers – data management, presentation (or end-user) and BR layer (Chisholm, 2002). In concern to SOA we propose to distinguish data, application, business logic and presentation layers to represent major components needed for BR based systems development (Fig. 2). Separate newly introduced components are used according to the needs, purpose and selected strategy of BR approach implementation and this is discussed in detail in Section 4.

Such new architectural components should be designed in development process. This leads to modification of traditional engineering process introducing new processes and development of new methods, languages, specifications and tools. First of all BR should be captured, specified, stored and represented in a formal and informal way. Then some component for BR interpretation usually named as business rules engine is needed. In-

terpreted BR should be executed or enforced calling specially designed executable instructions in application or database layer, or transformed into some executable components, or at least some part of them for servicing business logics (Vasilecas and Smaizys, 2007, 2008). We also have carried out research on business rule transformation into the requirements and some design specifications (Vasilecas and Smaizys, 2005) for further engineering using a traditional way of IS engineering, but such a way of IS development from our point of view is less productive.

Summarising all our obtained experience we can state that there is no existing standard and universal method or language for BR based ERP development. All the proposed methods are suitable only for modelling of specific BR classes and are limited by the architecture and tools used. The main development approaches of BR based ERP systems are discussed in Section 4 and details of BR based engineering processes are discussed further in this section.

3.1. Business Rule Specification and Representation

A representation of BR at a conceptual level offers several advantages. Design of the system and independent implementation of the business user oriented conceptual model may enable simulation of behaviour of logically dependent event–action–sequences or automated generation of rule-preserving implementations. According to Herbst *et al.* (1994) several types of conceptual models do not provide a systematic treatment of BR. Nevertheless, the available constructs for function-oriented BR specification like UML (Unified Modelling Language) (Erikson and Penker, 2000; OMG, 2007), Data Flow Diagrams, Merise conceptual processing model, State-Transition Diagrams, Transition Graphs, State Charts and Petri Nets, Decision tables (Smaizys and Vasilecas, 2009), Data-oriented Object Constraint Language (OCL) (OMG, 2006; Ceponiene *et al.*, 2009), Entity-Relationship Diagrams, NIAM (Nijssen and Halpen, 1989) and the Behaviour Integrated Entity–Relationship (BIER) model (Eder *et al.*, 1987); Object-oriented Event schemas, object-flow diagrams and process-dependency diagrams offer a certain potential to specify some semantics of BR.

Generally, the graphical representations of BR for system of realistic size may become very complex and cumbersome (Grosf, 2001). Also they cannot be interpreted or used for further transformations in automated way. For this reason, several XML based specification languages like SBVR (OMG, 2008), SRML (Thorpe and Ke, 2001), etc. were introduced as a possibly useful alternative or supplement to graphical presentations.

Meta-model and template based business rule specifications provide a rigorous definition of aspects, which are not or extremely inconvenient to express in graphical notations. Such specification languages are especially suitable for a formalization of rules used for further automated transformation or use in inference processors of rule engine. However, the relevant interdependencies between single rules or rules and objects are preferably expressed in a graphical notation. The most recently used business rule specification and representation techniques are meta-model based business rule templates using XML, Decision Tables (DT) and decision trees.

3.2. Business Rule Classification and Management

A huge amount of BR used in enterprises and complexity of their system causes difficulties in management and further use. There are few known management approaches known as business rule repository (Butleris, 2004; Vasilecas *et al.*, 2006) and Business Rule Management System (BRMS) (Hall and Harmon, 2006; Morgan, 2002; Butleris and Nenortaitė, 2009). They are used to maintain business rule version and configuration control, search, classification for its combination into rule sets to service special needs like automated transformations or logical inference in rule engine, rule testing, simulation and execution or enforcement.

BR according to the BRG (2003) can be separated into two main groups according to the differences in syntactical structure: structural rules and operational rules. The Object Management Group (OMG) proposed semantic classification of BR introducing structural, behavioural and managerial BR and a set of rule types for each category (Weiden *et al.*, 2004). However from our point of view it is a variation of BRG classifier of the rules. OMG semantic classification may be probably used for better support of the elicitation of BR in application domain, but such complex classification is more difficult to manage because of the need of a large set of different syntactical structures and different graphical notations. It complicates specification process but transformation of BR to the production rules can be arranged better. In our projects for different purposes we are using some mixed variant of classification, specifying rules according to the BRG classification schema and assigning classes similar to OMG semantic classification schema to allow selection of the rules and combine them into the rule sets used for different transformation purposes and relations to the objects in other models like object structure and process models in design phase.

There are few of BRMS for rules management which can be provided like QuickRules (YASU Technologies, 2007), OpenRules (2008) and Versata (2006). Popular BRMS Rule Solutions for Office System and Rule Studio from ILOG¹ ILOG (2008) uses Microsoft SharePoint portal and Microsoft Visual Studio² technologies together with Rule Execution Server integrated with running Applications.

For the same purpose we are using Tools provided by OpenRules and provide version control with Turtoise SVN tool instead of Microsoft SharePoint portal differently from ILOG's case. Both ways of BR management are quite similar, because of the use of decision tables and XML.

Summarising BR in BRMS, BR is the intelligence resource of the meta-system and BRMS provides resources for such meta-system to carry out smart resolution tasks in the inference engine. It usually consists of a compiler and a BR representation (structured frame knowledge) facility. The compiler converts external knowledge representation, obtained through meta-system interfaces into internal formal representation forms available

¹ILOG, Rule Studio are registered trademarks of ILOG, Inc. All other company and product names are trademarks of their respective owners.

²Microsoft, Microsoft SharePoint portal, Microsoft Visual Studio are registered trademarks of Microsoft, Inc.

for storing in BR repository and for processing in inference mechanism using logical operations described in the next section.

3.3. Logical Operations with Business Rules

BR are dedicated to represent business knowledge also needed for automation of an organization's business and decision-making processes, removing to some extent influence of the human factor that is prone to inconsistency and error. We propose to distinguish four main purposes of logical operations with BR that depend on the purpose and component used for such operations: validation of BR set for consistency; decision automation using logical inference for decision evaluation according to the BR and facts derived from actual data in ERP system; intelligent business rule mining using flow engines able to discover new BR of facts by doing automated business data analysis of the business data stored in DBMS used in the ERP; and a kind of intelligent decisions using fuzzy logic dealing with situations when it is impossible to get any decision because of inconsistent BR set or availability of several alternative valid decisions.

At the moment there are several well-known logical derivation methods used in inference engines to process BR such as: Backward resolution also called goal-directed reasoning (Kusiak, 1997) and Forward-Chaining also known as Data-Driven reasoning (Kusiak, 1997), Rete inferencing algorithm (Forgy, 1982) designed by Dr. Charles Forgy during the late 1970 which found its way into expert system inferencing engines during the 1980s and inferencing engines of BRMS nowadays, and Fuzzy logic (Bremdal and Olsen, 1997). Backward resolution involves choosing hypothetical conclusions and testing to see if the necessary rules underlying the conclusions hold true and there are no contradictions in underlying rules. This method is usually used for BR set consistency validation. Forward-Chaining means following pathways through from facts to resulting conclusions, when inference engine infer further facts that may in turn cascading triggering of further rules, which may generate further facts; Forward-Chaining is used in ILOG JRules (ILOG, 2008) based on Java platform for integration with J2SE, J2EE, Eclipse and SOA. Rete algorithm is supported by many modern BRE and BRMS like QuickRules (2007), Jess (2008), JBoss Rules (2008) formerly Drools, etc. However Flow engines using Neural networks (Jain *et al.*, 1999) or rule extraction from time-series and deduction using statistical methods like Cluster analysis (Das *et al.*, 1998), etc., in the nearest future will enable extended functionality of intelligent BRMS.

A lot of the rule engines out there were originally called expert systems or inference engines, then they were called Business Rule Engines (BRE), and today they are known as BRMS. This determines that logical operations with BR sets in modern systems are usually processed and enforced using the internal inference engine or so-called BRE of BRMS, which carries out various actions based on the reasoning results, e.g., validates and transfers newly users designed BR directly from business system into BR repository and uses them for further transformation or decisions in automated business processes already implemented in ERP systems. Such integration of BR and BR based logical operations do not mean Expert systems going away. They are just going undercover of ERP or have vent already like former MRP, CRM and others did.

3.4. Business Rule Execution and Enforcement

Usually BR determines what action should be taken based on an enterprise internal guidelines, business practices and/or regulatory compliance. Such determined actions should be somehow executed in the software applications of dedicated ERP system.

Such way of rule enforcement can be achieved in several ways:

- BR model and related models transformation into the executable code of applications or active DBMS components;
- BR model (e.g., represented as DT) transformation into the parameters of the software system and enforcement by execution of dedicated methods of business objects representing business logic developed according to the SOA;
- BR enforcement in BRE by execution of predefined business processes or workflows.

In Vasilecas and Smaizys (2008) we also discuss possibilities of different BR enforcement level implementation in the software applications, because in some business scenarios such enforcement level may change depending on the business system state.

4. Business Rule Based ERP System Development Approaches

Packaged ERP applications have some embedded business rules technology. Rules in ERP packages usually cover basic business rules, but in case you have, for instance, a complex financial or legal scenario you would need to extend rules and enable flexible BR enforcement. This could be allowed in Business rule based ERP systems, leading to the need of selection of corresponding architecture, components, techniques and tools. On the base of our experience we could define the following different strategies and BR approaches based on ERP development maturity and used ERP engineering process complexity:

- **Separation** of business logic model in design process and process-driven BR implementation in the ERP system, typically using model transformations into the active DBMS component code dynamically or manually in the stage of system design.
- **Centralisation** of business logic execution in dedicated BR execution component using interpretation mechanisms of rule engine and BR enforcement in the integrated ERP systems activating built-in executable business processes for selected decisions.
- **Intellectualisation** of ERP system by implementation of intelligent algorithms to allow ERP system self adaptation involving automated business rule mining out from enterprise data and artificial intelligence.

Further in this section we will discuss all these approaches of BR based ERP system development and analyze their weak and strong sides according to the particular purposes.

4.1. *Separate Development of Business Logic Model and Integration into the ERP System*

There are several ways of business rule enforcement and implementation into the final software system proposed: to implement into relational database constraints (Zimbrao *et al.*, 2003), triggers (Valatkaite and Vasilecas, 2002; Sosunovas and Vasilecas, 2004) or executable code in applications using resolution of separately stored BR and facts representing entered value instead of validation code (Tang *et al.*, 2005; Vasilecas and Smaizys, 2008). Usually this approach could be used for development of one particular component of ERP system, and achieved by separate development of static business logic model used in ERP system for execution of structured decisions using software system parameterisation (Casati *et al.*, 2000) and later reuse of components developed (Kilov and Simmonds, 1996) and model driven service composition (Oriens *et al.*, 2003a, 2003b; Rosca *et al.*, 2002). This approach is similar and usually does not need the use of rule engine. However inference processing and involvement of some rule engine is useful for validation of BR combined into the rule sets used for further automated transformations. Summarising our experience we could define two different alternatives of such BR enforcement and separately developed business logic model implementation in ERP systems:

- ERP system parameterisation using BR and business logics usually represented in decision tables.
- Business rule and business process model automated transformation into the executable business processes and business logics exchange between separate parts of the business processes implemented into the several software applications later on.

Based on our research we can define following two approaches of BR based ERP systems engineering: Parameter driven approach and Model transformation – driven approach. In Butleris and Kapočius (2002) the authors distinguish the third one called Independent process-driven approach, which is similar to the ILOGs approach of BR based business process execution using BRE.

For illustration of parameter driven approach in Smaizys and Vasilecas (2009) we have analysed decision automation based on BR represented as DT for use by separate components servicing in business logic layer dedicated for selection of action and execution of related executable code in application layer or database layer of software systems.

Model transformation – driven approach was examined in Vasilecas and Smaizys (2007), where we have introduced a method based on a framework early presented in Vasilecas and Smaizys (2005) for development of XML based web applications, using BR model transformations into the design specifications used for further development and/or validation, interpretation and transformation of BR sets into the executable code.

The main difference of our approach is that for example ILOG uses separate Rule Execution Server and we dedicate this functionality to the separate software system component according to the SOA by automated generation of business layer objects and direct deployment of BR represented in DT to the Application database. This approach is simpler and can be easy understand by IT people. ILOG's approach is more suitable for large applications based on Microsoft technology, Microsoft Windows Workflow Foundation

and BizTalk (ILOG, 2008). Our approach is based on open source software and is integrated with Microsoft Visual Studio for the use by software developers. We do not require the use of BizTalk or another separate rule execution server and use automatically generated dll's with business layer objects providing execution of business logic according to the rules stored in DBMS. The same shared dll's in software system layer are used for integration of several applications in the Enterprise to implement shared logics. We argue that structured rules in DTs representing decisions required are simple and it is enough to automate development of querying process in the software system according to the BR managed at the business system level. Although for more sophisticated and intelligent decision making methods based on statistical analysis (e.g., clustering), neural networks, risk models, fuzzy logic, etc. the use of separate rule execution server is unavoidable.

Judging from the works of Avdejenkov *et al.* (2009) we can draw a conclusion that the business logic exchange here is vital for easy and fast ERP modification and adaptation to the business changes. ILOG's approach to business logic exchange is achieved by using separate Rule management tools and Rule Execution Server. Such rules or actions according to ILOG are exchanged through BizTalk server, where transformations needed are implemented in a special tool ILOG (2008). Our approach is related with the use of similar XML based source to destination meta-model based transformations with development of special transformation instructions stored as XML transformation schema (Vasilecas and Smaizys, 2007). Another approach alternative to the business logic exchange achieved by centralization of business logic execution presented in the next section.

4.2. Centralized Business Logic Execution and Business Rule Enforcement in Integrated ERP Systems

There are strong arguments for a non-redundant and centralised implementation of the IS-relevant BR in the database. This approach has the additional advantage that the rules cannot be circumvented by operations on the DBMS because of triggers and stored procedures are used for rule integration (Badawy and Richta, 2002). However the business logic used in business processes is distributed through different information systems software and usually differs depending on technology used for each particular software system. This leads to the problems dealing with inadequate automated decision processing and implementation of changes into different software or several parts of ERP system. This can be avoided by using centralized business logic execution and BR enforcement in the integrated ERP system according to the framework presented in Avdejenkov *et al.* (2009).

The main advantage of this approach is that here the part of software dedicated to the business decisions together with the rules describing business logic is separated and implemented into the central component of BRMS. This component is responsible for business decisions and includes a model of decision propagation into the components of enterprise software through several available information system interfaces (Fig. 3). The central part of proposed framework includes some reasoning processor for business rule execution. For example, rule execution engine also called BRE carries out various actions

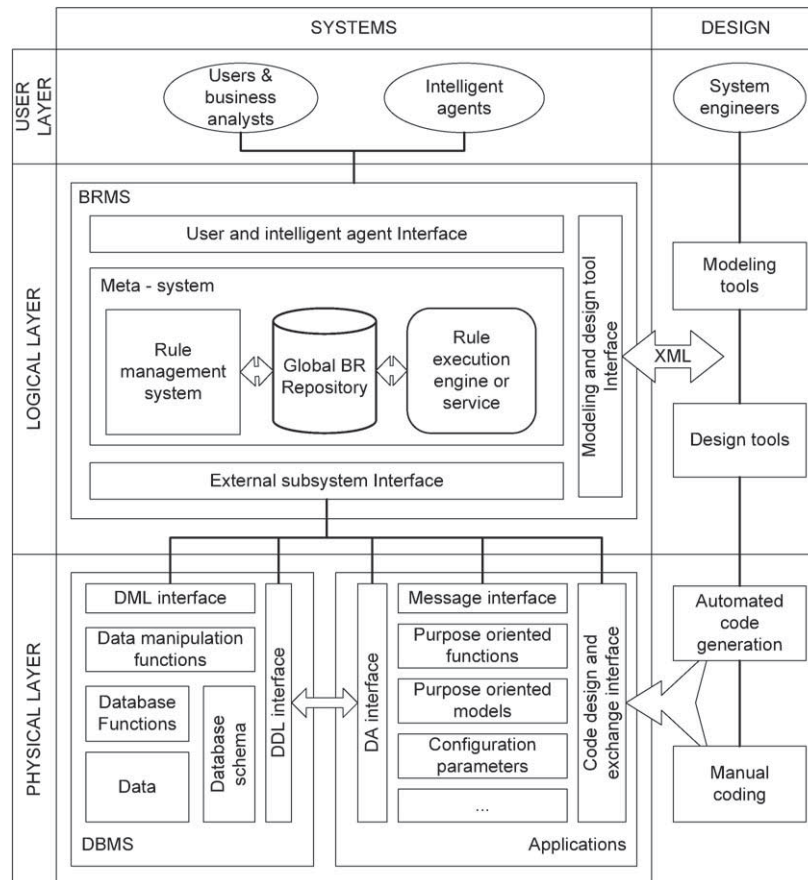


Fig. 3. The Framework of integrated intelligent system using business rule management system.

based on the reasoning results in the internal inference engine, ensuring data transferred between any two subsystems to conform according to the BR and data entered by users representing facts.

The interface to the external systems builds the communication between business users, internal BRMS system components and external software systems. The emerging agent software can be used for automation of BR discovery as well. The interface plays a key role in an open structured BRMS software system: codify human expertise into the software system in such a way that it can adopt the most creative intelligence and knowledge in decision making, transform formally represented BR into human language for review and edit by users, communicate with other external intelligent software system agents for more complicated tasks. BR based meta-rules concentration in one BRMS and rule exchange interfaces support intelligent interaction between different subsystems in an enterprise. Also, due to a flexible XML language used for BR formalization, it allows introducing an open organisation structure with a new intelligent functionality. Such a BR based meta-knowledge base system creates an open organization structure and allows

a new intelligent functionality to enter the meta-knowledge and use it in all subsystems with corresponding rule exchange and enforcement interfaces to engage more duties in decision making process.

BRMS in this context allows not only integration of new services and business logic into the existing applications at the physical layer but also allows indirect integration of applications through BRMS or DBMS, as well (Chorafas, 2001; Peña *et al.*, 2002). According to our proposed framework, the integration possibilities are limited only by architecture of the application and functionality of external communication interfaces of DBMS and applications. DBMS usually include at least two interfaces for message exchange in Data manipulation and description languages, represented as Data Manipulations Language (DML) and Data Definition Language (DDL) interfaces in the framework.

4.3. *Intelligent Self Adaptable ERP Systems*

Nowadays information system development is experiencing transition from crisp logic to fuzzy logic. This allows for both algorithmic and heuristic problem solving. Currently, there is a considerable research being done on Fuzzy Logic Relational Data Bases (FRDB) and Fuzzy Logic Query Languages (FSQL). Authors introduce fuzzy attributes, fuzzy constraints, and creation of FuzzyEER model with focus on semantic aspects in conceptual design (Galindo *et al.*, 2006). Fuzzy logic was defined in the late sixties of XXth century and enabled representation of more complicated rules with implication functions which were impossible using binary logic, e.g., decision systems, where a decision is dependent on large amount of factors.

During our research we have challenged a few problems not discussed before related to the situations, when BR are not known at the beginning or they are incomplete even contradicting but we still need decisions. Moreover such BR systems are not static and should be changed when a business situation changes. This means the need of development of the software systems that could be self adaptable to such changes and ensure information system adaptation to maximise achievement of goals dedicated by the executive staff. Such situations may be resolved by involvement of fuzzy logic or flow engines implemented in BRMS component called BR mining engine. We have made some experiments for solution of such problem by building a risk model of Vasilecas and Smaizys (2008). Similarly in Graeme *et al.* (2003) author uses statistical methods for simulation of decision influence on future state of business system. Such approach allows dynamical adaptation of the user interface of the ERP application to the changes in business system observed from the collected enterprise data and generation of the business service and results in the extensions or modifications to the workflow logic, business logic or data logic being directly reflected in the operative user front-end and the presentation logic of ERP system.

Summarising intelligent features of ERP systems that we can get from the approaches described in previous sections – they usually are acting by structured rules and do not deal with every possible impact on the business environment or future consequences. That is

the main reason why automated decisions based on such rules cannot take responsibility and requires involvement or approval of dedicated business people. This limits decision automation possibilities. The use of fuzzy logic instead or together with crisp logic in ERP systems can be used to simulate a real business environment and evaluate possible impact providing automated heuristic decisions.

5. Discussions

In this section we want to debate typical questions that may arise during discussions on the subject of this paper.

- **Why there are so many different BR approaches for the ERP system development?**

We argue that there are not widely accepted standards or universal BR approach for ERP development yet. This leads to a large scale of different methods and variations of systems engineering processes and architectures used. In this paper we have made efforts to group them into the three main groups according to the maturity level, complexity and components used to ensure the agile ERP system development. This started from a simple involvement of BR and separate management according to the first approach described in Section 4.1, centralised BR execution and enforcement involved in the second one which is described in Section 4.2 and intelligent way of BR mining and use for self adaptable ERP system development described in the third approach presented in Section 4.3. However this set is not final one and we expect new evolving methods of ERP systems development in the nearest future, especially those related to the intelligent self adaptable ERP systems. Every further approach is more complicated and the selection of the approach used should be made according to the particular needs of ERP systems developers and business users.

- **What about propagating changes of ERP systems data model when BR system has been changed?**

We agree that there are some limitations for modification of the existing data model. Especially, when using parameterisation approach in development process. However according to our method proposed in Smaizys and Vasilecas (2009) it is allowed to change the existing data model and we allow addition of new rules, conditions and actions, by propagating them directly into the data model of configuration part of the software representing BR based business logic by running automated transformations and executing DDL instructions in the new iteration of development process. Automated refresh of logical model at the business system layer, update and automated generation (refresh) of the code at business logic tier according to the n -tier architecture is also required. However, we experienced limitation assuring compliance of already existing business data records to the new BR and still we can not deal with changes in existing attributes and relations of business entities being propagated in an automated way.

- **Why we are presenting included ideas as important to ERP systems development only?**

We agree that most of the presented information could be applied to the development of intellectualized IS in general. That is because as we have noticed before – the border between ERP systems and other kind of IS is going to be heavily distinguished. However the ideas, methods and results presented in this paper are based on our running researches in the field of ERP systems only. Wider application of the methods presented in this paper for intellectualized IS in general should be produced with care and checked accordingly.

6. Conclusions

In this paper we have overviewed BR based ERP systems engineering process and separated three main approaches used for such a ERP systems development based on maturity and complexity of engineering processes and methods used. Depending on the approach used for BR based ERP systems development it may be possible to ensure different level of agility by an instant deployment of changes in the Business policy and immediate reaction to the changes on the market or competition by changing existing business rules and introducing new rules not by programmers, but by business analysts. Such advances allow ERP systems to be more transparent, auditable and to achieve cost reduction because of more efficient process of introduction of changes in business policy into the software systems of BIS used for implementation of decision automation and decision support.

Current commercial ERP systems are developed using traditional BIS engineering methods and only few of them are on the first – BR separation level of maturity and complexity according to the BR based ERP approach classification proposed. Basically business rules are used only for description of business logics in business processes modelled and implemented in traditional way.

Although BR based ERP systems have more complex development process in an initial phases, but such a system is more efficient in further maintenance and simplified modifications that is especially needed for businesses with frequently changing regulations and business policy, competitive behaviour on the market and requiring a high level of customisation and adaptation to the large scale of separate customer needs, especially when the software is designed using SOA.

Because of complexity of BR approach and no existing standardised methods of BR based ERP systems development current systems still not include any centralised component acting as BRMS. This prevents the use of more complicated and mature BR based approaches of ERP systems development. It still may be only partly achieved using separate additionally developed software applications, which ensure automation and execution of business processes responsible for decisions according to the business logics represented in BR. Such extensions usually are integrated with ERP systems using all the available standard ERP system interfaces or special messages passed through business process management server.

There are more problems left to solve in order to start a wide-ranging development of BR based intelligent ERP systems. Several early businesses oriented and ERP integrated products using business rules technology have their roots in the realm of artificial intelligence and inference systems. They were complex, expensive to run and maintain, and not business-user-friendly. However some modern ERP system developers (e.g., Oracle, etc.) have already started implementation of intelligent parts of the software systems using BR approach.

Acknowledgements

The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)" Reg. No. B-07042.

References

- Avdejenkov, V., Vasilecas, O., Smaizys, A. (2009). Business rule management in enterprise resource planning systems. In: *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems V*, Vol. 16. IOS Press, pp. 255–266.
- Badawy, M., Richta, K. (2002). Deriving triggers from UML/OCL specification. In: Kirikova, M. *et al.* (Eds.), *Proc. of ISD 2002 conference on Information Systems Development: Advances in Methodologies, Components and Management*. Kluwer Academic/Plenum Publishers, pp. 305–316.
- Bajec, M., Krisper, M. (2005). A methodology and tool support for managing business rules in organisations. *Information Systems*, 30(6), 423–443.
- Bocij, P., Chaffey, D., Hickie, S., Greasley, A. (2005). *Business Information Systems*. Pearson Education.
- Bremdal, B.A., Olsen, S. (1997). An introduction to fuzzy systems in production and mechanical engineering. In: Wang, K. *et al.* (Eds.), *Nordic-Baltic Summer School on Applications of AI to Production Engineering*, Kaunas University of Technology Press, Kaunas, pp. 115–129.
- BRG (2000). *Defining Business Rules – What Are They Really?* Version 1.3. Ross, G.R. (Ed.), Business Rules Group (BRG). http://www.businessrulesgroup.org/first_paper/br01c0.htm.
- BRG (2003). *The Business Rule Manifesto – The Principles of Rule Independence*. Version 2.0, Ross, G.R. (Ed.), Business Rules Group (BRG). <http://www.businessrulesgroup.org/brmanifesto/BRManifesto>.
- BRG (2005). *The Business Motivation Model – Business Governance in a Volatile World*. Version 1.2, Ross, G.R. (Ed.), Business Rules Group (BRG). <http://www.businessrulesgroup.org/bmm.shtml>.
- Butleris, R., Kapočius, K. (2002). The business rules repository for information systems design. In: *6th East-European Conference ADBIS'2002. Research Communications*, Vol. 2. STU, Bratislava, pp. 64–77.
- Butleris, R., Nenortaitė, J. (2009). Improving business rules management through the application of adaptive business intelligence technique. *Information Technology and Control*, 38(1), 21–28.
- Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.C. (2000). *Adaptive and Dynamic Service Composition in eFlow*. HP Lab. Techn. Report, HPL-2000-39. <http://www.hpl.hp.com/techreports/2000/HPL-2000-39.html>.
- Ceponiene, L., Nemuraite, L., Vedrickas, G. (2009). Separation of event and constraint rules in Uml&Ocl models of service oriented information systems. *Information Technology and Control*, 38(1), 29–37.
- Chisholm, M. (2002). The black box problem. *Business Rules Journal*, 3(3). <http://www.BRCommunity.com/a2002/b100.html>.
- Chisholm, M. (2004). *How to Build a Business Rules Engine*. Morgan Kaufman Publishers.
- Chorafas, D.N. (2001). *Integrating ERP, CRM, Supply Chain Management, and Smart Materials*. CRC Press LLC, Auerbach.

- Chung, C.H. (2003). Operation management. In: *Encyclopedia of Information Systems*, Vol. 3. Elsevier Science, pp. 391–402.
- Ciuksys, D., Caplinskas, A. (2007). Reusing ontological knowledge about business processes in IS engineering: Process configuration problem. *Informatica*, 18(4), 585–602.
- Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P. (1998). Rule discovery from time series. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI Press, pp. 16–22.
- Date, C.J. (2000). *What Not How: The Business Rules Approach to Application Development*. Reading, MA, Addison-Wesley.
- Eder, J., Kappel, G., Tjoa, A.M., Wagner, R.R. (1987). BIER – The behavior integrated entity-relationship approach. In: Spaccapietra, S. (Ed.), *Proceedings of the 5th International Conference on Entity-Relationship Approach*. North-Holland, Amsterdam, pp. 147–166.
- Eriksson, H-E., Penker, M. (2000). *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons.
- Forgy, C.L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1), 17–37.
- Galindo, J., Urrutia, A., Piattini, M. (2006). *Fuzzy Databases: Modeling, Design, and Implementation*. IGI Publishing, Hershey, PA.
- Graeme, S., Seddon, P.B., Willcocks, L. (2003). *Second-Wave Enterprise Resource Planning Systems*. Cambridge University Press.
- Grosz, B.N. (2001). Standardizing XML rules: preliminary outline of invited talk International Joint Conference on Artificial Intelligence (IJCAI-01). In: Preece, A. (Ed.), *Workshop on E-Business and the Intelligent Web*. <http://xml.coverpages.org/IJCAI-groszof20010518.pdf>.
- von Hale, B. (2001). *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. John Wiley & Sons, New York.
- Hall, C., Harmon, P. (2006). *The 2006 BP Trends Report on Business Rules Products, Business Process Trends*. www.bptrends.com/reports_toc_03.cfm
- Hay, D.C. (2003). *Requirement Analysis: From Business View to Architecture*. Prentice-Hall, New Jersey.
- Hay, D.C., Kolber, A., Anderson, H.K. (1997). *Guide Business Rule Project: Final Report*. <http://grace.evergreen.edu/businessRules>.
- Herbst, H. et al. (1994). The specification of business rules: A comparison of selected methodologies. In: Verrijn-Stuart, A.A., Olle, T.W. (Eds.), *Methods and Associated Tools for the Information System Life Cycle*. Elsevier, pp. 29–46.
- Hossain, L., Patrick, J.D., Rashid, M.A. (2002). *Enterprise Resource Planning: Global Opportunities & Challenges*. Idea Group Publishing.
- ILOG (2008). *Business Rules Management System ILOG Rules*. <http://www.ilog.co.uk/products/rules/>.
- Jain, A.K., Murty, M.N., Flynn, P.J. (1999). Data clustering: A review. In: *ACM Computing Surveys (CSUR)*, Vol. 31, Issue 3. ACM, New York, USA, pp. 264–323.
- JBoss (2008). *Business Rule Engine JBoss Rules*. <http://www.jboss.com/products/rules/>.
- Jess (2008). *Rule Engine Jess (Java Expert System Shell)*. <http://herzberg.ca.sandia.gov/jess/>.
- Kilov, H., Simmonds, I. (1996). Business patterns: Reusable abstract constructs for business specification. In: Humphreys, P. et al. (Eds.), *Implementing Systems for Supporting Management Decisions: Concepts, Methods and Experiences*. Chapman and Hall, pp. 225–248.
- Kusiak, A. (1997). Knowledge-based systems. In: Wang, K. et al. (Eds.), *Nordic-Baltic Summer School on Applications of AI to Production Engineering*. Kaunas University of Technology Press, Kaunas, pp. 45–79.
- Loucopoulos, P., Katsouli, E. (1992). Modelling business rules in an office environment. *SIGOIS Bulletin*, 13(2), 28–37.
- Mahajan, S. (1997). Building a data warehouse using Oracle OLAP tools. Oracle technical report. *ACTA Journal*, September. <http://www.oracle.com>.
- Morgan, T. (2002). *Business Rules and Information Systems: Aligning IT with Business Goals*. Addison-Wesley.
- Nijssen, G.M., Halpin, T.A. (1989). *Conceptual Schema and Relational Database Design – A Fact Oriented Approach*. Prentice-Hall, Englewood Cliffs.
- OMG (2006). *Object Constraint Language*. Version 2.0. Object Management Group (OMG). <http://www.omg.org/cgi-bin/doc?formal/2006-05-01/>.

- OMG (2007). *Unified Modeling Language*. Object Management Group. <http://www.uml.org/>.
- OMG (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, Version 1.0 (December). <http://www.omg.org/docs/formal/08-01-02.pdf>.
- OpenRules (2008). *Business Rules Management System Open Rules*. <http://openrules.com>.
- Orriens, B., Yang, J., Papazoglou, M.P. (2003a). Model driven service composition. In: *Lecture Notes in Computer Science*, Vol. 2910. Springer-Verlag, Berlin, pp. 75–90.
- Orriens, B., Yang, J., Papazoglou, M.P. (2003b). A framework for business rule driven service composition. In: *Lecture Notes in Computer Science*, Vol. 2819. Springer, Berlin, pp. 14–27.
- Penya, Y.K., Bratoukhine, A., Sauter, T. (2002). ERP integration into generic plant automation model. In: *Workshop on Information Systems for Mass Customization, ISMC'2002*, Sep. 2002, Málaga Spain, pp. 137–148.
- QuickRules (2007). *Business Rules Management System QuickRules*. <http://www.yasutech.com>.
- Rosca, D., Greenspan, S., Wild, C. (2002). Enterprise modelling and decision-support for automating the business rules lifecycle. *Automated Software Engineering*, 9(4), 361–404.
- Ross, R.G. (1997). *The Business Rule Book. Classifying, Defining and Modeling Rules*. Business Rules Solutions. LLC Houston.
- Ross, R.G. (2003). *Principles of the Business Rule Approach*. Addison-Wesley.
- Smaizys, A., Vasilecas, O. (2009). Agile software system development and customisation using business rules. In: *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems V*, Vol. 187. IOS Press, pp. 243–254.
- Sosunovas, S., Vasilecas, O. (2004). Transformation of business rules models in information systems development process. In: *Databases and Information Systems Doctoral Consortium: Sixth International Baltic Conference BalticDB&IS 2004*, Riga, Latvia, June 6–9, Vol. 672. University of Latvia, pp. 373–384.
- Tang Z., Maclennan, J., Kim, P.P. (2005). Building data mining solutions with OLE DB for BM and XML for analysis. *SIGMOD Rec.*, 34(2), 80–85.
- Thorpe, M., Ke, C. (2001). *Simple Rule Markup Language (SRML): A General XML Rule Representation for Forward-Chaining Rules*. ILOG. <http://xml.coverpages.org/srml.html>.
- Valatkaite, I., Vasilecas, O. (2002). Deriving active database triggers from business rules model with conceptual graphs. In: *Lithuanian Mathematical Collection*, Vol. 42. MII, Vilnius, pp. 289–293.
- Valatkaite, I., Vasilecas, O. (2005). On business rules automation: The BR-centric IS development framework. In: Eder, J. et al. (Eds.), *ADBIS 2005, LNCS 3631*. Springer-Verlag, Berlin, Heidelberg, pp. 350–365.
- Van Assche, F., Layzell, P., Loucopoulos, P., Speltincx, G. (1988). Information systems development: A rule-base approach. *Journal of Knowledge Based Systems*, 1(4), 227–234.
- Vasilecas, O., Lebedys, E. (2006). Moving business rules from system models to business rules repository. *INFOCOMP*, 5(2), 11–17.
- Vasilecas, O., Smaizys, A. (2005). Business rule based knowledge base integrated intelligent system framework. *Information Sciences*, 34, 195–200.
- Vasilecas, O., Smaizys, A. (2007). The framework for business rule based software modeling: An approach for data analysis models integration. In: *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, Vol. 155. IOS Press, pp. 175–188.
- Vasilecas, O., Smaizys, A. (2008). Business rule enforcement and conflict resolution using risk analysis. In: *Proceedings of the 14th International Conference on Information and Software Technologies (IT2008), Research Communications*. Kaunas University of Technology, Kaunas, pp. 92–98.
- Versata (2006). *Business Rules Management System Versata*. <http://www.versata.com>.
- Weiden, M., Hermans, L., Schreiber, G., Sven van der Zee (2004). *Classification and Representation of Business Rules*. www.omg.org/docs/ad/02-12-18.pdf
- YASU Technologies (2007). *Business Rules Management System QuickRules*. <http://www.yasutech.com>.
- Zachman, J. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3), 276.
- Zachman, J., Sowa, J. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, 31(3), 590.
- Zimbrão, G., Miranda, R., Souza, J.M., Estolano, M.H., Neto, F.P. (2003). Enforcement of business rules in relational databases using constraints. In: *Proceedings of XVIII Simposio Brasileiro de Bancos de Dados/SBBD 2003*. UFAM, pp. 129–141.

A. Smaizys has graduated from Klaipėda University in 2004 and received the MS degree in informatics. From 2007 he is a PhD student at Vilnius Gediminas Technical University, Faculty of Fundamental Sciences. He is a lector in Information System Department of the Klaipėda University. His research interests are in business rule based information system engineering.

O. Vasilecas has graduated from Kaunas University of Technology in 1967. He received the PhD degree from Vilnius University, in 1979. He is a professor of Information System Department, and the head of the Information System Research Laboratory in the Vilnius Gediminas Technical University. His research interests include intelligent information system, business, information and software system engineering, knowledge representation and formal concepts modelling, and business rules approach in information systems development.

Lankstus ERP sistemų kūrimas naudojant verslo taisykles

Aidas ŠMAIŽYS, Olegas VASILECAS

Pastarąjį dešimtmetį kilęs susidomėjimas verslo taisyklėmis lėmė naujų informacinių sistemų kūrimo metodų atsiradimą. Verslo taisyklės buvo pradėtos vis dažniau ir dažniau naudoti lanksčių (angl. Agile) informacinių sistemų kūrimui. Tačiau jų taikymas kuriant įmonės resursų planavimo (angl. Enterprise Resource Planning – ERP) sistemas išlieka visiškai nauja taikymo sritimi, teikiančia didelių vilčių siekiant ženkliai pagerinti kokybines ir kiekybines šios rūšies informacinių sistemų charakteristikas. Šiame straipsnyje apibendrinami žinomi verslo informacinių sistemų kūrimo metodai leidžiantis panaudoti verslo taisykles, nagrinėjamos su taisyklių taikymu susijusios problemos, pristatomi nauji verslo taisyklių taikymo būdai leidžiantys užtikrinti ERP sistemų kūrimo lankstumą, paremti straipsnio autorių vykdomų šios srities tyrimų rezultatais.