

A Meta-Model for Business Rules in Systems Analysis

H. Herbst

herbst@ie.jwi.unibe.ch

Institute of Information Systems, Research Unit 'Information Engineering', University of Berne
Engehaldenstrasse 8, CH-3012 Berne, Switzerland*

Abstract. Commonly used methodologies for systems analysis are data- or function-oriented and are sufficient for information systems which will be implemented on passive database management systems (DBMS). In the last years, several research prototypes of active DBMS and active mechanisms in commercially available DBMS have been developed. To fully use the potential of these rule-based mechanisms, a rule-based systems analysis methodology seems necessary. This paper defines and structures business rules as a main component of such a methodology and presents a meta-model for business rules; furthermore, an outlook on the implementation of the meta-model in a repository system is given.

1 Introduction

Many current database research projects are focusing on the specification and implementation of databases which provide active components (cf. [StHH87]; [DaBM88]; [WiCL91]; [Gatz94]; [ChMi93]). First results of this research are already available in commercial DBMS such as CA-Ingres, Oracle or Sybase (cf. [KnHS94]). In connection with this progress on the implementation level, the importance of treating business rules on the conceptual level has been emphasized (cf. [App184]; [VLLS88]; [BBSG90]; [Mori93]; [KnHe93]). Though the term *business rule* is often referred to, it is defined and used rather differently and often restricted to semantic integrity constraints (cf. e.g. [App188]; [SaVo91]). However, business rules do not only cover data integrity but may also impose restrictions on organizational dynamics; therefore, we define business rules with reference to [BBSG90] as statements about how the business is done, i.e., about guidelines and restrictions with respect to states and processes in an organization [HeKn94].

In [HKMS94] a comparison of selected function-oriented, data-oriented, and object-oriented methods revealed the lack of an approach for a convenient graphical specification of business rules at the conceptual level. To complete the ongoing research dealing with the graphical representation of business rules (cf. e.g. [Tana92], [MaOd92]) we focus on the administration of business rules in a repository and try to derive graphical views from it.

* The work presented in this paper has been partially supported by the Swiss National Science Foundation, Priority Program Informatics, Project 5003-034330.

2 Business Rules

2.1 Systems Analysis Focusing on Business Rules

A main purpose of systems analysis is to collect all relevant information about the universe of discourse [Pohl93]; these facts are primarily about processes and about the structure and manipulation of data objects. Business rules as defined above encompass both aspects; therefore, we want to treat business rules as a central element of systems analysis and adapt the process of systems analysis accordingly (cf. Figure 1). In this process facts are collected in the real world (e.g. by interviews or document analysis) and specified as business rules and data structures. After verifying the specifications, they can be used in systems design and implementation. The first part of the process may be supported by a repository system which encompasses functionalities to store, retrieve, validate represent, and manipulate the relevant meta-data including business rules.

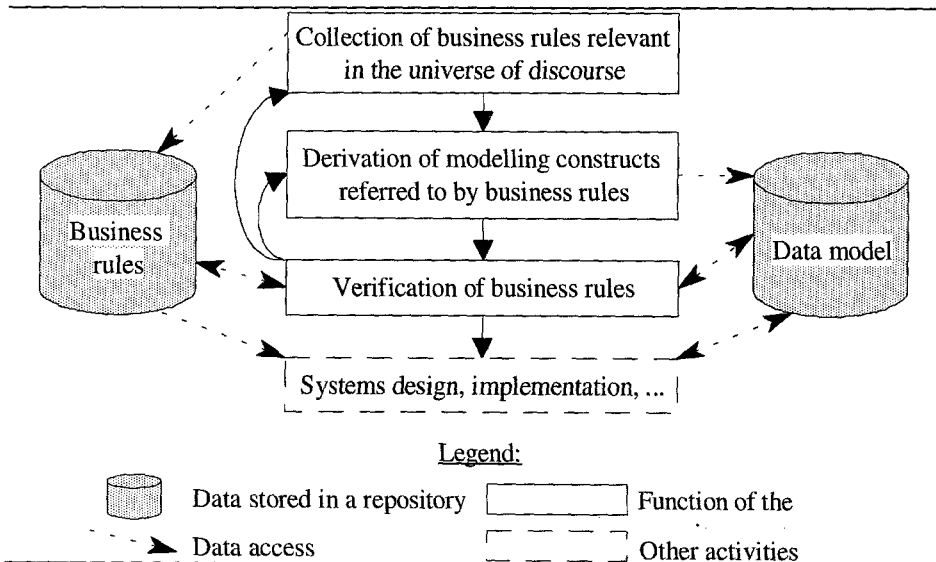


Figure 1: Systems analysis focusing on business rules

2.2 The Structure of Business Rules

Business rules can be represented according to the structure of rules in active databases (cf. [DaBM88]) using the three basic components event, condition and action (ECA) [TsKL90]. Several case studies to extract business rules from practically applied information systems (cf. [KnHS94]) revealed the need for an extension of this structure to ECAA:

- *Event*: When has a business rule to be processed?
- *Condition*: What has to be checked?
- *Then-Action*: What has to be done if the condition is true?
- *Else-Action*: What has to be done if the condition is false?

These basic component types of business rules can be further classified into several subtypes [HeKn94] which is especially relevant for the definition of a syntax for describing business rules in a repository.

The ECAA structure allows to specify single business rules (e.g. encompassing static integrity constraints) and the definition of entire processes consisting of business rules. The importance of events with respect to processes is emphasized in [Stre94] and [Sche94]. For specifying processes, the action component of a rule has to be linked to raised events which trigger other business rules; this link is specified within the action component.

To illustrate the scope and different types of business rules, we introduce some examples which may be relevant in an order processing system. In this system we want to specify the process of order entry and some constraints for the treatment of orders. In the first stage of the order processing we want to assure that a person who calls to put an order is a customer:

```
[BR1] ON    phone call of a person
        IF    (person is a registrated customer)
        THEN  specify order, ⇔ EVENT 'order specified'
        ELSE  reject order, ⇔ EVENT 'order rejected'
```

After being identified as a customer, he or she specifies an order; the order is only accepted if the total amount of the order does not exceed the actual credit limit of the customer:

```
[BR2] ON    order specified
        IF    (credit-limit of customer > order-total)
        THEN  registrate order, ⇔ EVENT 'order registrated'
              SET order-state := 'accepted'
              SET credit-limit := credit-limit - order-total
        ELSE  reject order, ⇔ EVENT 'order rejected'
```

The acceptance of an order results in a confirmation letter which is sent to the customer. Apart from the confirmation, the task of assembling the order is triggered. Because every order has to be delivered within 30 days, we remind the responsible clerk 20 days after the acceptance to deliver the order.

```
[BR3] ON    order registrated
        THEN  write confirmation letter
              SET order-state := 'confirmed'

[BR4] ON    order registrated
        THEN  assemble order, ⇔ EVENT 'order assembled'

[BR5] ON    20 days AFTER [order registrated]
        IF    order not assembled
        THEN  remind responsible clerk
```

Finally, the order is delivered to the customer who is expected to pay the order total within 30 days.

```
[BR6] ON    order assembled
        THEN  deliver order, ⇔ EVENT 'order delivered'

[BR7] ON    30 days AFTER [order delivered]
        IF    (order not paid)
        THEN  remind customer, ⇔ EVENT 'customer reminded'
```

```
[BR8] ON    order paid
      THEN SET order state = 'closed', ⇒ EVENT 'order closed'
           SET credit-limit := credit-limit + order-total
           SET closure date := today()
```

After payment the order is closed and kept in the system for ten years.

```
[BR9] ON    10 years AFTER [order closed]
      THEN attempt to delete order, ⇒ EVENT 'order deletion attempted'
```

To prevent an erase of an order prior to this delay event, the following rule is specified:

```
[BR10] ON   order deletion attempted
        IF   order closure date > (today() - 10 years)
        THEN reject deletion of order
             issue message „Orders must be kept for 10 years“
        ELSE delete order, ⇒ EVENT 'order deletion'
```

To ensure that no customer is deleted while having an order, the following business rule may be specified:

```
[BR11] ON   customer deletion attempted
        IF   ∃ order related to customer
        THEN reject deletion of customer
             issue message „Deletion not possible because of existing orders“
        ELSE delete customer, ⇒ EVENT 'customer deleted'
```

These examples show the close relation of business rules to the concepts of 'workflow management' (cf. e.g. [Heil94]) and 'business process re-engineering' [HaCh93]; therefore, the meta-model presented in this paper includes a submodel 'process' (cf. section 3.2.4).

2.3 Abstraction and Specialization of Business Rules

An IS may encompass plenty of business rules; therefore, mechanisms for reducing the complexity have to be provided. One possibility to achieve this is a leveled approach for the specification of business rules, i.e., their generalization and specialization of business rules. In [DeMa78] a balancing rule for leveled dataflow diagrams (DFD) is specified: The parent and the child diagram have to be *balanced*, i.e., „data flows into and out of the parent bubble are equivalent to data flows into and out of the child diagram“. An adaptation of this rule on the specialization of business rules is that on both levels the triggering event and the raised events have to be identical.

In our concept, different levels for specifying business rules are allocated to (sub)processes encompassing a specific subset of all business rules (cf. Figure 2). This allocation allows a reuse of business rules on different levels (e.g. the business rule 'BR-4' in Figure 2). The reused business rules always raise the same events but they may trigger different business rules, depending on the subset of business rules which define the current subprocess.

Because the concept of business rules incorporates these different abstraction levels, allows the description of concurrent activities and explicitly incorporates the time dimension (event types 'time event', 'interval events', 'periodical events' and 'delay events'), the model can be classified as dynamic (cf. [MeBS94]).

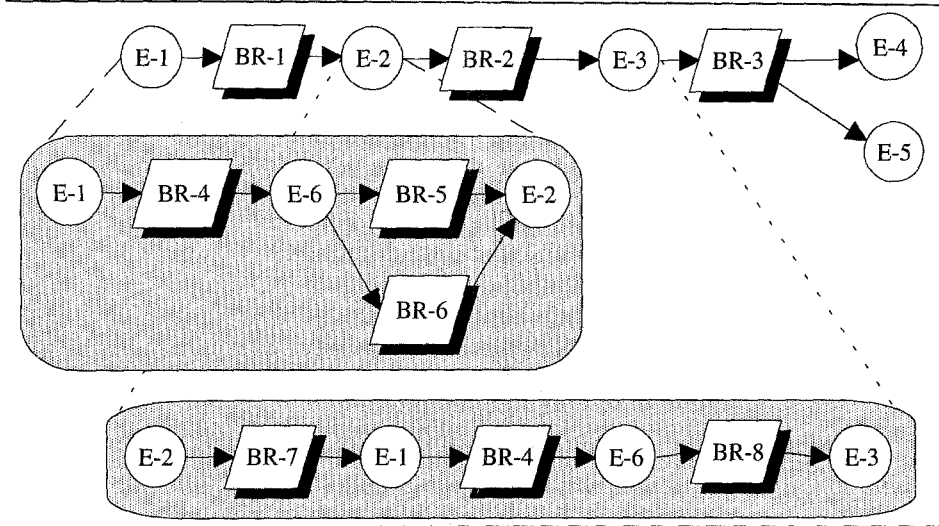


Figure 2: Specialization of business rules

3 A Meta-Model for Business Rules

The meta-model presented in this chapter is assigned to the second layer of the IRDS standard [ANSI89] and describes therefore information about the meta-structure of facts to be stored in the data dictionary. The meta-model is discussed in two parts: the first is about the core of the meta-model including the two submodels '*Business Rule*' and '*Modeling constructs*' and the second deals with the embedding of business rules into further submodels.

3.1 The Core of the Meta-Model

3.1.1 Submodel 'Business Rule'

The submodel 'Business rule' consists of the four meta-entity-types *Business rule*, *Event*, *Condition* and *Action* (cf. Figure 3). Every business rule has exactly one event, at most one condition and one or two actions (then/else). Furthermore, business rules can be specialized and generalized which leads to a recursive m:n relationship on *Business rule*. Events and conditions can be complex and have therefore also recursive m:n relationships. As shown in the examples actions of business rules may raise events which is depicted in the meta model as a relationship between the meta-entity-types *Action* and *Event*.

3.1.2 Submodel 'Modeling Construct'

The submodel 'Modeling construct' encompasses the meta-entity-types for the specification of a conceptual data model. The most popular data model for conceptual modeling is the Entity Relationship Model; therefore, we incorporate into the submodel 'Modeling construct' the meta-entity-types *entity type*, *relationship type* and *attribute* (cf. Figure 3). The semantic of the relationship between components of business rules and modeling constructs is e.g. the retrieval, modification, derivation or deletion of the data of a modeling construct. The following relationships between components of business rules and modeling constructs are possible:

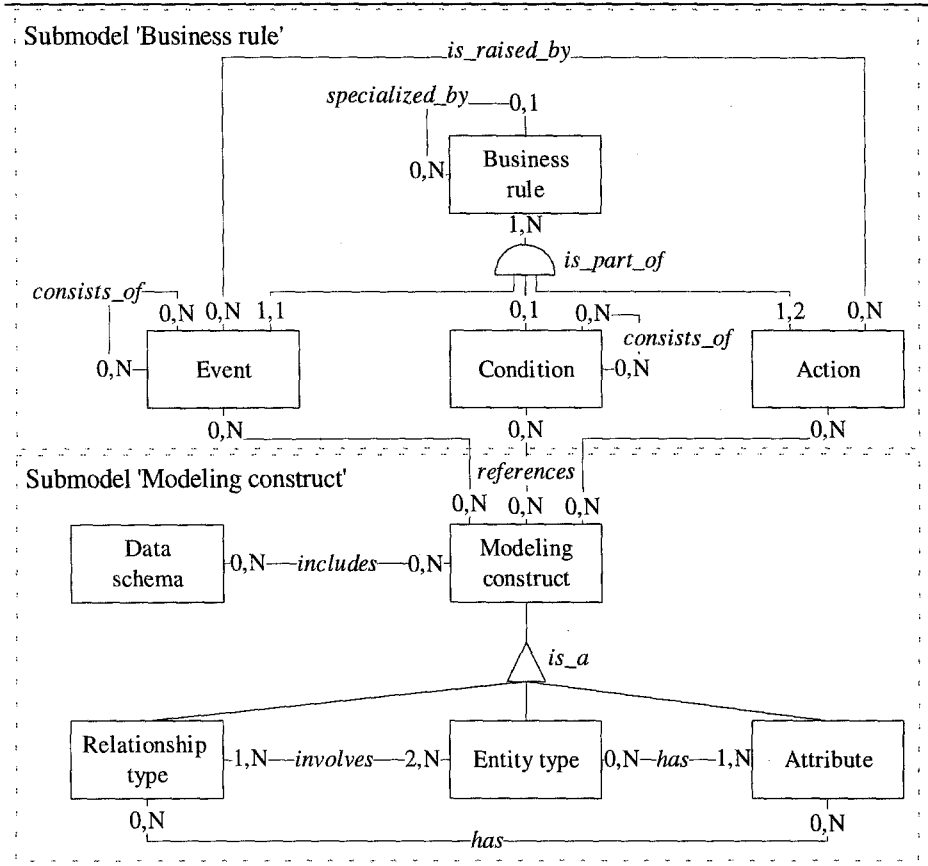


Figure 3: Submodels 'Business rule' and 'Modeling construct'

- Relationship between *Event* and *Modeling construct*: Data related events (which may also be part of a complex event) make reference to modeling constructs within the universe of discourse.
- Relationship between *Condition* and *Modeling construct*: The specification of conditions always refers to modeling constructs whose content has to be retrieved for the evaluation of the condition.
- Relationship between *Action* and *Modeling construct*: An action may explicitly refer to modeling constructs.

These relationships between modeling constructs and business rules may be depicted e.g. in an Entity-Relationship-Event-Rule (ER²) diagram [Tana92].

As discussed in the description of a business rule based systems analysis (cf. Figure 1), modeling constructs can be derived from a collection of business rules and afterwards structured in a conceptual data model (cf. Figure 4).

The submodels '*business rule*' and '*modeling construct*' allow a data view on the meta model focusing on the relationship between modeling constructs and business rules; it can be regarded from the following points of view:

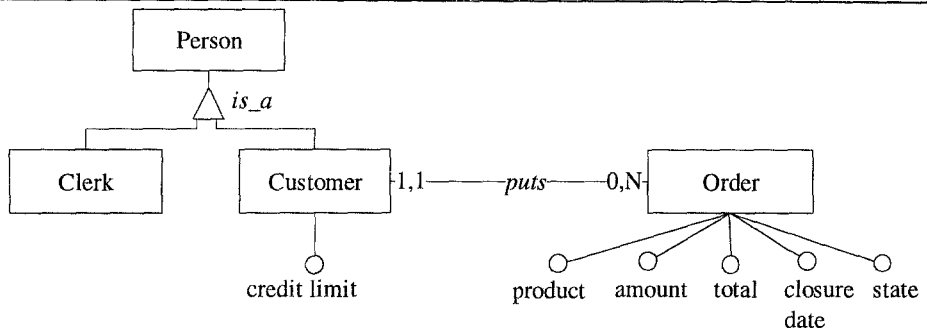


Figure 4: Data model resulting from the example business rules

1. Business rule \Leftrightarrow modeling construct: Which impact has a specific business rule on modeling constructs?

Example: The business rule [BR2]

includes registration of order (\cong insertion)
order state (modification)

...

2. Modeling construct \Leftrightarrow business rule: Which business rules use a specific modeling construct?

Example: The entity type 'Order' is referenced by

[BR2] *Event, action* (modification)

[BR3] *Event, action* (modification)

...

3.2 Embedding Business Rules into their Environment

3.2.1 Overview

Business rules as defined and exemplified above describe how the business is done; however, to obtain an integrated view on the universe of discourse they have to be embedded into their own environment. Therefore, the meta-model consists of several submodels linked to the central submodel 'business rule'. Figure 5 shows the relationships between the submodel 'business rule' and the other submodels:

- 'Modeling construct': As discussed above, business rule components may refer to modeling constructs.
- 'Origin': A business rule has at least one origin.
- 'Organizational unit': Each component of a business rule is processed within an organizational unit.
- 'Process': Business rules may be an element of a process, which consists of at least one business rule. Processes are furthermore linked to organizational units.
- 'Software component': Components of business rules may be implemented in a software component as e.g. a module or a stored procedure.

These relationships allow a classification of business rules [HeKn94] which is briefly discussed within the sections describing the submodels.

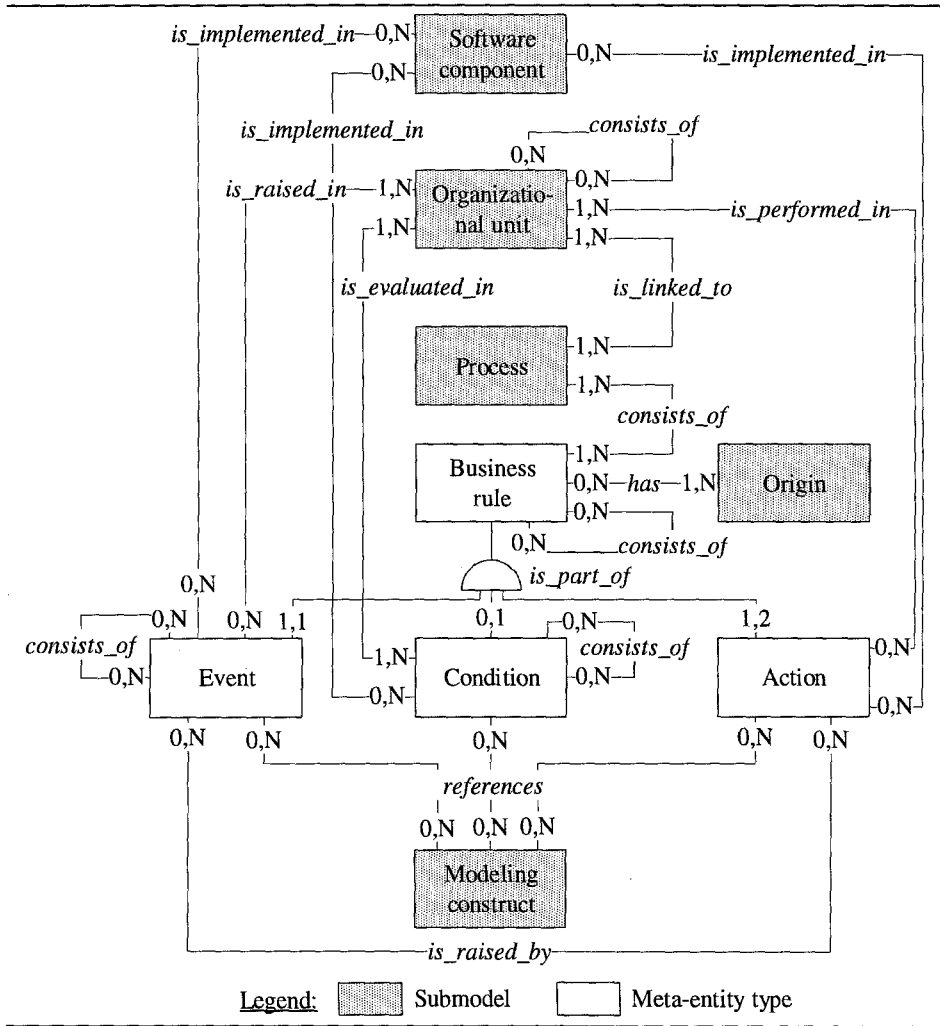


Figure 5: Embedding business rules

3.2.2 Submodel 'Origin'

Business rules may originate outside or inside an organization. Externally originating rules can be further divided into *natural facts* which are eternally fixed and (e.g. legal) *norms* which are specified by the society and may change. Internal origins can be either *primary* or *secondary*; an origin is primary if its content is originally described in a source document, whereas a secondary origin has previously been derived from another source. The knowledge about the origin of a business rule allows among others an analysis from the following two viewpoints:

1. Origin ⇔ business rule: Which business rules originate from a specific origin (e.g. in case of modifications of real world rules from a specific origin)?
2. Business rule ⇔ origin: Where does a specific business rule originate from (e.g. for checking the consistency between the implementation and the real world)?

3.2.3 Submodel ‘Organizational Unit’

The assignment of business rule components to the organizational units which are responsible for processing the components leads to *intra* and *inter* unit rules. This classification may help to support the administration of business rules in an organization. The analysis of the relationship between business rules and organizational units can again be done from two different points of view:

1. Organizational unit \Leftrightarrow (components of a) business rule:

- Which business rules are triggered in a specific organizational unit?
Example: Tasks of the accounts department trigger [BR7] and [BR8]
- Which conditions of business rules are checked in a specific organizational unit?
Example: The accounts department checks the condition of [BR7]
- Which actions of business rules are performed in a specific organizational unit?
Example: The accounts department performs the actions of [BR7] and [BR8]

2. Business rule \Leftrightarrow organizational unit:

- Which organizational units are responsible for the processing of the components of a specific business rule?
Example [BR4]: The registration of an order occurs e.g. in the sales department, whereas the order is assembled by the storage department.

3.2.4 Submodel ‘Process’

Actions of business rules can be related to events resulting in ECA-chains describing the dynamic of processes (cf. [Sche94]) like the example depicted in Figure 6. Such a process is defined by starting and ending business rules; however, because of additional properties, processes can not be entirely described by business rules. These facts about processes include e.g. process name, process owner, involved employees/organizational units and relationships to other processes.

The focus on the process dynamics, i.e., the functional view, can be analyzed from two different viewpoints:

1. Process \Leftrightarrow business rule: Which business rules define a specific process?

Example: Order processing (without accounting) is specified by rules [BR1] to [BR7]

2. Business rule \Leftrightarrow process: Which processes are referred to by components of a business rule?

Example: Within the order specification a reservation for the ordered products is done. This reservation could reduce the available stock below a threshold and cause the reordering of the concerned product. [BR1] would therefore be a part of the order processing and would simultaneously trigger the process of reordering the product from our supplier.

The integrated consideration of the data and functional view leads to information about the modeling constructs to which a process refers.

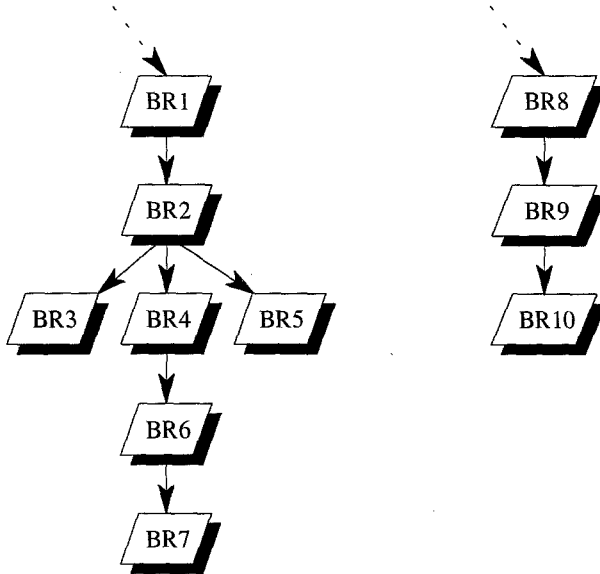


Figure 6: Order processing described by business rules

The representation of the process in Figure 6 shows that most activities are linked through actions raising events and triggering the next business rules. Only between the rules [BR7] and [BR8] no link exists because [BR8] is triggered by the user event 'order paid' which cannot be raised by an action of another business rule of the enterprise. Therefore, the completion of the process depends on raising this event. The diagram shows the importance of this view on business rules because among others it indicates such critical points where the continuation of a process depends on an event from the environment.

3.2.5 Submodel 'Software Component'

The process depicted in Figure 6 delivers no information about the distribution of business rules between the computer system and its environment, i.e., the real world. To obtain this information, the ECA components of the business rules have to be assigned to software components (e.g. functions or database triggers) and/or the system environment (cf. Figure 7), resulting in $2^3=8$ types of rules. This assignment can be determined from the following viewpoints:

- Description of the *current* system (systems survey)
- Description of *possible* assignments (alternatives analysis)
- Description of the *planned* assignments (analysis of target systems).

As stated in [Poo92], high-level business policies are normally transformed into low-level computational representations and are „buried deep within the system program code“. To support the maintenance of an IS, the examination of the *current assignments* helps to find the (possibly redundant) implementations of a specific business rule: the modification of e.g. an organizational guideline leads to the determination of business rules originating from this document; afterwards, the actual implementation can be derived by regarding the relationship to software components.

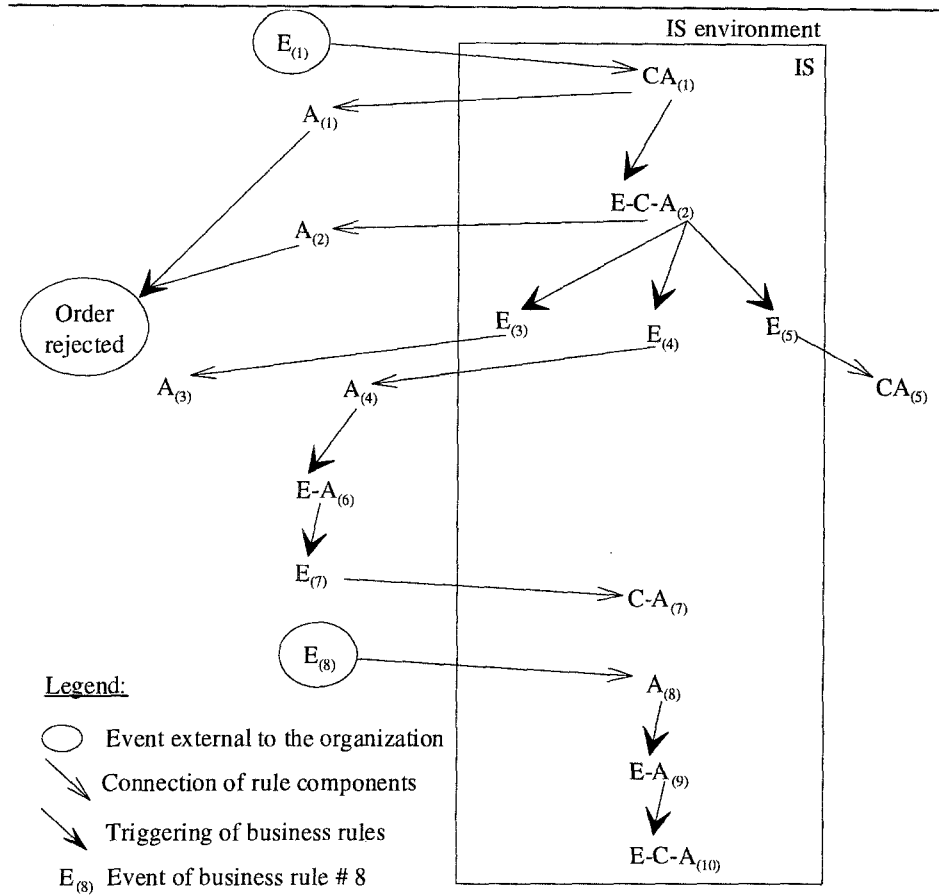


Figure 7: Order processing with assignment of the ECA components to software components and/or the system environment.

The analysis of *alternatives* is only necessary for defining a planned assignment if the current system is not satisfying. The possible assignment especially depends on the question if it is feasible to implement a rule component within an IS. For determining the *planned assignment* of the components, several aspects have to be taken into account [HeKn94]:

- **Flexibility:** Rule components implemented in a computer system are processed very strictly; this may prevent adequate actions in exceptional situations.
- **Consistency of the rule enforcing:** The disadvantage mentioned above may also be regarded as an advantage, because those components implemented in computer systems are always processed in the same consistent way.
- **Data:** How difficult is it to store all data needed for computer based processing of a rule component?
- **Complexity:** Is it possible to implement the logic of a component with acceptable expenses (e.g. non-feasible algorithms)?
- **Number of interactions:** The amount of interfaces necessary for processing a business should be minimized. One should strive for a homogenous rule proces-

sing and assign the whole business rule either to the IS or to its environment.

- *Desirability*: Beneath technical aspects one has to determine whether the control on processing a specific rule component should be left to a computer system (e.g. with respect to ethical reasons).

4 The Implementation of the Meta-Model within a Repository System for Business Rules

The repository system based on the meta-model described is currently being implemented using the repository system Rochade, whose functionalities allow the specification of the meta-model and the implementation of a graphical user-interface for its manipulation.

Repository systems have to provide some basic application independent functionality (cf. [Myra94]); some additional functionality resulting from the specific use for business rule based systems analysis can be derived from Figure 1. For the basic administration of the main submodels 'Business rule' and 'Modeling construct', the repository system has to support the tasks

- business rule registration,
- verification of the business rules (e.g. checking for loops),
- derivation of modeling constructs,
- specification of the data structure and
- verification of the consistency between business rules and modeling constructs.

To cope with the large number of verbally described business rules, the repository system has to provide functionalities for detailed analysis as the ones discussed above. These analysis may result in written output or can be partially visualized using graphical notations as e.g.

- the overview on the relationship between business rules and modeling objects with ER² diagrams [Tana92] or the dynamic of a specific entity type with Entity Life History diagrams [DoCC92]
- the dynamics of business rules with original or adapted Petri-Nets (cf. e.g. [Tana92];[TsGH94]), event-driven process chains [HoKS93], State-Transition Diagrams [Lipe89] or Event Schemes [MaOd92].

Further functions of the repository system will probably include the manipulation of the other submodels.

5 Conclusions and Outlook

Business rules are an important element of IS but rather neglected in commonly used methodologies of systems analysis. According to research in the area of active databases and to proposals for graphical representations of rules implemented in those databases, a meta-model for the treatment of business rules on the conceptual level is proposed. In this meta-model business rules are embedded into their environment which leads to various possibilities for analysis and administration of business rules. This embedded business rules as a technique for systems analysis fulfill requirements of systems users and analysts as e.g. the ones described in [McGi92]. The meta-model is currently being implemented using the commercial repository system Rochade [HeMy95]. The functionality of the business rules repository system allows an easy administration of a large amount of business rules;

furthermore, several graphical representation of views on the meta-data is implemented within Rochade. The business rule repository will subsequently be applied on a large case study encompassing approximately 750 business rules from an insurance application.

References

- [ANSI89] ANSI, American National Standard X3.138-1988: Information Resource Dictionary System (IRDS), New York: American National Standard Institute 1989.
- [Appl84] Appleton, D.S., Business Rules: The Missing Link, in: *Datamation* 30 (1984) 16, pp. 145 - 150.
- [Appl88] Appleton, D.S., Second Generation Applications, in: *Database Programming & Design* 1 (1988) 2, pp. 48 - 54.
- [BBSG90] Bell, J., Brooks, D., Goldbloom, E., Sarro, R., Wood, J.: Re-Engineering Case Study Analysis of Business Rules and Recommendations for Treatment of Rules in a Relational Database Environment, Bellevue Golden: US West Information Technologies Group 1990.
- [ChMi93] Chakravarthy, S., Mishra, D., Snoop: An Expressive Event Specification Language for Active Databases, Technical Report UF-CIS-TR-93-007, University of Florida, 1993.
- [DaBM88] Dayal, U., Buchmann, A.P., McCarthy, D.R., Rules Are Objects Too: A Knowledge Model for an Active, Object-Oriented Database Management System, in: K.R. Dittrich (Eds.), *Advances in Object-Oriented Database Systems*, Berlin et al.: Springer 1988, pp. 129 - 143.
- [DeMa78] De Marco, T., *Structured Analysis and System Specification*, New York: Yourdon 1978.
- [DoCC92] Downs, E., Clare, P., Coe, I., *Structured Systems Analysis and Design Method - Application and Context*, 2nd ed., Englewood Cliffs: Prentice-Hall 1992.
- [Gatz94] Gatzju, S., *Events in an Active Object-Oriented Database System*, Hamburg: Dr. Kovac 1994.
- [HaCh93] Hammer, M., Champy, J., *Reengineering the corporation*, New York: Harper Business 1993.
- [Heil94] Heilmann, H., *Workflow Management: Integration von Organisation und Informationsverarbeitung*, in: *HMD* (1994) 176, pp. 8 - 21.
- [HeKn94] Herbst, H., Knolmayer, G., *Ansätze zur Klassifikation von Geschäftsregeln*, Working Paper 46, Institute of Information Systems, University of Berne 1994, to appear in: *Wirtschaftsinformatik* 37 (1995).
- [HKMS94] Herbst, H., Knolmayer, G., Myrach, T., Schlesinger, M., *The Specification of Business Rules: A Comparison of Selected Methodologies*, in: A.A. Verrijn-Stuart, T. W. Olle (Eds.), *Methods and Associated Tools for the Information System Life Cycle*, Amsterdam et al.: Elsevier 1994, pp. 29 - 46.
- [HeMy95] Herbst, H., Myrach, T., *A Repository System for Business Rules*, Working Paper 57, Institute of Information Systems, University of Berne 1995.
- [HoKS93] Hoffmann, W., Kirsch, J., Scheer, A.-W., *Modellierung mit Ereignisgesteuerten Prozessketten*, Working Paper 101, Institute of Information Systems, University of Saarbrücken 1993.
- [KnHe93] Knolmayer, G., Herbst, H., *Business Rules*, in: *Wirtschaftsinformatik* 35 (1993) 4, pp. 386 - 390.
- [KnHS94] Knolmayer, G., Herbst, H., Schlesinger, M., *Enforcing Business Rules by the Application of Trigger Concepts*, in: *Proceedings Priority Programme Informatics Research, Information Conference Module 1*, Berne 1994.

- [Lipe89] Lipeck, U.W., *Dynamische Integrität von Datenbanken, Grundlagen der Spezifikation und Überwachung*, Berlin et al.: Springer 1989.
- [MaOd92] Martin, J., Odell, J., *Object-Oriented Analysis & Design*, Englewood Cliffs: Prentice-Hall 1992.
- [McGi92] McGinnes, S., *How Objective is Object-Oriented Analysis?*, in: P. Loucopoulos (Ed.), *Proceedings of the Forth International Conference on Advanced Information Systems Engineering*, Berlin et al.: Springer 1992, pp. 1 - 16.
- [MeBS94] Van Meel, J.W., Bots, P.W.G., Sol, H.G., 'A Hard Core for Soft Problems' - A Business Engineering Case Study within the Amsterdam Municipal Police Force, in: A. Verbraeck, H.G. Sol, P.W.G. Bots (Eds.), *Proceedings of the Fourth International Conference on Dynamic Modelling and Information Systems*, Delft: Delft University Press 1994, pp. 239 - 270.
- [Mori93] Moriarty, T., *The Next Paradigm*, in: *Database Programming & Design* 6 (1993) 2, pp. 66 - 69.
- [Myra94] Myrach, T., *Konzeption und Stand des Einsatzes von Data Dictionaries*, Heidelberg: Physica 1995.
- [Pohl93] Pohl, K., *The Three Dimensions of Requirements Engineering*, in: C. Rolland, F. Bodart, C. Cauvet (Eds.), *Proceedings of the Fifth International Conference on Advanced Information Systems Engineering*, Berlin et al.: Springer 1993, pp. 275 - 292.
- [Poo92] Poo, C.-C. D., *A Framework for Software Maintenance*, in: P. Loucopoulos (Ed.), *Proceedings of the Forth International Conference on Advanced Information Systems Engineering*, Berlin et al.: Springer 1992, pp. 88 - 104.
- [SaVo91] Sandifer, A., Von Halle, B., *Linking Rules to Models*, in: *Database Programming & Design* 4 (1991) 7, pp. 13 - 16.
- [Sche94] Scheer, A.-W., *Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse* 5th. edition, Berlin et al.: Springer 1994.
- [StHH87] Stonebraker, M., Hanson, E., Hong, C.-H., *The Design of the POSTGRES Rules System*, in: *Proceedings of the IEEE International Conference on Data Engineering* 1987, pp. 365 - 374.
- [Stre94] Streng, R.J., *BPR needs BIR and BTR: The PIT-framework for Business Re-engineering*, in: *Proceedings of the Second SISnet Conference*, Barcelona 1994.
- [Tana92] Tanaka, A.K., *On Conceptual Design of Active Databases*, PhD Thesis, Georgia Institute of Technology 1992.
- [TNCK91] Tanaka, A.K., Navathe, S.B., Chakravarthy, S., Karlapalem, K., *ER-R: An Enhanced ER Model with Situation-Action Rules to Capture Application Semantics*, in: T.J. Teorey (Ed.), *Proceedings of the 10th International Conference on the Entity Relationship Approach*, San Mateo: E/R Institute 1991, pp. 59 - 75.
- [TsGH94] Tsalgatidou, A., Gouscos, D., Halatsis, C., *Dynamic Process Modelling Through Multi-Level RBNs*, in: A. Verbraeck, H.G. Sol, P.W.G. Bots (Eds.), *Proceedings of the Fourth International Conference on Dynamic Modelling and Information Systems*, Delft: Delft University Press 1994, pp. 327 - 341.
- [TsKL90] Tsalgatidou, A., Karakostas, V., Loucopoulos, P., *Rule-Based Requirements Specification and Validation*, in: B. Steinholtz, A. Sjølvberg, L. Bergman (Eds.), *Proceedings of the Second Nordic Conference on Advanced Information Systems Engineering*, Berlin et al.: Springer 1990, pp. 251 - 263.
- [VLLS88] Van Assche, F., Layzell, P., Loucopoulos, P., Speltinck, G., *Information systems development: a rule-base approach*, in: *Journal of Knowledge Based Systems* 1 (1988) 4, pp. 227 - 234.
- [WiCL91] Widom, J., Cochran, R.J., Lindsay, B.G., *Implementing Set-Oriented Production Rules as an Extension to Starburst*, in: *Proceedings of the 17th International Conference on Very Large Databases*, pp. 275 - 285.