ISSUES AND CHALLANGES IN BUSINESS RULE-BASED INFORMATION SYSTEMS DEVELOPMENT

Bajec, Marko, University of Ljubljana, Faculty of Computer & Information Science, Trzaska 25, 1000 Ljubljana, Slovenia, marko.bajec@fri.uni-lj.si

Krisper, Marjan, University of Ljubljana, Faculty of Computer & Information Science, Trzaska 25, 1000 Ljubljana, Slovenia, marjan.krisper@fri.uni-lj.si

Abstract

An explicit manipulation of business rules in information systems development (ISD) is an old domain. There were many attempts in the last two decades to define how the rules should be dealt with throughout the ISD activities. Despite many results that have been achieved, several questions regarding business rule manipulation within ISD remain unresolved and present challenges for future research. The objective of this paper is to discuss these challenges and where possible to point out some directions for potential solutions.

Keywords: business rules, business rule approach, business rule-based information systems development

1 INTRODUCTION

Business rules have received a lot of attention in the information system (IS) community. The credit for this goes to their ability to make applications flexible and amenable to change. Both researchers and practitioners are convinced that business rules require explicit treatment during IS development (ISD) to ensure the IS agility (Herbst 1996, Petrounias and Loucopoulos 1994, Bajec and Krisper 2004, Struck 1999, Youdeowei 1997, Moriarty 2000, Barnes and Kelly 1997, Date 2000, Gottesdiener 1999, Moriarty 1993, Sinur 2003). Otherwise many problems may occur. For example:

- Since not acquired systematically and completely, business rules do not reflect real conditions of the business environment. Consequently, applications are developed that do not meet business needs.
- There is a lack of documentation on business rules. Business rules are not traceable to their sources. The motivation for business rules in unclear.
- Business rules are buried into program code. It is not clear, what kind of rules govern an application, when the rules are triggered and how they are implemented.
- Business logic is hard to maintain as rules are distributed across the application logic.
- Business rules are hard to control, since there is no common and single purpose store for them.

The problems that are listed above have initiated a lot of research in academia and industry. A lot of work has been done to understand the rationale behind business rules and to define the way how business rules should be manipulated throughout the entire systems development lifecycle (Bajec and Krisper 2004, Assche et al 1988, Herbst 1996, Petrounias and Loucopoulos 1994, Struck 1999, Youdeowei 1997, Bajec 2001, Hay and Healy 1997). Various tools have been developed offering wide scope of functionality to support the management of business rules[1]. Nevertheless, in ISD practice business rules are rarely used in a way as defined by the so called *business rule approach*[2], where they are separated from components that are not relevant to the business knowledge, documented so that they are traceable from/to their sources, and positioned so that they are easily changeable and adaptable. Furthermore, there is only limited support for an explicit business rule manipulation in today most popular development environments and CASE tools (Bajec and Krisper 2004). It seems like the business rule approach is propagated only by a certain number of consultant companies that sell their own tools and methods. If this is true, then what are the obstacles that hinder the use of the business rule based ISD in practice?

The objective of this paper is to discuss some of the questions regarding the business rule approach that in our opinion remain unanswered and thus present challenges for future research. The paper presents literature review but also our own view on the subject area, which was shaped through personal experiences in the business rule-based ISD.

The paper is organised as follows. Section 2 gives a brief history of business rule related research, discussing contributions from different research areas. The core part of the paper is than presented in section 3. After a short introduction into the research scope and objectives five business rule-related problem areas are identified. Each problem area is than discussed in subsequent sections. Finally, a conclusion is given.

---

[1] For the list of tools see Struck (1999). Acomparision of tools is given in Bajec and Krisper (2004).

[2] The term "Business rule approach" has been coined by a group of consultants who have established the business rule community. According to their definition (Ross 2003), the Business Rules Approach is "a combination of existing and new techniques and technologies in order to identify the rules required to run a business, to document these rules, to reason with it, to make them operational in a consistent way, to systematically adapt it to ever changing market forces and to automate the enforcement of the rules where possible". For the purpose of this article we use this term to designate the approach to ISD in which business rules are separated from components that are not relevant to the business knowledge, documented (sources and usages of every of business rule), made explicit, and positioned, so that they are easily changeable and adaptable (Von Halle 2001).

2 BUSINESS RULES BACKGROUND

The roots of business rules come from the field of Artificial Intelligence, where they have been successfully applied as a way of representing knowledge. In the knowledge-based systems the knowledge and reasoning of a human expert can be captured and stored in a form of complex networks of rules. The rules are typically described using declarative languages that do not imply order or flow of control. The rules are stored in a Rule base and processed in a special component, called Inference engine. The inference engine evaluates the conditions of the rules and at any point in time determines which ones are eligible to fire.

An extensive history of the business rule related research could also be found within the Database research community. As a result of this effort active databases had emerged (Morgenstern 1983, Schlesinger 1987, Dayal 1988, Widom and Ceri 1996, Tanaka 1992), which, as opposed to passive databases, employ active components such as triggers and database procedures to perform their own data integrity functions. Another database related research on rules addresses deductive databases (Gallaire et al 1984, Minker 1988, Petrounias and Loucopoulos 1994). While traditional database systems only manage extensional knowledge that is embedded in facts and instances, deductive databases add intentional knowledge which is beyond the factual content of the database. This kind of knowledge can be fully specified with rules and is stored in a rule base before the database is established (Elmasri and Navathe 1989).

Once the idea of implementing business rules in database systems emerged, substantial effort was put into discovering a robust and powerful method for the representation of business rules in data models (Herbst 1994, Loucopoulos 1991, Ross 1997, Tanaka 1992, Nijssen and Halpin 1989). As opposed to static business rules that can be expressed in Entity-Relationship-Models (ERM), dynamic business rules are not supported, as ERMs do not allow an explicit representation of events, conditions or actions. Consequently, several extensions to the ERM have been proposed (e.g. ER-RM (Tanaka et al 1991) or BIER (Elder et al 1987) as well as other techniques and methods (e.g. ELH of SSADM (Downs et al 1992) or the Ross Method (Ross 1997). A comparison of selected methods can be found in (Herbst et al 1994).

As interest in business rules grew, advocates of the approach became aware that an explicit manipulation of business rules was required in order to support the entire business rule life cycle. Numerous research projects have been carried out in support of business rule discovery, analysis, modelling, classification, articulation, formalisation and documentation (c.f. Hay and Healy 1997, Youdeowei 1997, Ross 1997, Herbst 1997, Herbst 1996, Loucopoulos 2000, Ceri and Fraternale 1997). Significant contributions regarding the rule-based paradigm have been received from ESPIRIT-projects RUBRIC, TEMPORA and F3 (From Fuzzy to Formal).

Business rules also appeared in the Object-oriented community. While it seems that proponents of the object-oriented approaches share opinion that business rules deserve attention, they still disagree on where to put them in object-oriented models (Diaz et al 1998, Danny 1999, Greenspan et al 1982, Struck 1999). Some believe that since objects are responsible for their own data and behaviour, business rules should be modelled in the object/class models as properties of classes. Others are trying to achieve the synergy by merging numerous paradigms, including business rules. Nilsson (1999) points out that the fact that rules are spread and converted into methods is one of the weakest points in object orientation. Although the recently standardized UML (Unified Modelling Language) offers elaborative meta-model, it does not provide much guidance for modelling business rules. The only usable element for expressing business rules is a constraint element, which can be attached to any other element. The OCL (Object Constraint Language), which is provided for detailed specifications of constraints, seems useful at a design level; however it does not prove that well in systems requirements analysis when working directly with business people (Gottesdiener 1999).

Business rule-related research remains active also in present days. Besides few centres of business rule exploration and development in academic environment, there is also high interest in industry. One of

the recent novelties that impacts on how business rules are manipulated in business environments is the penetration of the tools that employ rule engines to process rules. These tools follow the paradigm of expert systems. They are often based on results of a rich heritage of expert system technology. They are typically versatile tools offering a wide variety of features for business rule management at business and IS level. According to Gartner Group, their prospects for the future are more than promising. In conjunction with business process management technology, which is pervading many markets today, their success is expected to continue (Anthes 2003, Sinur 2003).

3 ISSUES AND CHALLENGES FOR THE FUTURE RESEARCH

For the purpose of our work we have speculated that the ultimate goal of the business rule research is to find a way and facilities that would support automatic propagation of business changes from business environment to IS and its application software. This would help to bridge the gap between business and IS. Aligning IS and business operation is one of the fundamental problems in all organisations, which still has to be resolved effectively. To reach that goal, it has to be clear how business rules should be dealt with in each phase of the ISD lifecycle. A clear foundation is required at least for the following topics:

- **The scope of business rules**: What exactly are business rules?
- **The acquisition of business rules**: How to acquire business rules from business objectives and business people? How to extract business rules from legacy code?
- **The specification of business rules**: How to represent business rules so that they will be understandable to both, business and IS people? How to model business rules?
- **The implementation of business rules**: How to implement business rules. What technology to use? What is the best place for business rule implementation (to minimize the effort required to change business rules)?
- **The management of business rules**: Where to store business rules (including their context)? How to maintain business rule changes and versions? How to manage business rules for an entire organisation?

In the rest of the paper we will address each of the above topics in detail. We will try to identify the issues that we believe are still unresolved and thus present challenges for future research towards the achievement of the ultimate goal.

3.1 What exactly are business rules?

Despite the fact that business rules receive a lot of attention from both, industry and academia, there is still no common or widely accepted definition of what exactly business rules are. Many authors have tried to clearly and concisely define the business rule concept. Let's see some examples:

- *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*. (Hay and Healy 1997).
- *A business rule is an explicit statement stipulating a condition that must exist in business information environment for information extracted from the environment to be consistent with business policy*. (Appelton 1994).
- *A business rule is a rule stating something which impacts the business of concern, and the interpretation of the rule may heavily impact the quality of the information system to be developed*. (Selveith 1991).
- *Business rules respond to application needs; they model the reaction to events which occur in the real world, with tangible side effects on the database content, so as to encapsulate the application's reactive behaviour to such events*. (Ceri and Fraternale 1997).
- *Business rules are requirements that arise from the business objectives of an enterprise*. (Rosca et al 1995).

- *Business rules correspond in some way to an organisation's economic activity. They may manifest themselves in all possible patterns that effect organisation – organisational, business, service, and information patterns*. (Morabito et 2001).

It is clear from the above definitions that the idea of a business rule is still not represented as a clear concept. One of the frequently held misconceptions about business rules – typically present among developers – is that all the rules governing an application are business rules. The relationship between business rule and business is emphasised in almost all definitions but in practice we can find many instructions in programme code that developers call business rules even though they have no correlation with business. Design instructions are a good example of such rules. Here are some examples:

```
"If the total sum is greater than 1.000 EUR then colour the result in
BLUE."
```

```
"If the purchase value is greater than 2500 EUR then display the message
'You have become a gold customer'"
```

Design rules are based on design decisions and do not derive from business world. In other words, for business people these rules do not exist at all (except if they were involved in the system design). From the point of view of an IS developer the situation is different. As stated by Ross (1988) business people treat business rules in a different manner than developers. For business people business rules are directives that are intended to influence or guide business behaviour. Developers, on the other hand, tend to see business rules as atomic pieces of program logic that are typically in "IF THEN" form. It does not mater whether the source of such rules derives from business or not. This is the reason why there are many rules in a so called "business logic" that have no source in business policy.

The bottom line here is that we have to make clear what are the rules that we are interested in? From the ISD perspective all the rules that are dynamic in nature and thus present potential problems for the IS maintenance need appropriate attention. Not just business rules that derive directly from business, but also other rules that emerge in the process of the IS development. In our opinion the business rule research requires wider scope. Our suggestion is to extend the business rule research on rule research where all the rules that are exposed to changes will be included.

3.2 Business rule acquisition

The acquisition of BR is not an easy task as many rules are difficult to identify. In particular this holds for the rules that have no explicit representation. Depending on their information contents, business rules can be based on either explicit or tacit knowledge. Explicit knowledge is formalised knowledge that is easy to express in form of principles, procedures, facts, figures, rules, formulas, etc. Contrariwise, tacit knowledge is not easily expressed and visible (Nonaka 1991). But when information contents of a business rule correspond to sufficiently deterministic behaviour, the rule takes the form of explicit knowledge. Such are, for example, rules that govern important operations, like customer credit approval in a bank, damage declaration in an insurance organisation, or billing, payroll, and other similar operations that can be found in almost every organisation. In ISD, transformation of rules that apply to such operations (into requirements) is almost straightforward, since specifications of the rules are already present in documents, procedures, policies, regulations, user manuals, etc. This, however, is not always the case. Many rules that are used in business operation have no formal representation. They are based on tacit knowledge of individuals and are dynamic in nature and hard to express. They are highly personal and subjective, based on experiences, ideals, emotions and intuition. They can be formed and destroyed dynamically as a consequence of a process execution and feedback. Elicitation of such rules and their transformation into unambiguous declarations is complicated and typically not performed at all.

This would not be a problem if tacit knowledge would not be so important for clarifying the context of explicitly stated business rules. The fact is that explicit business rules are only manifestation of

typically richer knowledge. In every organisation, operation procedures can be found that are standardised as a result of experience and feedback from their use. Business rules that are derived from such operations can be documented and translated into explicit business rules and further automated within an IS. In this way, they become available to those who lack the knowledge of the operation. However, in order to understand why a certain explicit rule exists, and what kind of motivation is behind it, one must first comprehend the knowledge on which the explicit rule is developed (Morabito et al 2001, Zachman 1987). Otherwise it may happen we do not know why some of the rules exist and whether they still reflect the actual business environment.

While a lot of work has been done to support the acquisition of business rules (tool support, templates, etc.) it seems that there is only little interest in capturing the rationale behind the rules. This is specifically visible in practice, where the rules are typically captured without any source that would explain their motivation. This makes business rules difficult to track.

The challenge in this area is thus to identify possibilities to capture the rationale behind business rules. The knowledge and experiences from the field of knowledge management should be very valuable here. Important input for modelling sources of business rules should also be found in the business modelling community. An idea how business rules might be propagated from business models to IS development is described in Bajec & Krisper (2004).

3.3 Business rule specification and modelling

Once business rules are acquired, they have to be specified in an appropriate manner. This, of course, is not easy, as the rules have to be understandable to business and to IS people. While business people are typically not familiar with formal languages, developers require unambiguous declarations to be able to write program code for the execution of rules. It is thus clear that business rules require specifications in different levels of formality to be understandable to all the people that work with them.

The formal specification of business rules is nowadays not a problem. Besides traditional techniques and methods (e.g. UML diagrams) there have been other solutions proposed, including specifically designed languages, such as External rule language (McBrien et al 1991), Object Constraint Language (Warmer and Anneke 1999), RossMethod (Ross 1997), etc. The problem lies in finding an informal language, such that would be understandable to business people but still clear enough to be unambiguous.

The solution for informal or semi-formal specification of rules that has received a lot of attention recently is the use of rule templates. A rule template can be seen as a sentence pattern that tells how to describe the rules that belong to a particular category. An example of a rule classification scheme supported with rule templates can be found in (Martin and Odell 1998). Rule templates have been supported also in many tools that have been developed in support of business rule approach.

The challenge in this area is to find a standard taxonomy that will define a closed set of categories with supporting rule templates for the rule specification. Numerous taxonomies that have been proposed so far (see e.g. Struck 1999) only hinder the use of rule templates, as there are too many of them. Furthermore, it is not clear whether it is even possible to define the categories for the specification of all kinds of rules. There will always be rules that will not fit into existing categories and will require new templates. And if the rule categories and templates have to be created, who is the one that defines them? If these are not business people than we are faced with the same problem again: business people can not define business rules in their own but they have to rely on developers who define rule templates for them.

Another problem that deserves to be mentioned here is the fact that business rules are inherent in many traditional models. In a typical data model, for example, there are many business rules that define or constrain data structures being modelled. Other models that use graphical notations, like process diagrams, activity diagrams, class diagrams, etc. include many rules too (Bajec and Krisper 2004).

Should these rules that are intrinsic in traditional models be specified also in a rule language? In a typical data model, there could be hundreds of business rules. And if the rules are both, modelled in some graphical notation (implicitly through other perspectives) and specified in some rule language, how can we keep them synchronised? The developers of CASE tools do not seem to give any significant attention to this problem, as modelling tools that are in use today rarely support this feature. This is in fact one of the biggest problems that hinder an explicit manipulation of business rules in ISD. It is also not realistic to expect the developers of CASE tools will finally accept this idea since the foundation for explicit business rule manipulation is known over 15 years but nothing happened yet. At least not in CASE tools or IDE environments that are most popular today. Fortunately, there are some indices the explicit business rule manipulation could be supported by the "model driven approach". We will discuss this in the next section.

3.4 Business rule implementation

One of the most important activities in business rule-based ISD is the business rule implementation. There are a number of different technologies and tools available to support business rule implementation and maintenance (including code generation) (Bajec et al 2000). They range from database-oriented tools that enforce rules using database mechanisms, such as triggers and stored procedures, to rule oriented systems that offer declarative rule specification languages and special mechanisms to take care of the rule execution. Which technology should be used depends on several factors, but particularly on the type of the system being developed. For example, in a typical knowledge-based application, rules will be captured and stored into a rule base and executed by a rule engine. In a typical workflow system, business rules will be integrated in the workflow definition, which will be used by a workflow engine to run the workflow. In a typically database oriented system, business rules will be probably spread across the entire application.

One of the challenges in the business rule implementation is to provide a tool support for the generation of program code (for the execution of business rules) from the business rule specifications. The tools that are available today and offer this kind of functionality are technology-oriented. They offer code generation but only for selected technologies. For instance, if using a tool that uses a rule engine to process rules, the programme code can only be generated for the use in that rule engine. Furthermore, since specifications must be rigorous to interpret them by a computer, the tools often support only limited number of rule categories.

A promising idea for the solution of this problem is presented by the Model Driven Architecture - MDA (Mellor et al 2004). In its most simple form MDA can be seen as a model-based code generation. In MDA, a model represents an abstraction of the target system. It serves as a prototype and a proof-of-concept. MDA defines two types of models: a Platform Independent Model (PIM) and a Platform Specific Model (PSM). While PIM describes the target system without any details about the specifics of the implementation platform, PSM do the rest and describes the target system on its intended platform (e.g. .NET, J2EE, etc.). The process of converting PIM to PSM depends on the modelling languages that are used to express PIM and PSM. MDA is not restricted to any particular modelling language. The only restriction is that PIM and PSM are both expressed in a way that can be unambiguously interpreted by a computer. The process of transformation can be then written for each particular technology separately. For instance, for each technology a specific "plug-in" is written that takes care for the transformation from PIM to PSM. If a new element is defined on the platform-independent or platform-specific level, the plug-in is adapted. For each new technology, a new plug-in is written.

The MDA concept, as described above, can also be applied in business rule-based development where business rules are transformed from specifications to code. There are several conditions however, that have to be satisfied:

- Appropriate languages for PIM and PSM are needed. Since PIM and PSM both require formal languages, an additional level of abstraction (an abstraction of PIM) has to be defined to support

business rule specification in a form that is understandable to business people. This means that formal languages like UML or OCL are not an option.

- Plug-ins have to be written to support transformation of PIM to various technologies that are available for the processing and execution of business rules. In addition, an automatic conversion has to be supported to transform business language (the highest level of business rule abstraction) to PIM. Here we face again with the problems that have been mentioned in the previous section.
- Since MDA is only a concept, an appropriate tool support is required.

3.5 Business rule management

If business rules are not handled in traditional ways, implicitly within other aspects of IS (e.g. data, processes, functions, objects etc.), but are addressed explicitly, than they also require an appropriate level of management. This includes business rule acquisition, specification, modelling, implementation, and also maintenance, such as version control, change control, monitoring rule efficiency, etc. While there is a variety of tools and approaches that provide IS developers with facilities for managing business rules through IS development, there is only limited support for an enterprise-wide rule management (Bajec and Krisper 2004). Business rules, however, do not pertain to IS or to its application software. Business rules are set and owned by the business and have to be therefore managed by the business.

Changes in an organisation's business environment almost never happen spontaneously, without any reason, but are typically driven either from internal decisions of the organisation's management or from external forces, such as government laws and regulations. Such changes very often lead to adaptation of existing business processes and frequently require new or modified systems support. What usually changes in the business processes and in the supporting systems are business rules and their implementations, which are re-examined and modified according to the new objectives, goals and policies. This requires the changes to be coordinated at the enterprise level, as the business rules are spread across the entire organisation and its supporting systems. A particular business rule may be involved in several business processes, and supported by several subsystems. Furthermore, in a particular subsystem, each rule may be implemented in a number of different ways (e.g. as a database trigger, stored procedure, middle-tire component, etc.). In order to be able to keep supporting systems consistent with the business requirements, it has to be documented how business rules evolve from their origin in business environment to their implementation in IS. In this way, it is easier to determine what are or what could be the implications of business changes for the supporting systems.

There are many challenges in the field of enterprise-wide business rule management. Although a lot of work has been done to find appropriate ways of managing business rules on enterprise level (see e.g. Bubenko et al 2001, Loucopoulos 2000, Bajec 2001, Bubenko and Wangler 1993, Tempora 1994, Assche et al 1988, McBrien et al 1991, Locopolus et al 1997, Nellborn and Gustafsson 1992, Yu et al 1995, Dobson 1992), the results are not used in practice very often. It seems that organisations are not interested enough to manage business rules on enterprise level. There are at least two problems that we should call attention to:

- The establishment of an environment in which business rules could be managed for an entire organization requires an extensive work, for which organizations may not be motivated enough. It is therefore necessary to find ways to minimize the overhead that comes with the enterprise-wide rule management. Perhaps with integration with the processes that deal with business analysis, such as strategy planning, business process reengineering, IS renovation, etc.
- There is only limited tool support for the business rule management on enterprise level (see Bajec and Krisper 2004). To be able to manage business rules for an entire organization appropriate tool support is central.

4 SUMMARY AND CONCLUSIONS

As it was remarked in the business rule workshop in 1998 (see Mens et al 1998), '*business rules are nothing new. They are used by every organisation to state their practices and policies'*. Consequently, they are also present in every business application. Their importance for the IS community has become evident with the notion that they represent IS components which are far most influenced by the changes that happen in a turbulent business environment.

As it was shown in the background section, a substantial effort was put into discovering an efficient and robust way to deal with business rules explicitly during their evolution through the entire ISD lifecycle. Nevertheless, we are still far from the position to say that business rules are dealt with that way. The ISD approaches and supporting tools that are most popular today do not seem to put that much emphasis on business rules and their explicit manipulation. It is also evident that there is no consensus among the members of the IS community on whether business rules require an explicit treatment throughout the entire ISD lifecycle. Many agree that business rules are just another perspective that should call attention to in ISD and that it should be enough to know what the rules are and where are they implemented.

Our objective in this paper was to identify the issues that we think are problematic in the context of business rule explicit manipulation and thus present challenges for the future research. Specifically, the focus was put on the five areas: the business rule scope, acquisition, specification, implementation, and management. For each of the areas we have pointed out the issues that present obstacles for using business rules as an approach to ISD. Based on our work we conclude that the business rule research has to continue and that there are still many unknowns that prevent the use of the business rule approach in everyday practice.

We would like to emphasise here that some of the arguments stated in the paper present the authors' own view on the subject and thus may not receive total agreement among other researchers and practitioners. Should they draw attention and lead into new discussions the paper has met its intention.

5 FURTHER RESEARCH

This paper is the product of an ongoing research on business rules. It is based mainly on our own experiences in business rule-based ISD, literature review and numerous discussions that we had with other researchers and practitioners from the field. The next stage of this work will include the definition of the conceptual framework to analytically compare the various approaches in business rule-based ISD. An in-depth discussion of the needs, rationale for, philosophy of business rules and critical evaluation of different approaches shall indicate what are the gaps and prospects of the business rule-based ISD.

6 REFERENCES

Anthes, G.H. (2003). Eyes everywhere: Business activity monitoring offers a constant watch on business processes, Computerworld, November 2003.

Appelton, D.S. (1984). Business Rules – The Missing Link. Datamation, 30(16), 145-150.

Van Assche, F., Layzell, P. J. and Anderson, M. (1988). RUBRIC – A Rule-Based Approach to Information Systems Requirements. Proceedings of the 1st European Conference on Information Technology for Organisational Systems. Athens.

Bajec, M. and Krisper, M. (2004). A methodology and tool-support for business rule management in organsiations. Information Systems, article in press.

Bajec, M. (2001). Definition of the Conceptual Framework for Business Rule Management in Organisations, PhD Thesis, Laboratory of Information Systems, UL - Faculty of Computer & Information Science, Ljubljana, Slovenia.

Bajec, M., Krisper, M. and Rupnik, R. (2000). Using Business Rules Technologies To Bridge The Gap Between Business And Business Applications. Proceedings of the IFIP 16th World Computer Congress 2000, Information Technology for Business Management (G Rechnu, Ed), Beijing, China, 77-85.

Barnes, M. and Kelly, D. (1997). Play by the Rules. Byte (Special Report), 22(6), 98-102.

Bubenko, J.A. and Wangler, B. (1993). Objectives driven capture of business rules and of information systems requirements. Proceedings of International Conference on Systems, Man and Cybernetics. 'Systems Engineering in the Service of Humans', Vol. 1, 670-677.

Bubenko, J. A., Persson, A. and Stirna, J. (2001). D3 Appendix B: EKD User Guide, Royal Institute of Technology (KTH) and Stockholm University, Stockholm, Sweden.

Ceri, S. and Fraternale, P. (1997). Designing Database Applications with Objects and Rules: The IDEA Methodology, Addison-Wesley.

Date, C. J. (2000). What Not How: The Business Rules Approach To Application Development. Addison Wesley Longman, Inc.

Dayal, U., Buchmann, A.P. and McCharty, D.R. (1988). Rules are Objects too: A Knowledge Model for an Active, Object-Oriented Database Management System. Advances in Object-Oriented Database Systems (Ed. K.R. Dittrich), Springer, Berlin, 129-143.

Danny, C. C. P. (1999). Events in Use Cases as a Basis for Identifying and Specifying Classes and Business Rules. In Proceedings TOOLS 29 (TOOLS Europe 99) Conference, 7-10 June 1999, Nancy, France, 204-213.

Dobson, J. (1992). A Methodology for Managing Organisational Requirements. University of Newcastle upon Tyne, Newcastle, UK.

Diaz, O., Iturrioz, J. and Piattini, G. M. (1998). Promoting business policies in object-oriented methods. The Journal of Systems and Software 41(1998), 105-115.

Downs, E., Clare, P. and Coe, I. (1992). Structured System Analysis and Design Method – Application and Context, 2nd edition, Prentice-Hall, Englewood Cliffs.

Elder, J., Kappel, G., Tjoa, A.M. and Wagner, R.R. (1987). BIER - The Behaviour Integrated Entity-Relationship Approach. Proceedings of the 5th International Conference on Entity-Relationship Approach (Ed: S. Spaccapietra), North Holland, Amsterdam, 147-166.

Elmasri, R. and Navathe, S. B. (1989). Fundamentals of Database Systems, The Benjamin/Cummings Publishing Company, Inc.

Gallaire, H., Minker, J. and Nicolas, J.M. (1984). Logic and Databases: A Deductive Approach, Computing Surveys, 16(2).

Greenspan, S. J., Mylopoulos, J. and Borgida, A. (1982). Capturing More World Knowledge in the Requirements Specification, Proceedings of the International Conference on Software Engineering, Tokyo.

Gottesdiener, E. (1999). Capturing Business Rules. Software Development Magazine: Management Forum, (7)12, (1999).

Hay, D. and Healy, K.A. (1997). GUIDE Business Rules Project, Final Report – revision 1.2. GUIDE International Corporation, Chicago.

Herbst, H. (1997). Business Rule-Oriented Conceptual Modelling, Heiderberg, Physica.

Herbst, H. (1996). Business Rules in Systems Analysis: A Meta-Model and Repository System. Information Systems, 21 (2), 147-166.

Herbst, H., Knolmayer, G., Myrach, T. and Schlesinger, M.. The Specification of Business Rules: A Comprison of Selected Methodologies. Methods and Associated Tools for the Information Systems Life Cycle (A. Verrijin and T. W. Olle, Ed), Amsterdam at al.: Elsevier 1994, 29-46.

Layzell, P.J. and Loucopoulos, P. (1988). A Rule-Based Approach to the Construction and Evolution of Business Information Systems. Proceedings of the 4th IEEE International Conference on Software Maintenance, Phoenix, Arizona, USA, 258-264.

Loucopoulos, P., Theodoulidis, B. and Pantazis, D. (1991). Business Rules Modelling: Conceptual Modelling and Object Oriented Specifications. Proceedings of the IFIP TC8/WG8.1 Working Conference, Netherlands, Nov 28-31, 323-342.

Loucopoulos, P., Kavakli, V., Prekas, N., Rolland, C., Grosz, G. and Nurcan, S. (1997). Using the EKD Approach: The Modelling Component, UMIST, Manchester, UK.
Loucopoulos, P. (2000). From Information Modelling to Enterprise Modelling, in Information Systems Engineering: State of the Art and Research Themes, (Ed: S. Brinkkemper, E. Lindencrona, A. Solvberg), Springer, 67-78.
Martin, J. and Odell, J. (1998). Object-Oriented Methods, A Foundation, Prentice Hall.
McBrien, P.J., Niézette, M., Pantazis, D., Seltveit, A.H., Sundin, U., Theodoulidis, B., Tziallas, G. and Wohed, R. (1991). A Rule Language to Capture and Model Business Policy Specifications. Proceedings of CAiSE 1991: Spinger-Verlag LNCS 498, 307-318.
Mellor, S. J., Scott, K., Uhl, A. and Weise, D. (2004). MDA Distilled: Principles of Model-Driven Architecture. Addison-Wesley.
Mens, K., Wuyts, R., Bontridder, D. and Grijseels, A. (1998). Tools and Environments for Business Rules, Workshop Report, ECOOP'98 Workshop 7. 12th European Conference on Object-Oriented Programming, Brussels, Belgium, July 20 - 24, 1998.
Minker, J. (ed). (1988). Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann Publishers, Inc., (1988).
Morabito, J., Sack, I. and Bhate, A. (2001). Organisation Modelling, Innovative Architectures for the 21st Century. NJ: Prentice Hall.
Morgenstern, M. (1983). Active Databases as a Paradigm for Enhanced Computing Environments. Proceeding of the 9th International conference on Very Large Databases, Florence, Italy, 34-42.
Moriarty, T. (2000). Business Rule Management Facility: System Architect 2001, Intelligent Enterprise, 3 (12).
Moriarty, T. (1993). The Next Paradigm. Database Programming and Design. 6 (2), 66-69.
Nellborn, C., Gustafsson, M.R., Bubenko, J.A. jr. (1992). Enterprise Modelling - an Approach to Capture Requirements, report no P6612.SISU.RP.001.1, SISU (Swedish Institute for Systems Development), (1992).
Nijssen, G.M. and Halpin, T.A. (1989). Conceptual Schema and Relational Database Design – A Fact Oriented Approach, Prentice-Hall, Englewood Cliffs.
Nilsson, B. E. (1999). On Why to Model What and How: Concepts and Architecture for Change. Perspectives on Business Modelling – Understanding and Changing Organisations (Ed: A. G. Nilsson et. Al.), Springer Verlag.
Nonaka, I. (1991). The Knowledge-Creating Company. Harvard Business Review, Nov.-Dec. 1991.
Petrounias, I. and Loucopoulos, P. (1994). A Rule Based Approach for the Design and Implementation of Information Systems. Proceedings EDBT '94 (Ed: M. Jarke), Springer-Verlag, Cambridge, U.K.
Rosca, D., Greenspan, S. Wild, C., Reubeinstein, H., Maly, K. and Feblowitz, M. (1995). Application of a decision support mechanism to the business rules life cycle. Proceedings of the 10th Knowledge-Based Software Engineering Conference, 114-121.
Ross, R. (2003). Principles of the Business Rule Approach. Addison Wesley Professional Series, Addison Wesley.
Ross, R. (1998). Business Rule Concepts. The New Mechanics of Business Information Systems. Business Rule Solutions, Inc., Huston, Texas.
Ross, R. (1997). The Business Rule Book: Classifying, Defining and Modelling Rules, Second Edition, (Ross Method, version 4.0). Business Rule Solutions, Inc., Huston, Texas.
Schlesinger, M., Hanson, E. and Hong, C. (1987). The Design of the POSTGRES Rules System. Proceedings of the IEEE International Conference on Data Engineering, 365-374.
Selveith, H. (1991). Modelling Business Rules. Report on TEMPORA Project, January 1991.
Sinur, J. (2003). The Business Rule Engine 2003 Magic Quadrant, Gartner Group Research Note, April 2003.
Struck, D.L. (1999). Business Rule Continuous Requirements Environment. PhD Thesis, Colorado Technical University, Colorado Springs, Colorado.
Tanaka, A.K. (1992). On Conceptual Design of Active Databases. PhD Thesis, Georgia Institute of Technology, Georga.

Tanaka, A.K., Navathe, S.B., Chakravarthy, S. and Karlapalem, K. (1991). ER-R: An Enhanced ER Model with Situation-Action Rules to Capture Application Semantics. Proceedings of the 10th International Conference on the Entity Relationship Approach (Ed: T.J. Teorey), E/R Institute, San Mateo, 59-75.

Tempora (1994). TEMPORA final review. Technical report, TEMPORA Consortium.

Von Halle, B. (2001). Business Rules Applied: Building Better Systems Using the Business Rules Approach. Wiley.

Zachman, Y. A. (1987). A Framework for Information System Architecture. IBM System Journal, 26(3), str.276-292. (1987).

Warmer, J. B. and Anneke, G. K. (1999). The Object Constraint Language: Precise Modelling With UML (Addison-Wesley Object Technology Series), Addison-Wesley. (1999).

Widom, J. and Ceri, S. (1996). Active Database Systems – Triggers and Rules For Advanced Database Processing. Morgan Kaufmann, San Francisco (1996).

Yu, E., Dubois, P., Dubois, E. and Mylopoulus, J. (1995). From Organization Models to System Requirements - A 'Cooperating Agents' Approach. Proceedings of the 3rd International Conference on Cooperative Information Systems - CoopIS-95, Vienna (Austria), May 9-12, 194-204.

Youdeowei, A. (1997). The B-Rule Methodology: A Business Rule Approach to Information Systems Development, PhD Thesis, Department of Computation UMIST, Manchester, United Kingdom.