Business Rules Driven Information Systems Development

On Business Rules Automation: the Business Rules Centric Information Systems Development Framework

prof. dr. Olegas Vasilecas
Vilnius Gediminas Technical University
Information systems department
Senior researcher
Information Systems Research Laboratory
olegas@fm.vgtu.lt

Riga, RTU, Oct. 6, 2009

Related Papers

1. A. Smaizys, O. Vasilecas. Business Rules Based Agile ERP Systems Development. INFORMATICA, ISSN 0868-4952, 2009, Vol. 20, No. 3, 439-460
2. O. Vasilecas, D. Kalibatiene. Towards a Formal Method for Transforming Ontology Axioms to Application Domain Rules. Information Technology and Control, 2009
3. O. Vasilecas, S. Sosunovas. Practical application of BRTL approach for financial reporting domain // Information technology and control. Vol. 37, No 2, 2008, pp. 106-113.
4. O. Vasilecas, D. Bugaite. Applying the meta-model based approach to the transformation of ontology axioms into rule model // Information Technology and Control, 2007, Vol. 36, No 1A, p. 122-125
5. O. Vasilecas, E. Lebedys. Application of business rules for data validation // Information Technology and Control, 2007, Vol. 36, No 1A, p. 273-277.
6. O. Vasilecas, D. Bugaite. Ontology-based Information Systems Development: the Problem of Automation of Information Processing Rules // In E. Neuhold, T. Yakhno (eds.), Proc. of the Fourth International Conference Advances in Information Systems (ADVIS'2006), Izmir, Turkey, 18-20 October, 2006, Springer, LNCS 4243, ISSN 0302-9743, p. 187-196.
7. Valatkaite I., O. Vasilecas. On Business Rules Automation: the BR-centric IS Development Framework. In J. Eder at al. (Eds.): ADBIS 2005, LNCS 3631, Springer-Verlag Berlin Heidelberg, pp. 350 – 365, 2005
8. I. Valatkaite, O. Vasilecas. A Conceptual Graphs Approach for Business Rules Modeling. LNCS 2798, Springer-Verlag 2003, pp 178-189.

- ✓Business rules requirements
- ✓Business rules modelling

Problem

- ✓ The concept of business rule (BR) differs at the levels of business systems and information systems;
- ✓ It is difficult to create business rules statements that would be formal enough and understandable for business users;
- ✓ Visualization of business rules systems is complicated, because most business systems have hundreds or thousands of rules.

Characteristics of BR

- ✓ Broadly speaking, business rules can be considered as constraints;
- ✓ Business rules define the conditions under which a business process is carried out;
- ✓ Business rules define what must be the case rather than how it has do be done;
- ✓ Business rule statements in the business model define the desired logic of the business.

Business rules specification

- ✓ Analyst should clearly specify business logic statements;
- ✓ Business rules specification should be understandable for business users;
- ✓ There must be possible to transform business rules specification into information processing specification in an unambiguous manner;
- ✓ One can start from recording of ordinary business statements.

Principles of defining business rules statements

- ✓ Atomicity – rules can't be broken down any further without losing information;
- ✓ Unambiguous – have only one, obvious, interpretation;
- ✓ Compact – typically, a single short sentence;
- ✓ Consistent – together, rules provide a unified and coherent description;
- ✓ Compatible – use the same terms as the rest of the business model.

Rules can be associated with various aspects of the business

- ✓ Consistency of information;
- ✓ Entity relationships;
- ✓ Identification of situations;
- ✓ Data integrity.

A business rule should not define…

- ✓ Who initiates the rule - this is described in a use case or a process description;
- ✓ When the rule is executed - this is described in a business event, a use case or a process description;
- ✓ Where the rule executes - this will be described in design;
- ✓ How the rule is to be implemented - this will described in design.

Formality levels of rules expression

- ✓ Informal - this provides natural-language statements;
- ✓ Technical - this combines structured data, mathematical operators and constrained natural language;
- ✓ Formal - this provides a more formal defined statements with particular mathematical properties.

Example of rules formality levels

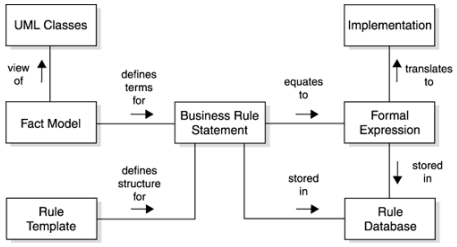
- ✓ Informal expression: A credit account customer must be at least 18 years old;
- ✓ Technical expression:

  CreditAccount

  self.customer.age >= 18;
- ✓ Formal expression:

  { X , Y, (customer X) (creditAccount Y) (holder X Y) }==> (ge (age X) 18).

Forming rule statements



Taken from: T. Morgan. Business Rules and Information Systems:Aligning IT with Business Goals

Pattern conventions

- ✓ Parentheses ( ) enclose a group of terms;
- ✓ Brackets [ ] enclose optional terms;
- ✓ Vertical bars | separate alternative terms;
- ✓ Angle brackets < > enclose special terms.

Special terms 1

- ✓ <det> - determiner (a, the, each or nothing);
- ✓ <subject> - business object or property of an object;
- ✓ <characteristic> - business behavior or relationship among objects;
- ✓ <fact> - relationship between terms identified in the fact model;
- ✓ <fact-list> - a list of <fact> items;
- ✓ <m>,<n> - numeric parameters;

Special terms 2

- ✓ <result> - any value that some business meaning;
- ✓ <algorithm> - definition of the technique to be used to derive the value of a result;
- ✓ <classification> - definition of a term in the fact model;
- ✓ <enum-list> - list of enumerated values.

Classes of rule patterns

- ✓ Basic constraint;
- ✓ List constraint;
- ✓ Classification;
- ✓ Computation;
- ✓ Enumeration.

Example

- ✓ Basic constraint:

  <det> <subject> (must | should) [not]<characteristic>[(if | unless) <fact>].

  An urgent order must not be accepted if the order value is less than $30.

Business rule relationship with fact model

- ✓ Business rules build on a known set of facts described in a model;
- ✓ Business rule associated with particular situation is associated to corresponding fact from fact model;
- ✓ Business rules can be associated with more than one fact.

Fact model

- ✓ Business objects and relationships between business objects are specified;
- ✓ Business objects parameters and possible values are specified;
- ✓ Facts that are realized in IS and facts that are not realized in IS are specified.

Visualization of fact model using class diagram

- ✓ Class model should not be complex;
- ✓ Relationships between classes should not be complex;
- ✓ It is useful to separate business rules from class diagram used to visualize data schema.

Facts in business rule model

- ✓ Single facts can be used to specify business rules;
- ✓ List of facts can be used to specify business rules;
- ✓ Additional operations can be specified when lists of facts are used to describe business rules.

Business parameters

- ✓ Constant parameters are used in rules model;
- ✓ It is useful to specify constant values in model and use them as business parameters;
- ✓ There are business parameters of different types: concrete values, lists, tables and so on.

Potential modelling problems are in the following areas…

- ✓ Facts;
- ✓ Simple constraints;
- ✓ Quantifications and qualifications;
- ✓ States and events;
- ✓ Actors;
- ✓ Dangerous verbs;
- ✓ Computation;
- ✓ Structure and consistency.

Fact model

- ✓ There should not be assumptions in rule statements;
- ✓ Terms should be short and clear;
- ✓ Relationships should be clear.

Simple constraints

- ✓ No permissions. Use of words such as "can" or "may," in a rule are undesirable;
- ✓ Avoid padding;
- ✓ Avoid using “OR”;
- ✓ Avoid using “AND”;
- ✓ Avoid to create complex rules;
- ✓ Avoid starting a rule with "if," which can lead to sentences that confuse the underlying logic.

Quantifications and qualifications

- ✓ Wherever possible, avoid using plurals as terms of rules;
- ✓ Don't be afraid to say each or every if it improves the clarity of the rule.

States and events

- ✓ Avoid using a business event as the subject of a rule;
- ✓ Avoid ambiguous states;
- ✓ Avoid ambiguous time frames;
- ✓ Avoid creating rules that use "when," which implies that an action is fixed to a particular point.

Actors

- ✓ Avoid making actors the subjects of rules.

Wrong: A customer representative may issue a replacement charge card only if the old card expired within the last 30 days

Right: A replacement charge card may be issued only if the old card expired within the last 30 days.

Dangerous verbs

- ✓ Avoid command verb forms;
- ✓ Avoid action verbs, which are likely to create unclear definitions;
- ✓ Look carefully at rules that use words create, read, update, delete or other terms relating to possible implementations .

Computation

- ✓ The essence of a computation can be clarified by making the result of the computation the subject of the rule;
- ✓ Computations embedded within rules should not be used. These should be separated out.

Structure and consistency

- ✓ Missing rules should be taken in mind;
- ✓ To avoid overlapping, rules that relate to the same objects should be checked;
- ✓ Rules that are the same or similar should be removed;
- ✓ Conflicts may arise when two or more rules produce contradictory results;
- ✓ Global consistency with other models should be taken in mind.

Example: application Outlook

- ✓ Rule structure:

a) initiating event;
b) condition/conditions;
c) action/actions;
d) exception/exceptions.

Example: application Outlook 2

- ✓ Conditions, actions and exceptions can be selected from lists;
- ✓ A rule with many conditions can be described as rule with a fewer exceptions;
- ✓ Rule structure:

If (And (c1, c2, c3, ... , Not (Or (e1, e2, e3, ...))))
Then (a1, a2, a3, ...).

Summary of materials presented before

- ✓ Business rules build on a known set of facts described in a fact model;
- ✓ Fact model shows business objects, their relationships, and their attributes;
- ✓ Business parameters are critical values that are meaningful to business;
- ✓ Different aspects of rules construction should be taken in mind.

What we denominate as rules?

- ✓ One extreme: it's possible to see rules everywhere, but this approach isn't practical;
- ✓ Other extreme: it's possible to push all rules down into a general mush, but in this case it's difficult to pick out the logic of business;
- ✓ The right case: It is necessary to draw a line. Just because something could be formulated as a rule doesn't mean to be treated as rule.

Where rules come from?

- ✓ It's useful to understand rules origins;
- ✓ The rules are reflections of the way that the organization works;
- ✓ It is difficult to recognise that a particular piece of text in a document is a business rule;
- ✓ An approach to pulling together the right kind of information is needed.

Main types of rules

- ✓ Structural rules describe constraints or relationships among various elements of a model or system;
- ✓ Behavioral rules define a course of action to be followed in a particular situation;
- ✓ Definitional rules provide a definition of a term or a quantitative or a qualitative relationship between terms.

Sources of information about rules

- ✓ Documentation;
- ✓ Tacit know-how;
- ✓ Automation systems;
- ✓ Business records.

Indicators

- ✓ Features defined by external agencies;
- ✓ Systematic variations among organizational units;
- ✓ Entities with multiple states;
- ✓ Specializations or subclasses;
- ✓ Automated business decision making;
- ✓ Definitions of boundaries;
- ✓ Conditions linked to time;
- ✓ Quality manual;
- ✓ Significant discriminators;
- ✓ Activities related to particular circumstances or events;
- ✓ Information constraints;
- ✓ Definitions, derivations, or calculations.

Important to know

- ✓ It's unlikely that an individual analyst will have the breadth of knowledge and experience to understand the business at the right level;
- ✓ It's important to cover the ground in a systematic way so that no gaps are left;
- ✓ The rules have to be expressed at the correct level;
- ✓ The short timeframes imposed by such demands as e-commerce make it necessary to spread the work over several people, who must be organized as a team to be effective.

Approaches for rule discovery

- ✓ Static analysis is the best approach when relevant documentation is available;
- ✓ Interactive sessions bring together analysts and business specialists to explore areas where business knowledge is not readily available in a documented form;
- ✓ Automated rule discovery can be used in particular cases to find rules through machine analysis, providing that suitable source data can be made available.

Applicability of discovery methods

Source	Static Analysis	Interactive	Automated
Documentation	High	Moderate	Unreliable
Tacit know-how	Not applicable	High	Not applicable
Automation systems	Low	Moderate	High
Business records	Depends on source	Low	Depends on source

Types of source documents

- ✓ Internal sources;
- ✓ External sources.

Internal sources include

- ✓ Specifications covering the system you are analyzing;
- ✓ Slides and other materials from internal training courses;
- ✓ Business plans;
- ✓ Marketing brochures;
- ✓ Internally generated reports;
- ✓ Deliverables produced in earlier projects;
- ✓ Commissioned reports—for example, from a firm of management consultants;
- ✓ Internal directives, manuals, and other forms of guidance provided for staff;
- ✓ Correspondence relating to the system or business areal.

External sources include

- ✓ Legislation affecting the business area concerned;
- ✓ Standards defined by official organizations, such as ISO, or industry groups;
- ✓ Voluntary codes of practice observed within the industry;
- ✓ Standard references for common facts, such as scientific or geographic data;
- ✓ Information produced by postal, telecommunications, and electrical and other utilities;
- ✓ Subscription reports from analyst companies;
- ✓ Magazines, journals, and other publications containing relevant materials;
- ✓ Information provided by suppliers, customers, or business partners.

Analysing documents

- ✓ Try to get an electronic copy of the document;
- ✓ Work through the material in a systematic way;
- ✓ Work in a consistent way;
- ✓ Be sensitive to wording that might imply a logical stat;
- ✓ Aim to get rules expressed in the right form as early as possible;
- ✓ Don't get distracted by with numbering and cross-reference systems;
- ✓ Check for agreement with the current business model as you go along;
- ✓ Be sensitive to political issues;
- ✓ Insist on clarity.

Interactive sessions

Feature	Session Type	
	Interview	**Workshop**
Typical number of participants	Two or three	Six to ten
Typical duration	Half-hour to a few hours	Several days
Areas of business expertise explored	Single	Multiple
Logistics	Simple	Complex
Typical venue	Personal office	Conference room

Summary of materials presented before

- ✓ It's possible to see rules everywhere;
- ✓ It's possible to push all rules down into a general mush;
- ✓ It's useful to understand rules origins;
- ✓ Rules sources: documentation, tacit know-how, automation systems, business records;
- ✓ Approaches for rule discovery: static analysis, interactive sessions, automated rule discovery.

Interactive sessions

Feature	Session Type	
	Interview	Workshop
Typical number of participants	Two or three	Six to ten
Typical duration	Half-hour to a few hours	Several days
Areas of business expertise explored	Single	Multiple
Logistics	Simple	Complex
Typical venue	Personal office	Conference room

Structured interviews

- ✓ Structured interviews;
- ✓ Observe the courtesies;
- ✓ Keep notes;
- ✓ Make records;
- ✓ Feedback.

Analysis workshops

- ✓ Define goal and approach;
- ✓ Prepare for workshop;
- ✓ Conduct the workshop session;
- ✓ Pursue immediate follow-up activities;
- ✓ Follow up with consolidation and research;
- ✓ Review.

Automated rule discovery

- ✓ Automating the process of rule discovery sounds like a grand idea, but it's not one that's generally practical;
- ✓ Even when business knowledge is available, it's expressed in natural language;
- ✓ Machine processing can be used in a few special situations: data mining and code analysis.

Data mining

- ✓ The most common application is the examination of sales records to try to identify interesting customer behavior;
- ✓ This can't be a totally automated process, because many patterns are obvious;
- ✓ A human observer can direct the process, diverting effort away from useless information;
- ✓ Most data mining is carried out using commercial tools.

Code analysis

- ✓ It is good to understand what earlier automation system does;
- ✓ One way is to reengineer the system;
- ✓ Several companies and tools provide the kinds of code-analysis facilities;
- ✓ The most important clues are in the branching structures, the IF-THEN-ELSE constructs;
- ✓ That's not to say that code analysis is a waste of time - it can greatly help in many reengineering processes.

Rule quality

- ✓ Before we can think about technology realization, we need to check the quality of the rules;
- ✓ A quality review should always precede a handover;
- ✓ A well-designed quality system will help to improve your process.

Quality control mechanisms

- ✓ Walkthroughs;
- ✓ Inspections;
- ✓ Testing.

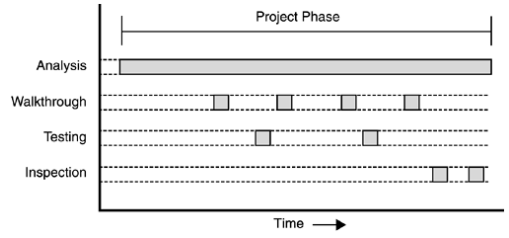
Main perspectives of rules analysis

- ✓ Are the rules correct in a business sense?
- ✓ Are the rule definitions well structured and consistent?

The general pattern to control quality

- ✓ Begin walkthroughs as soon as enough rules have been defined to support a business scenario;
- ✓ Plan for at least one inspection before the end of a phase;
- ✓ Carry out testing as soon as you've defined a rule set that's worth testing.

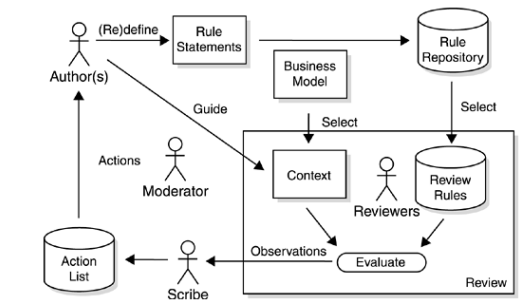
Assessment activity pattern



What to look for in reviewing rules?

- ✓ Rules that are malformed, that don't conform to your local standards or preferred rule patterns;
- ✓ Rules that are incomplete, typically found when a reviewer points out a situation that's not been properly covered;
- ✓ Rules that are inconsistent, leading to ambiguous results with different rules;
- ✓ Rules that are redundant, that serve no business purpose or are covered by another rule;
- ✓ Rules that use terms not properly rooted in the supporting fact model.

Roles



Review outcomes

- ✓ Corrections to the rule statements;
- ✓ Problems with auxiliary information;
- ✓ Missing rules;
- ✓ Problems elsewhere in the business model.

Typical actions

- ✓ Tidying up cross-references between rules and other parts of the business model;
- ✓ Correcting rule statements that aren't sufficiently clear;
- ✓ Adding missing rules to make sure that the logic joins up properly.

Resources (1)

- ✓ A clear statement of the review objective and scope and the meeting format
- ✓ The rules repository, containing the definitions of all the rules to be reviewed, as well as, perhaps, others that are out of scope for the review but that are needed for a full understanding
- ✓ The source business descriptions from which the rules were mined - descriptions of the flow of events, business process descriptions and miscellaneous external specifications.

Resources (2)

- ✓ Other elements of the business model used as a basis for rule definition, including class diagrams, sequence diagrams, state transition diagrams, and so on;
- ✓ Materials that define the business goals and intentions stated by the business owner and the associated levels of business risk that have been deemed acceptable;
- ✓ Results from any testing that's already been carried out, typically relating to complex rule sets.

Perspectives to create rules subsets

- ✓ Focus on business functions;
- ✓ Focus on organizational units;
- ✓ Focus on business objects;
- ✓ Focus on geographical locations;
- ✓ Focus on a business perspective;
- ✓ Focus on specific market offerings
- ✓ Focus on business partners.

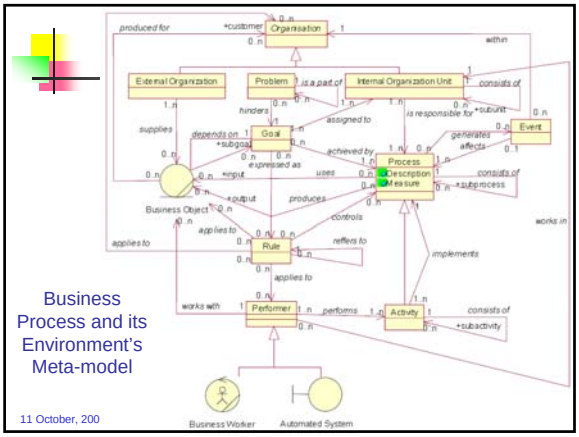
Business Processes and Business Rules

Business Processes and Business Rules

- ✓Business Process (BP) is an abstraction of active part of the business that describes how the work is performed in business environment
- ✓One of the best ways to describe a general concept is to model it and its relationships to other concepts in a form of meta-model used as a pattern to create specific business models.

Business Processes and Business Rules

- ✓Next figure is a meta-model centred on Business Process concept. This meta-model is an UML class diagram in which each concept is depicted as a meta-class, with stereotype icons used in some cases.
- ✓The relationships of the modeling concepts are either an association or specialization.
- ✓The meta-model shows only the most important attributes of Business Process.



Business Process and its Environment's Meta-model

Business Processes and Business Rules

- ✓Business Rules: Business Rules (BR) are statements that can control the execution of the Business Process, affect the structure of the Organisation and its Business Objects and constrain the behaviour of Business Actors.
- ✓Some Business Rules express Business Goals. Business Rules represent business knowledge and can be categorised as functional, structural and behavioural and can refer to each other.

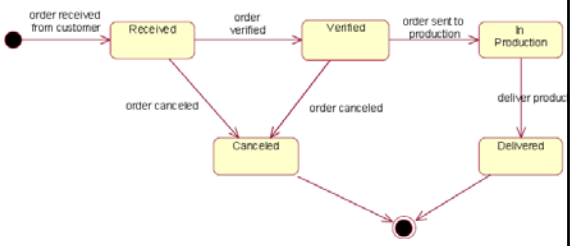
Business Processes and Business Rules

- ✓BR are expressed through various elements of business models such as inheritance, association types and multiplicity, persistence of Business Objects, state chart diagram structure and many others.
- ✓A formal modelling of BR can be done e.g. with UML Object Constrain Language.

Customer Order processing state-chart diagram

- An example of Customer Order state-chart diagram is given in next figure.
- As we can see from this model, *Customer Order can't be cancelled after it has been sent to production* (expression of business level Business Rule).

Customer Order processing state-chart diagram

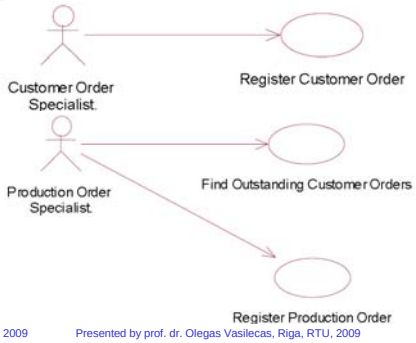


Business Rule: Customer Order can not be canceled at the state "In Production"

Use Cases required to support the Process and the System

- The next model shows a group of Use Cases required to support the Process and the System
- Actors interacting with the IT system.
- Creating a complete and accurate use case model to support Business Processes is a second part of transferring the business models into system models.

Use Case Model to Support 'Place Customer Order' Process



Description of 'Register Customer Order' use case

- The description of 'Register Customer Order' use case (the success scenario if it) may look like as presented below
  - Customer Order Specialist (COS) enters the Customer Order (CO) data into the system.
  - COS requests the system to register CO.
  - The system validates the data entered <E1> and verifies whether CO can be registered <E2>.
  - The system saves CO and confirms this fact to COS.

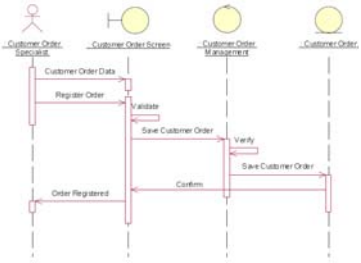
Description of 'Register Customer Order' use case

- Possible exceptions:
  - <E1>: Data entered are incorrect. System displays error message.
  - <E2>: Business rules do not allow the order to be registered (e.g. 'out of stock'). Error message specifying the reason is displayed by the system.

Use Cases required to support the Process and the System

- ✓With use case model in place we have fully transferred the analysis into system specification phase through use case modelling.
- ✓To illustrate the power of this technique, we will show the sequence diagram describing interaction of the System Actor and its three types of classes (boundary, control and entity class) in the realization of this use case.

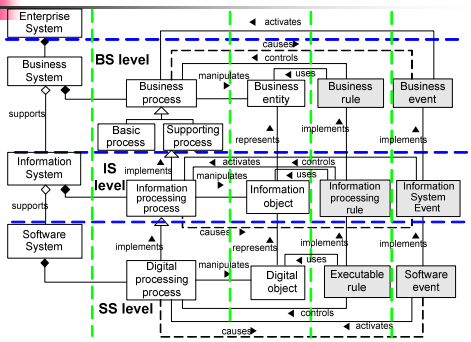
Sequence Diagram: 'Register Customer Order' Use Case Realisation



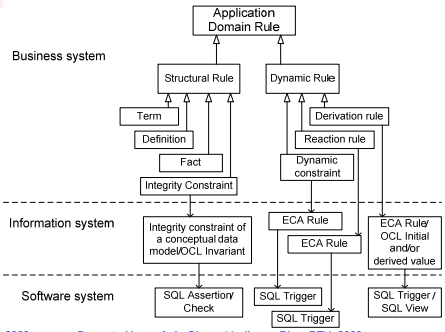
Business rule may not allow the order to be registered after verification (:Customer Order Management / Verify)

- ✓Implementation of application domain rules at different system's abstraction levels

Abstraction levels of systems involved



Application domain rules at different abstraction levels



BR at business system level

- ✓At business system level, application domain rules can be classified to:
  - Structural rules (terms, definitions, facts, and integrity constraints), which can be implemented by a conceptual data model of an application domain, e.g., entity-relationship or UML class model

BR at business system level

- ✓ Implementation of structural rules is defined quite precisely (it can be seen from the precise definitions of integrity constraints in a conceptual data model, like CHECK, DOMAIN, NOT NULL, referential integrity and other constraints)

BR at business system level

- ✓ Dynamic rules, which can be expressed by ECA rules and implemented, like SQL triggers and SQL views (for the case of some derivation rules).
  - A dynamic constraint restricts transitions from one state of the application domain to another.
  - A derivation rule creates new information from existing information by calculating or logical inference from facts.
  - A reaction rule evaluates a condition and upon finding it true performs a predefined action.

BR at business system level

No	Axiom representation in a natural language at Business level	EZPal constraint (representation close to Information system level)	Ontology axioms Classifier
1	The salary of an editor should be greater than the salary of any employee for which the editor is responsible for.	For every instance I1 of Class Editor, if the value of Slot responsible for: Class Editor has instance I2 of class Staff, then Slot salary: Class Editor of I1 has a value > to Slot salary: Class Staff of I2.	CA
2	Every advertisement on the same page must be authored by a different salesperson.	Every Instance of Class Advertisement that share the same value in Slot page_number: Class Advertisement must not share values in Slot salesperson : Class Advertisement.	CA
3	Author cannot be Editor of the same Article.	For every instance of Class Article, Slot author: Class Article and Slot editor: Class Article cannot have the same value.	CA
4	The Newspaper should not include Article, which expiration date (expiration_date) is before (less) then Newspaper's date	For every instance I1 of Class Article, if the value of Slot published_in: Class Article has instance I2 of class Newspaper, then Slot expiration_date: Class Article of I1 has a value less than Slot date: Class Newspaper of I2.	CA
5	No two distinct Articles have the same headline.	Every instance of Article: Class Article has a unique Slot headline : Class Article.	CA
6	The new salary of a reporter equals to the 1.1*old salary of a reporter, if he/she writes more then 16 articles per year.	There is no a template for the implementation of this derivation axiom. Therefore, a template base should be extended by a new template for calculating vales of slots from existing values.	DA (mathema-tical calculation)

- ✓ On Business Rules Automation: the Business Rules Centric Information Systems Development Framework

Outline

- ✓ Motivation to use business rules (BR) information systems (IS) development
- ✓ Benefits of BR approach
- ✓ BR centric IS development framework
  - Levels/Layers
  - Objects and models flow
  - Business rules automation
  - Instantiation of the framework
- ✓ Problems to be solved
- ✓ Conclusions

Introduction

- ✓ Enterprise knowledge based system
- ✓ Intelligent IS
- ✓ Knowledge based IS
- ✓ Rule based knowledge representation
- ✓ Business rules
- ✓ Business rules automation - automatic transformation of specifications from highest modeling abstraction levels to lower
- ✓ Implementation of business rules up to executable code

Business rules and knowledge

- ✓ Knowledge based IS that belong to intelligent IS are widely used in modern business systems.
- ✓ BR are specific and important part of knowledge about business domain.

Motivation to use business rules in information systems

- ✓ The concept of a BR is not a new one, and business representatives actually use it to describe processes and decision making in their organization.
- ✓ BR are relatively new addition in the field of IS engineering. It has been accepted since year 1988 that BR are an important element of all type of IS and BIS as kind of IS therein.
- ✓ Using BR also are the way to structure business knowledge.

Benefits of business rules approach I

- ✓ Benefits from the enterprise management perspective: the explicit business rules model, the elicitation and modelling activities reveal the implicit rules of running the business and serve as the means to first check and then ensure consistency and uniformity of those rules.
- ✓ In particular Business systems are functioning according to BR approved in specific business domain.

OMG and business rule definition

- ✓ 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“.

Business rule and different perspectives

- ✓ BR from database engineering perspective are somewhat similar to
  - DB integrity constraints
  - various programming language constructions used for description of structured logics for making decisions in software systems.
- ✓ Some authors discuss BR similarity to requirements or state that BR can be used to represent both user requirements and conditions to which the system should conform
- ✓ According to the OMG the BR resides at the borderline between business engineering and software engineering.

Business rule and different perspectives

- ✓ The Business Rules Group (BRG) gives BR 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

The concept of business rule I

- ✓From the business system perspective:
  - a business rule is a statement that defines, directs or constrains some aspects of a particular business.
  - business rules are intended to assert the business structure, or to control, or influence the behaviour of the business.
  - at the business system abstraction level, business rules are statements expressing business policies in a declarative manner.

The concept of business rule Ia

- ✓From IS development perspective: the business rules are inherent concept of business itself adopted by IS engineering.
- ✓Thus business rules elicitation and modelling serves as the bridge between business practitioners and IS engineers as the commonly understood language.

The concept of business rule II

- ✓From the ***information systems*** perspective :
  - a business rules from business system are transformed into a statements, which define the major *information processing rules* using some rule-based language.

The concept of business rule III

- ✓ From business rules enforcement perspective: explicit, consistent and uniform business rules model ensures that operational systems do enforce the right *rules* in the right way and conforms with the operating principles of the enterprise
- ✓ This *rules* during implementation process are transformed from business system level to software system level using some transformation procedure.

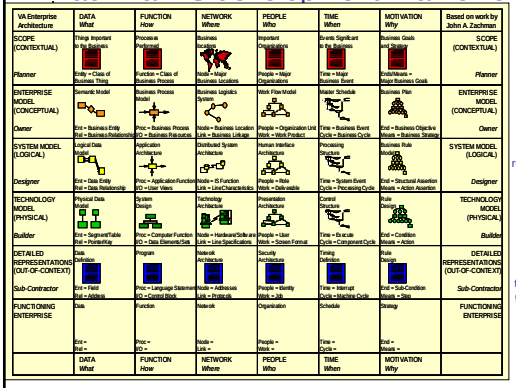
The concept of business rule III

- ✓From an ***implementation*** (software systems) perspective
  - Information processing rules from IS level are transformed into digital information processing rules that can be mapped into structural assertions and dynamic assertions. Dynamic assertions as it was mentioned before smoothly map to the ECA (*event-condition-action,* when *event* occurs, if *condition* is true, do the *action)* rules paradigm.

Motivation to use business rules in information systems

- ✓Part of requirements of the information systems are in the form of, or are related to business rules.
- ✓Structure of business rules are close to the structure of production rules thus it is possible to represent business rules using formal modeling language.

Zachman IS development Framework



List of business goals

Business plan

Business rules model

Business rules design

Rules specification in program logic

Enforced rules

Zachman Framework

Finally, in 1992 John A. Zachman and John T. Sowa could fill out framework matrix, as shown in Figure above. Zachman Framework Motivation column is presented below separately

	Motivation (Why)
Objectives / Scope	List of business goals / strategies
Model of the Business	Business constraints
Model of the Information System	Business rule model
Technology Model	Business rule design
Detailed Representation	Rule specification in program logic
Functioning system	Enforced rules

Business Rules Approach 1

Business need to manage knowledge

- ✓ Business rules approach to IS development emerged as the response to the growing need of business organisations to manage their knowledge explicitly and map them effectively to IS and to *Business Information Systems* in specific in order to support daily business operations.

Business Rules Approach 2

- ✓ Business rules are at the heart of business: they state core business policies, control and influence the behaviour of people, business subsystems, and business processes in the organisation.

Business Rules Approach 3

- ✓ Therefore it is vital to find and capture the rules and ensure that the rules are appropriate. In the process of finding the right rules set, the consensus from all the business stakeholders is obtained on what the rules should be, and conflicting policies are reconciled.
- ✓ Business rules are:
  - "What" of business
  - "How" of business

Business Rules Approach 4

It is important for business adapt to changing environment

- ✓ As the business changes the rules set must be properly maintained and adapted to the new conditions/requirements.
- ✓ These activities carry the label of business modeling and answer the question of "what" – what the business is.

Business Rules Approach 5

- ✓ The second question is "how" – how to use the knowledge obtained in the first step? There are *two ways* how to put business rules to work.
- ✓ *First one* is simply to use business rules for guidance of people in the business.
- ✓ *The second one* comprises the technology, using, when a substantial subset of discovered and agreed business rules can be automated in computerised information systems.

Business Rules Approach 6

- ✓ In this presentation our focus is on the second way.
- ✓ By the way we look at what business rules are, what types of business rules can be enforced into business IS, and what technologies can be employed for business rules implementation.

Business Rules Approach 7

- ✓ A business IS development framework is proposed, which defines methods to analyse, model and structure business knowledge represented by BR, and map the resulting business rules model to the implementation level.

Taxonomies

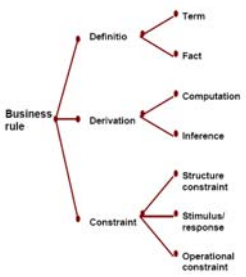
- ✓ There is no agreed-upon taxonomy for business rules, nor does there need to be (Ellen Gottesdiener 2002)
- ✓ 15 taxonomies

BR classification by R. Ross

- ✓ Facts,
  - VAT in Lithuania is 21%
- ✓ Terms,
  - Acceptable profit is > 30%
- ✓ Rules,
  - *If profit < 30% detail sales analysis is needed*
- ✓ Constraints,
  - *Liabilities < asset/2*
- ✓ Derivations,
  - *Total Item price = price + VAT*
- ✓ Inferences,
  - *If total price > 1000 EUR, apply discount*
- ✓ Timing,
  - *Insurance contract should be signed in 9 min.*
- ✓ Sequence,
  - *Enter risk factors, Print contract, sign ...*
- ✓ Heuristics.
  - *Friday 13th is bad day for business*

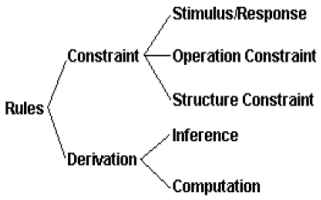
R. Ross

Taxonomies



Taxonomies

Odell suggests the following taxonomy of business rules:



This taxonomy maps well to standard information concepts, including OO concepts. For example, operation constraints are pre- and post-conditions on object operations, and structure constraints are class invariants.

Taxonomies

- ✓ However, stimulus/response rules, which state that when something occurs, something should be done, are not commonly supported by OO languages and are typically implemented outside of an OO application as triggers in the database.
- ✓ On the other hand, knowledge-based tools provide software support for stimulus/response rules using demons.

Taxonomies

- ✓ Derivation rules derive a fact about an object.
- ✓ Computational business rules state how something is calculated; they are typically allocated to procedural code in object operations.
- ✓ Inference rules may be associated with rules in knowledge-based systems.

Taxonomies

- ✓ Object-oriented techniques often ignore inference and reduce inference rules to procedural code, thereby violating the declarativeness of business rules.
- ✓ In order to separate code and data from rules, and add flexibility to systems development, rule engines and rule management systems have to be added. (“Knowledge-based systems did not really go away. They went undercover").

Taxonomies 2

Source	Business rules taxonomy
Business Rules Group (1995, revised in 1997 and 2000) [3]	• term (a type of structural assertion) • fact (terms linked together; another type of structural assertion) • derivation (facts derived from other facts) • constraint (also called action assertions)
Graham (1998) [8]	• attribute assertion (range constraints, enumeration constraints, type constraints) • operational assertion (pre-condition, post-condition, invariance condition) • class invariant
Morgan (2002) [10]	• basic constraint • list constraint • classification • computation • enumeration
Ross (2001) [13]	• rejecter (constraint) • producer (subcategories: computation and derivation rules) • projector (stimulus/response rule which include enabler rules, • copier rules and executive (trigger) rules)

Taxonomies 3

Source	Business rules taxonomy
Ross (2001) [14]	• term • fact • computation • constraint • inferred knowledge • action business rule
Barbara von Halle [23]	• definition • fact • constraint • derivation • inference
Barbara von Halle (2002) [22]	• terms • facts, *and 5 type of rules*: o mandatory constraint o guideline o action enabler o computation o inference

Taxonomies 4

Source	Business rules taxonomy
Date (2000) [5]	• constraint • state constraint • transition constraint • stimulus/response • derivation • computation • inference
James Odell [12]	• constraint (state constraint, transition, event/action • or/and stimulus/response, operation, structure) • derivation (computations, inference) *Notes: rules can be global, local, and/or temporal*
Dan Tasker, Air New Zealand [18]	• action restricting • action triggering • constraint

Taxonomies 5

Source	Business rules taxonomy
Margaret Thorpe, Tangram [16]	• definitions • basic integrity constraints • general declarative constraints • procedural constraints • inferential • derivation
Ellen Gottesdiener [7]	• term definitions • facts relating terms • constraints • action enablers • derivations • inferences

Structural and Dynamic Assertions 1

- ✓ Presented list is quite extensive, it covers the opinions of business rules researchers working in this field.
- ✓ We state that from implementation perspectives the business rules categories can be divided into two broad types:
  - structural assertions
  - dynamic assertions

Structural and Dynamic Assertions 2

- ✓ Terms, facts, and state integrity constraints can be considered as structural assertions, which introduce the definitions of business entities and describe the connections between them.
- ✓ Since they can be captured by a conceptual model of the problem domain, e.g. by an Entity-Relationship or a UML class model, they can be regarded not as business rules, but rather as concepts forming the business vocabulary (or ontology).

Structural and Dynamic Assertions 3

- ✓ Other categories can be considered as dynamic assertions, which fall into the following main types:
  - dynamic constraints restrict the admissible transitions from one state of the system to another;
  - derivation rule is a statement of knowledge that is derived from other knowledge by an inference or a mathematical calculation;
  - reaction rules are concerned with the invocation of actions in response to events, they state the conditions under which actions must be taken.

Dynamic assertions and ECA paradigm

- ✓ The rules of the last category (dynamic assertions) directly fall under ECA paradigm (when *event* occurs, if *condition* is true, the *action*) from active databases field

Dynamic assertions smoothly map to ECA paradigm 1

- ✓ All business rules operate on data (are data driven) and are triggered by data state transitions.
- ✓ Therefore each business rule has an event, which may be stated explicitly or assumed implicitly. For the latter case implicit event is formulated as the appropriate data state transition.

Dynamic assertions smoothly map to ECA paradigm 2

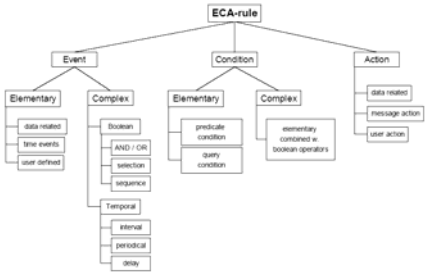
- ✓ Some business rules have explicit condition, some do not. The missing condition can always be substituted with a default condition stated as *TRUE*.

Dynamic assertions smoothly map to ECA paradigm 3

- ✓ All dynamic assertions have defined action – it is at the heart of reaction and derivation rules.
- ✓ Dynamic constraints may not have the explicit action, since they can state what transition from one data state to another is not admissible.
- ✓ The action for this type of rules could be defined as corrective actions or presenting the appropriate instructions to the business user.

ECA rule components

ECA components can be divided further into subcategories according to their characteristics



Short description of ECA rule components

This detailed classification is needed at the design and implementation level in order to enable different treatment of different types of rules in the database. The following gives a short description of each category.

- ✓ Events can be either elementary or complex. A complex event is composed of elementary events but has additional properties, that are lost when decomposing it to elementary events.
- ✓ Elementary events can be classified as data related events, time events or user defined events.

Short description of ECA rule components

- ✓ *Data Related Events* are events that define the occurrence of data manipulation, e.g. insert, update and delete.
- ✓ *Time Events* are events that occur at one given point in time, e.g. at 10:00 am, June 1st 1996.
- ✓ *User Defined Events* are those events that can neither be classified as Data Related nor Time Events, e.g. customer calls to complaint about a delivery.

Short description of ECA rule components

- ✓ Complex events are either *boolean* events or temporal events. The temporal events can be divided further into subcategories as follows.
- ✓ *Interval Event* (Temporal) are events occurring between two other events, e.g. reception of payment BETWEEN invoice sent AND 30 days after invoice sent
- ✓ *Periodical event* (Temporal) are reoccurring events with regular intervals, e.g. once a month.
- ✓ *Delay event* (Temporal) are event that occur after another event has occurred, with a specified delay.

Short description of ECA rule components

- ✓ Conditions can be used to further specify the situation under which the action is to be taken.
- ✓ Like with events, conditions can be either elementary or complex, where a complex condition is elementary conditions combined with boolean operators.
- ✓ The elementary conditions can be divided into *predicate condition* and *query conditions*.

Short description of ECA rule components

- ✓ Actions define what is to be done when the event occurs and if the condition values true.
- ✓ The actions are divided into similar categories as elementary events, i.e. data related and user actions, as actions can raise an event in other rules.

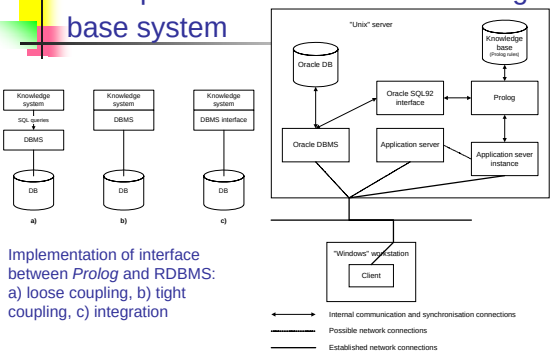
Short description of ECA rule components

- ✓ *Data Related Actions* manipulates data, e.g. insert, update, delete, etc.
- ✓ *Message Actions* are performed by users or applications and contain message to a user.
- ✓ *User Actions* are those that can not be classified as a Data Related Action or a Message Action.

On BR-centric Frameworks I

- ✓ According to the survey of BR-centric frameworks, architectures and technologies, the proposed ideas are rather diverse and can be summarised as follows:
  - From implementation perspective the proposed business rules approaches can be classified into three broad types:
    - rules implemented as application logic components,
    - rules implemented using active databases technologies,
    - rules implemented in rules engines (enforcement, inference, etc).

Example of architecture of knowledge base system



Implementation of interface between *Prolog* and RDBMS: a) loose coupling, b) tight coupling, c) integration

On BR-centric Frameworks II

- ✓ According to the survey of BR-centric frameworks, architectures and technologies, the proposed ideas are rather diverse and can be summarised as follows:
  - From architectural focus: different authors stress different IS development life cycle phases – from elicitation to maintenance; accordingly their proposed frameworks vary. Some concentrate on business objects definitions and modelling, others go for automatic implementation frameworks and technologies.

On BR-centric Frameworks III

- ✓ According to the survey of BR-centric frameworks, architectures and technologies, the proposed ideas are rather diverse and can be summarized as follows:
  - From modelling perspective: a lot of attention is paid to the modelling issues of business rules. Some proposed modelling techniques, for example, by Ross, are both modelling language and modelling method in one. Another approaches stem from adapting popular modelling languages, such as UML and OCL, to business rules modelling activity. However, none of the proposed languages or methods are accepted as technology standard yet.

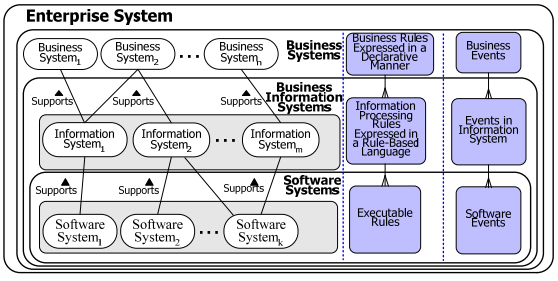
The BR-centric IS Development Framework I

- ✓ The proposed BR-centric IS development framework has several intended purposes:
  - to describe the set of information objects and related modeling activities in the process of BR-based IS development;
  - to give the information objects flow through different abstraction layers coming from the most abstract – business systems layer to the implementation layer;

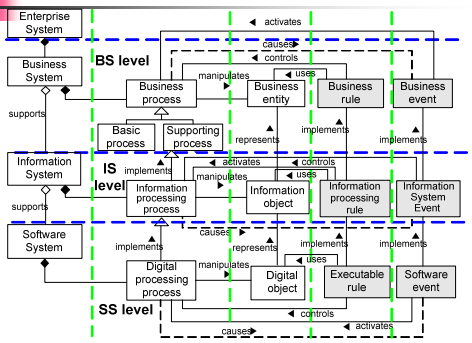
The BR-centric IS Development Framework I I

- ✓ The proposed BR-centric IS development framework has several intended purposes:
  - to give the information objects models flow through the same sequence of layers;
  - to provide the possibility of tracing the flow of objects from business system down to the implementation layer and backwards in order to ensure the consistency between the concepts, especially traceability is required upon making any modifications in any abstraction layer – the corresponding modifications must be traced back to the modification origin.

Enterprise system and business rules



Abstraction levels of systems involved



Business rule: term I

- ✓ Actually the term “business rule” currently is used in very disparate ways:
  - to refer the business rules in real business environment,
  - to refer actually the information processing rule at information systems level, and
  - even sometimes to refer the implementation (software) level objects corresponding to business rules.
- ✓ We propose to distinguish among the mappings of the business systems level concepts and their mappings in subsequent levels.

Business rule: term II

- ✓ We state that business rules exist only in business system environment and they can be mapped to information processing rules in information system environment.
- ✓ That is, we do not have “business” as such at information system level – we have only mappings of the business supporting information processing and flow.

Layers Description I

- ✓ Business system abstraction level represents the real objects – this is the business system itself, which can be defined as the closed world with the input and output channels to and from its environment.
  - The business objects under concern are those, which should be mapped to information system level objects in order to simulate the natural information processing with the mappings in information system objects.

Layers Description II

- ✓ Information system abstraction level represents the information flows among the information objects. It deals with informational mappings of the business objects from business system level. Therefore it may be called the informational mapping of the real business system.
  - The type of objects at information system level is information processing objects. These objects are also models of the business objects because they represent business objects only for their intended purpose.

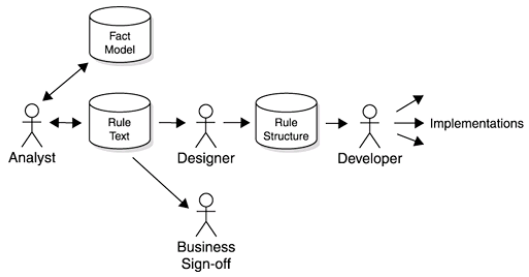
Layers Description III

- ✓ The information objects models are later used at the third abstraction level – information system implementation level, where information objects are mapped to digital objects and data flows among the objects are implemented using the specific technology and architecture.

Business rules automation

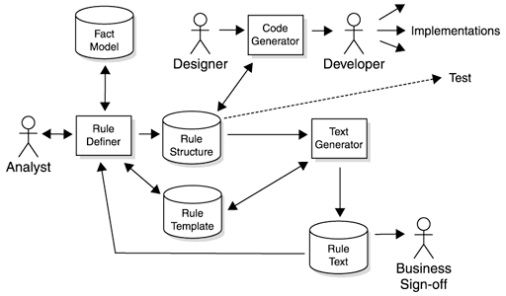
- ✓ Different technology can be used for rule definition and implementation
- ✓ Business rules automation (automatic transformation of specifications from highest modeling abstraction levels to lower) is possible only in case of using formal specifications of business rules.

Low-technology rule definition and implementation



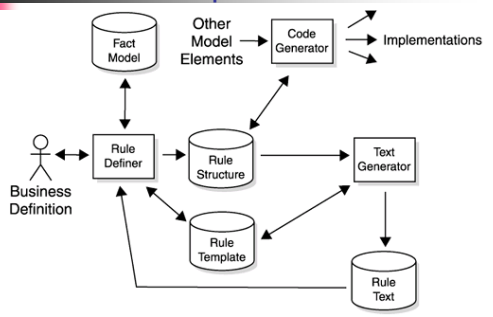
T. Morgan. Business Rules and Information Systems: Aligning IT with Business Goals, Addison-Wesley, 2002.

Controlled rule definition and implementation



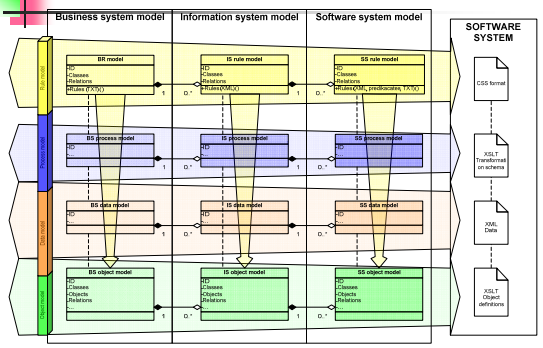
T. Morgan. Business Rules and Information Systems: Aligning IT with Business Goals, Addison-Wesley, 2002.

High-technology rule definition and implementation

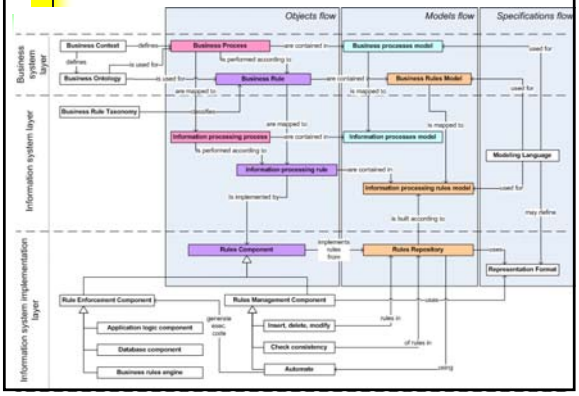


T. Morgan. Business Rules and Information Systems: Aligning IT with Business Goals, Addison-Wesley, 2002.

(Divagation) BR model in engineering process



The BR-centric IS Development Framework



Objects and Models Flow I

- ✓ Business objects at business system level are the business environment objects, such as the physical order placed for the goods, etc. The purpose of having the business objects in the defined business context is to draw the boundaries of the business system in scope.

Objects and Models Flow II

- ✓ Information objects at IS level are the information maps of the business objects in the information system level . Their intended purpose is to represent the relevant informational properties of the business objects that are of importance to the information flows described in information system.

Objects and Models Flow III

- ✓ In the third layer, which can be regarded as the physical level – information system implementation as software components level – the information objects are mapped to digital objects and implemented as various software system components.

Application of the framework using business system example I

- ✓ Since the framework proposed is too generic for the direct application, we have instantiated it using the specific components instead of generic ones (where it is necessary):
- ✓ The business context used: Production Ordering System;
- ✓ The business ontology was not used for this phase of the instantiation, however, specific business domains ontologies should be investigated as sources of standardised set of business objects (entities) for modelling purposes;

Application of the framework using business system example II

- ✓ Conceptual graphs were used for modelling both business system and information system layers objects;
- ✓ Rules repository was created in active database management system;
- ✓ Rules automatic generation component was developed implemented for the rules enforcement in active database system as triggers;

Application of the framework using business system example III

- ✓ The business processes modelling activity was omitted for the first application of the framework, however, the business processes modelling is a compulsory activity for traceability purposes. By traceability we mean the tracking of changes from business system level down to the implementation level.

Modeling languages can be used

- ✓ UML and OCL
  - UML is a de facto standard
  - OCL is a logic based language
  - OCL does not have any graphical notation
- ✓ Ross method
  - both a language and a method
  - only graphical notation
  - on top of data model
- ✓ Oracle RuleSLang
  - vendor specific
  - OCL subset: lack of graphical notation
- ✓ Systems of logic

Possible business rules representation

- **PROLOG** (*PROgramming in LOGic*), predicate calculus , …

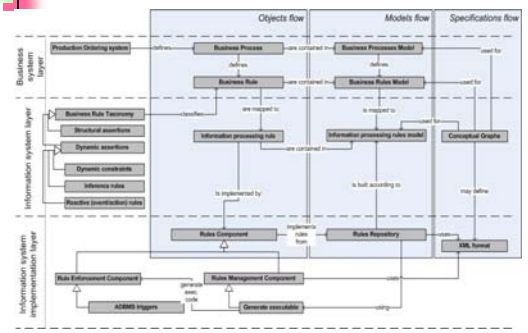
  get_discount_of(person, discount) …

- **KIF** (*Knowledge Interchange Format*) - JESS

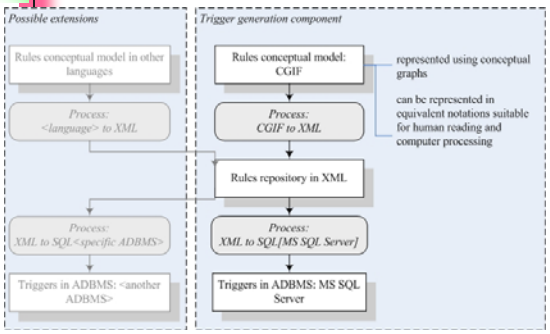
(=> (and (real-number ?x) (even-number ?n)) (> (expt ?x ?n) 0))

- **XML** (eXtensible Markup Language)
  - **RuleML** - IBM CommonRules,
  - **SRML** (*Simple Rule Markup Language*) - ILOG Jrules,
  - Other rule representation languages based on XML.
- **Conceptual Graphs**

Instantiation of the Framework



Trigger Generation Component I



Trigger Generation Component II

- ✓ In the current research we concentrate on the business rules model represented in conceptual graphs CGIF notation (Conceptual Graphs Interchange Format). However, the architecture is not limited in that sense and the extensions are possible (shown in on the left):
  - Usage of another modeling language for business rules conceptual model;
  - Usage of another underlying active database management system (because of syntactic differences of trigger definition in different active database management systems).

Trigger Generation Component III

- ✓ The component consist on two processes
- ✓ First process CGIF2XML. The scope of the process is a transformation of business rules representation from conceptual graphs model in CGIF format to the intermediate business rules representation in XML. The latter can be used as the basis for the business rules repository. In current implementation the input format is limited to CGIF; the output format adheres to the rules representation using XML structure.

Trigger Generation Component IV

- ✓ Second process XML2SQL. The scope of the second process is a transformation of rules contained in intermediate format of XML based structure to the actual implementation representation – namely, active databases triggers. In the current implementation the Microsoft SQL Server active database management system trigger definition syntax is used.

Trigger Generation Component V

- Resulting trigger: The resulting trigger implementing the business rule is shown below. The trigger is generated using MS SQL Server trigger definition syntax.

```
CREATE trigger Rule1 ON ManufOrder
FOR UPDATE
AS
IF UPDATE(status)
BEGIN
 if((SELECT status FROM INSERTED)='Confirmed' and (SELECT id FROM
INSERTED)+2>15 )
   BEGIN
     UPDATE ManufOrder
     SET Finish_date=getdate()
     WHERE id=(SELECT id FROM INSERTED)
  END
END
```


Example for future work

- Telecommunications domain, task: implement analytics into self-service platform
- Analytics = rules sets (business rules set at business system layer)
- Example:
  - Customers that generate international outgoing traffic more than x minutes/month for 3 months, should receive the offer to upgrade to international price plan.

Problems to be solved

- Nevertheless, in IS development (ISD) practice business rules are rarely used in a way where they would be separated from components that are not relevant to the business knowledge, documented so that they would be traceable from/to their sources, and implemented so that would be 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

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: what exactly are business rules?

- One of the frequently held misconceptions about business rules is that all the rules governing an application are business rules.
- The relationship between business rules 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 a rules.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: 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, rules, etc.
- Contrariwise, tacit knowledge is not easily expressed and visible

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: business rule specification and modelling

- Once business rules are acquired, they have to be specified in an appropriate manner. This 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 different roles in the ISD lifecycle.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: business rule implementation I

- ✓ 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).
- ✓ They range from database-oriented tools that enforce rules using database mechanisms to rule oriented systems that offer declarative rule specification languages and special mechanisms to take care of the rule execution.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: business rule implementation II

- ✓ 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.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: enterprise-wide business rule management I

- ✓ While there is a variety of tools and approaches that provide IS developers with facilities for managing business rules through ISD, there is only limited support for an enterprise-wide rule management.
- ✓ Business rules, however, do not pertain to IS or to its application software. Business rules are set by the business and have to be therefore managed by the business.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Problems to be solved: enterprise-wide business rule management II

- ✓ Changes in organisation's business environment almost never happen spontaneously, but are typically driven 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.

Marko Bajec at al., State of the art and remaining challenges in the business rule-based ISD

Conclusions I

- ✓ We have presented a generic BR-based IS development framework which can be instantiated based on available techniques, methods or languages. We have carried out the experiment of framework application substituting the generic framework components with specific ones – such as selecting the ADBMS technology for rules repository implementation, conceptual graphs for business rules and information processing rules modelling, etc.

Conclusions II

- ✓ The other motivation for this paper and the framework itself was to show the diverse concepts depending on the abstraction level and the flow and mapping scheme of the objects from layer to layer. We have differentiated three layers – that of business system, information system and computerised information system implementation – software system, and have shown how the business objects map to information objects and to executable specifications.

Conclusions III

- The proposed BR-centric IS development framework enables business rules automation activity as the integral part of the development cycle; the traceability issues, although supported by the framework, were not elaborated in this paper and remain one of the important future research directions.
- The next step in this research should be further refinement of the objects mapping and transformations in different abstractions layers and the full case study employing the proposed concepts and ideas.

References 1

- [1] ACT-NET Consortium: The Active Database Management Systems Manifesto: A Rulebase of ADBMS Features. ACM Sigmod Record, Vol. 25(30) (1996) 40-49
- [2] Bajec, M., Rupnik, R., Krisper, M. Using Business Rules Technologies To Bridge The Gap Between Business And Business Applications. In Proceedings of the IFIP 16th World Computer Congress 2000, Information Technology for Business Management (G Rechnu, Ed), Beijing, China, 2000, pp. 77-85
- [3] Business Rules Group. Defining Business Rules ~ What Are They Really? (formerly known as the "GUIDE Business Rules Project Final Report," November 1995), Business Rules Group, (3rd Ed.), July 2000. (Available at www.businessrulesgroup.org.)

References 2

- [4] Chein, M., Mugnier, M.L. Conceptual Graphs: Fundamental Notions. Revue d'Intelligence Artificielle, Vol. 6, No.4 (1992) 365-406
- [5] Date C.J. What Not How: The Business Rules Approach to Application development. Reading. Mass.: Addison-Wesley Longman, Inc.
- [6] Gottesdiener, E. Top Ten Ways Project Teams Misuse Use Cases -- and How to Correct Them. Part I: Content and Style Issues. The Rational Edge, June 2002, URL: http://www.therationaledge.com/ content/jun_02/ t_misuseUseCases_eg.jsp.
- [7] Gottesdiener, E. Business RULES Show Power, Promise, Application Development Trends, March, 1997 Volume 4, Number 31, March 1997: see: http://www.ebgconsulting.com/publications.html#business_rules

References 3

- [8] Graham, I. Requirements Engineering and Rapid Development, Addison-Wesley, 1998
- [9] Lin, N. Alternatives For Rule-based Application Development, Business Rules Journal, Vol. 3, No. 10 (October 2002), URL:http://www.BRCommunity.com/a2002/n007.html.
- [10] Morgan, T. Business Rules and Information Systems: Aligning IT with Business Goals, Addison-Wesley, 2002.
- [11] Object Management Group. Business Rules in Models: Request for Information. URL: http://cgi.omg.org/cgi-bin/doc?ad/2002-9-13 (2002)

References 4

- [12] Odell, J. Business Rules, Object Magazine, February, 1995 republished in Wisdom of the Gurus: A Vision for Object Technology, Charles Bowman, ed., SIGS Books, 1996. and in Advanced Object-Oriented Analysis & Design using UML, SIGS Books, 1998.
- [13] Ross, R.G. The Business Rules Classification Scheme, DataToKnowledge Newsletter Volume 29, Number 5, 2001. See: http://www.brcommunity.com/a2001/b086.html
- [14] Ross, R.G., and Lam, G.S.W. RuleSpeakTMSentence Templates: Developing Rule Statements Using Sentence Patterns, Business Rule Solutions, 2001. Downloadable via: http://www.brsolutions.com/rulespeak_download.shtml

References 5

- [15] Ross, R.G., The Business Rule Book: Classifying, Defining and Modeling Rules, Database Research Group, Boston, MA, 2nd edition, 1997
- [16] Sobieski, J, Krovvidy, S, McClintock, C., and Thorpe, M. KARMA: Managing Business rules from Specification to Implementation, paper presented at American Association of Artificial Intelligence Conference, Innovative Applications of AI track, Portland, Or, 7/96. Business Rule Summit, February 26-28, 1996, Miller Freeman Inc., 600 Harrison Street, San Francisco, CA 94107
- [17] Sowa, J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pasific Grove et al. (2000)

References 6

- [18] Tasker, D. The Problem Space: Practical Techniques for Gathering and Specifying Requirements using Objects, Events, Rules, Participants and Locations, 1993, electronic book, self-published by the author: Dan Tasker, 100241.2337@compuserve.com
- [19] Valatkaite, I., Vasilecas, O. Deriving active database triggers from business rules model with conceptual graphs. In Lithuanian Mathematical Journal, Vilnius, MII, t. 42, special issue "Proceedings of the XLIII conference of Lithuanian Mathematical Society", pp. 289-293, 2002 (in Lithuanian)
- [20] Valatkaite, I., Vasilecas, O. Application Domain Knowledge Modeling Using Conceptual Graphs. In: Kirikova, M. (eds.): Information Systems Development: Advances in Methodologies, Components and Management. Kluwer Academic/Plenum Publishers, 2002, pp. 193-202.

References 7

- [21] Valatkaite, I., Vasilecas, O. A Conceptual Graphs Approach for Business Rules Modelling. Accepted by Seventh East-European Conference on Advance in Databases and Information Systems (ADBIS), September 3 - 6, 2003, Dresden, Germany. To be published in Lecture Notes of Computer Science, Springer-Verlag, 2003.
- [22] Von Halle, B. Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons, 2002
- [23] Von Halle, B. The business rule roadmap. Database Programming and Design, 10(10), 1997, pp. 13-15.
- [24] Zachman, J.A. A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26, no. 3, 1987. IBM Publication G321-5298.

Thank you for your attention and patience !

Your questions please?