An international journal

INNOVATIVE INFOTECHNOLOGIES FOR SCIENCE, BUSINESS AND EDUCATION

Vol. 1 (6) 2009

Cover article **Leena Laari-Muinonen, Kirsi Viskari**
EXPERIENCES IN TEACHING INTERNATIONAL BUSINESS WITH BUSINESS SIMULATION GAME

Hot article **Kęstutis Normantas, Olegas Vasilevas**
MODELLING OF THE BUSINESS RULES USING UML/OCL

VILNIUS BUSINESS COLLEGE

ISSN 2029-1035

International journal
INNOVATIVE INFOTECHNOLOGIES FOR SCIENCE, BUSINESS AND EDUCATION
is published quarterly by Vilnius Business College.
Journal (first published in 2007) is dedicated to new concepts and new insights
into the issues of computer technology applications.
Short communications, review articles and full-length articles
combining theoretical approaches and experimental applications are welcome.
The main emphasis is placed on the novelty, the so-called „first idea" of a new paradigm
in the field of advanced programming, business realization, teaching tools etc.

TOPICS

Advanced programming
Conceptual Modelling and Databases. Natural Language Processing.
Formal Analysis and Design Methods. Algorithms and Applications. Bioinformatics

Information Technology Applications
E-Government. Grid computing systems. High-performance scientific computing
IT - business solution
Business Process Modelling, Analysis and Design. Information and Software Systems Engineering

New teaching technologies
Artificial Intelligence and Knowledge Engineering. IT in Teaching and Learning.
Computer-based Learning. Distance Education. E-tools

Hardware and Computer Networks
Security and Privacy

Submissions and subscription orders to be directed to:
Vilnius Business College, Kalvarijų str., 125, LT-08221, Vilnius, Lithuania,
Email: infotechnologies@kolegija.lt, http://journal.kolegija.lt

Print UAB „Sapnu sala"
Circulation 50
Quire 10
Designer S. Kuznecovas

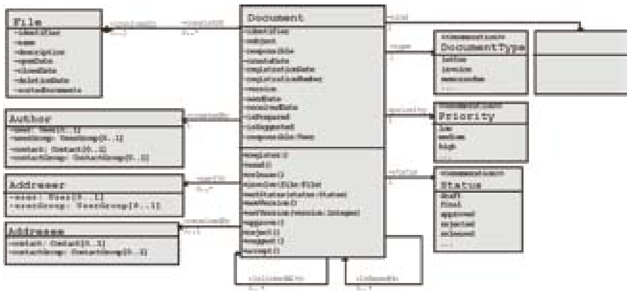
INNOVATIVE INFOTECHNOLOGIES FOR SCIENCE, BUSINESS AND EDUCATION

Vol. 1 (6) 2009

CONTENTS

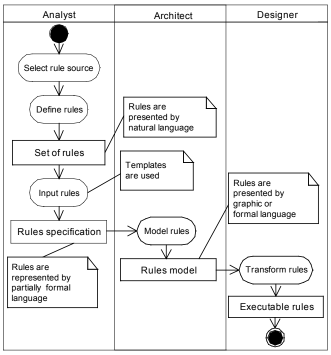
2.1–10 Kęstutis Normantas, Olegas Vasilecas

MODELLING OF THE BUSINESS RULES USING UML/OCL



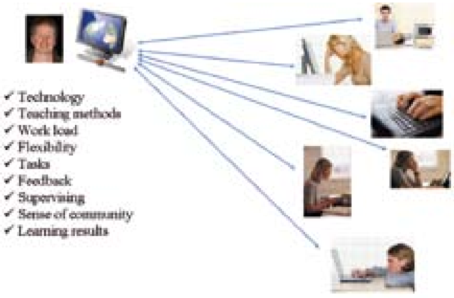
3.1–5 Rūta Dubauskaitė, Olegas Vasilecas

AN OPEN ISSUES IN BUSINESS RULES – BASED INFORMATION SYSTEM DEVELOPMENT



4.1–5 Leena Laari – Muinonen, Kirsi Viskari

EXPERIENCES IN TEACHING INTERNATIONAL BUSINESS WITH BUSINESS SIMULATION GAME



5.1–5 Tomas Butvilas

VALIDITY OF PSYCHOMETRIC TOOLS IN SOCIAL SCIENCES: MEASURING GRIEVING CHILDREN VALUE SYSTEM

$$\alpha = \frac{N}{N-1}\left(1-\frac{\sum_{i=1}^{N}\sigma^{2}_{Y_i}}{\sigma^{2}_{X}}\right)$$

MODELLING OF THE BUSINESS RULES USING UML/OCL

Kęstutis Normantas*, Olegas Vasilecas

*Information Systems Research Laboratory,
Vilnius Gediminas Technical University,
Sauletekio 11, LT20221, Vilnius, Lithuania*

Received Februar 6, 2009 , accepted August 21, 2009

Abstract. Business rules are a crucial business category because they describe how enterprises are conducting business. Their value in developing software systems, which must be susceptible to fit rapidly changing business requirements, has made them attractive also within information system domain. As the formalization of business rules becomes a part of the commonly practiced systems analysis process, it is desirable for there to be a single, coherent representation for all kinds of business rules. The *Object Constraint Language* (OCL) as a part of the *Unified Modelling Language* (UML) provides the possibility to express business rules in formal and unambiguous manner. In this paper we investigate possibilities how to express different kinds of business rules with the UML/OCL, and discuss their advantages and disadvantages.

Keywords: Business rules, UML, OCL

Short title: Modelling business rules

* *Corresponding author, email: kestutis.normantas@isl.vgtu.lt*

Introduction

Business success requires flexibility to fit rapidly changing business environment. Therefore, business supporting information systems (IS) should be able to adjust to those changes in time. Unfortunately, existing systems development approaches are not so flexible as business requires. It is difficult to readjust traditional business supporting IS to business changes because this process involves many troublesome tasks, e.g. revision of the system specification, redesigning, recoding etc. More often, in order to satisfy business needs in time, changes are made directly in code, without additional documentation. Later these changes grow over to a headache to those who are responsible for their management, and system specification looses its significant value. Moreover, Morgan in Ref. [1] notices that many research projects have shown that vast majority of software problems originates from specification error, not from the code as such. Therefore, demand on different approach to systems development arises.

According to the well-known Zachman Framework – see Ref. [2], in IS development process every system is represented by a single or several models depending on different aspects of modelling system and different points of view to the system. The business rules (BR) as their predecessors (business scope, motivation and strategy) and as their successors (business rules model and executable code) are concurrent with other aspects of enterprise system (data, functions, places, people and time). Therefore, it is very important that integration of BR model with other system's models as well as clear and unambiguous understating of BR and possibility to access and manage them should be guaranteed.

Regarding Barbara von Hale – see Ref. [3], an enterprise operates according to many different kinds of rules, such as legal mandates and rules it constructs for itself. The basic element of a BR is the language used to express it [4]. The most understandable form of BR is natural language, however, this form is ambiguous and informal to use BR in IS development process. As the formalization of business rules becomes a part of commonly practiced systems analysis process, it is desirable to be a single, coherent representation for all kinds of business rules.

The *Unified Modelling Language* (UML) – see Ref. [5] – has established itself as the leading object-oriented (OO) analysis and design methodology. UML is used for modelling systems within different abstraction levels [5]. The *Object Constraint Language* (OCL) has been developed as business modelling language within IBM Insurance division [6]. Recently, the second version of the OCL has been adopted as a part of the UML standard. The OCL supplements the UML methodology with possibility to specify system models in more detailed and unambiguous manner. According to Ref. [7], OCL is easy to use for an average business or system analyst, because its syntax is more relative to the natural language than traditional programming languages. Nonetheless, it is a formal language. Thus, the combination of the UML and the OCL is a formal way to express BR in IS development process.

Unfortunately, not much research is made on this topic. Erricson and Penker presented all possibilities to model a business with the UML – see Ref. [8] – and append a section with a description of expressing BR with the OCL. However, they discussed it considering few types of BR. Moreover, they did not educe advantages and disadvantages of the OCL as a language for expressing BR.

In Ref. [9], the main focus is set to the realization of BR of constraint type expressed with the OCL into database systems. However, the authors do not consider other types of BR though the OCL can be used as a query language as well as definition of derived values. More research is made in Ref. [10], where authors consider expressiveness of the OCL according to different types of BR. Apparently, the investigation has been made using the first version of the OCL. Therefore, many BR expressions with the OCL might be limited due to its first version of provided syntax.

The main aim of this research is to examine expressiveness power of the UML/OCL to model different types of BR. For the UML support possibility to represent systems in different abstraction levels, BR specified in the IS level model could be preserved until the implementation of specific level model. The existing tools provide opportunities for automated generation of those models. Therefore, BR specified in IS model, after elaboration and refinement, could be implemented in executive code.

For the examination of rules, widely adopted BR classification scheme, proposed by the GUIDE project – see Ref. [4], was selected. In this research we appeal to the extended list of action assertion BR types.

The paper is structured as follows. A brief overview of business rules is presented in Section 1. The objectives are discussed in Section 2. Section 3 deals with the tasks realization considering example business system is presented.

1 Business rules: overview

The interest in business rules has been shown for several decades. Many definitions of business rules concept have been presented as well as techniques to discover and express the rules, and a lot of classification schemes for the categorization of them have been proposed. Unfortunately, there is no industry standard definition for the term *business rules*.

Regarding Morgan, see Ref. [1], a business rule is a compact statement about some aspect of a business: it can be expressed in terms that can be directly related to the business, using simple, unambiguous language that is accessible to all interested parts i.e. a business owner, a business analyst, a technical architect and so on. In general, business rules describe how a company conducts its business.

The Business Rules Group (BRG) – see Ref. [4] – defined business rule in both business perspective and information system perspectives: from the business perspective, a business rule is guidance that there is an obligation concerning conduct, action, practice or procedure within a particular activity or sphere, and from the information system perspective, a business rule is a statement that defines or constrains some aspect of the business. Business rules may be captured by business experts, business owners or end users for keeping business works. While IT professionals, who are also in charge of BR capture, aim at making their applications usable in reality.

The BRG formalized an approach for identifying and articulating the rules which define the structure and control the operation of an enterprise. They had presented business rules classification scheme in the GUIDE project report – see Ref. [4]. According to the GUIDE, a statement of a business rule falls into one of the four categories.

1. Definitions of business terms – terms in glossaries or

entities, objects, classes depending on specification language. Typical examples could be presented by words: “Customer”, ”Party”, ”Employee” etc.

2. Facts relating terms to each other – natural language sentences or relationships, attributes and generalization structures in a graphical representation of the model. Typical examples could be presented by short sentences: “Customer address”, “Party identification number”, “Employee is absent” etc.
3. Constraints (Structural/Action Assertions) – constraints the structure or the behaviour of enterprise. Full length sentences express behaviour of constrains: “Customer may be one of the following status: gold, silver or bronze” etc.
4. Derivations – definitions of how knowledge in one form may be transformed into the other knowledge. For instance, sentence containing *know-how* elements: “For each item price ratio is calculated comparing with the previous month”.

Application of BR in IS development process differs into three domains: business system, information system, and software system (SS). The last one is closely related with implementation of BR in application and is not in the scope of this paper.

From the business system perspective, business rules should be described in a form relative to business people. The most understandable form is natural language. However, this form is ambiguous to be used for specification of rules from IS perspective. For this purpose, BRG has provided the specification of Semantics of Business Vocabulary and Business Rules (SVBR), see Ref. [11]. SVBR is oriented to business people and is designed to be used for business purposes independently of information systems designs.

From the IS perspective, BR must be defined in unambiguous, clear and precise manner to be implemented within SS later. For this purpose many different BR modelling methods and techniques are provided. The comparison of these techniques is presented in Ref. [12], [13]. Substantially, selection of one of these techniques depends on IS development approach, adopted by IS development team. Object-oriented approach is widely adopted and differs from others by its capability to capture the structure (data) and the behaviour (functions) of business system, while others are able to capture only one of them. Recently, the UML is the de-facto standard for object-oriented analysis and design. The OCL as adopted part of the UML, supplements this methodology with possibility to specify system models in more detailed and unambiguous way. Therefore, in the third section, investigation on modelling different types of BR with the UML/OCL will be provided.

2 Object Constraint Language: objectives

2.1 Main characteristics

Object Constraint Language (OCL) is a formal language allowing the specification of constraints in context of the UML model. The OCL has been developed as a business modelling language and has its roots in the Syntropy method, see Ref. [6]. It has been adjusted to the UML to compliment its modelling because it is not expressive enough to provide all the relevant details of a specification. However, the OCL is side-effects free language, therefore it cannot change anything in the model: the state of the system will never change because of the evaluation of an OCL expression, even though an OCL expression can be used to *specify* a state change (e.g., in a post-condition), see Ref. [7]. Furthermore, it is not possible to write program logic or flow control in the OCL as well as express implementation details.

The main characteristics of the OCL are presented as follows according to Ref. [7].

1. Both query and constraint language – it is possible to write a query expression of a body of an operation as well as define constraint to some attribute’s value or existence of some object.
2. Mathematical foundation – it is based on mathematical set theory and predicate logic and it has a formal mathematical semantics.
3. Strongly typed language – model elements used in the OCL expressions must conform with types, therefore the OCL expressions can be checked during modelling.
4. Declarative language – modeller can make decisions at a high level of abstraction without going into details how something should be calculated.
5. Object-Oriented analysis and design method.

The main purpose of the OCL could be formulated as follows according to Ref. [6]: i) specify invariants on classes, types or stereotypes in class model; ii) specify pre and post conditions of operations; iii) describe guards on transitions in state diagrams; iv) specify target for messages and actions; v) specify derivation rules for any expression over a UML model. Thus, considering the OCL characteristics and using the OCL 2.0 notation [7],[16],[17] different types of business rules will be discussed in the next sections.

2.2 Simplified document management system as an example.

Fig. 1 represents the class diagram of simplified document management system based on model requirements for electronic document and records management systems according to Ref. [14]. In presented system every document must have a type, a kind and some priority. Every document must be created by some author which may be a user or a group as well as some contact or a contact group. Document kind of receivable must be received by some address and kind of outgoing must be sending to some addressee. According to the processes with documents they may be in some state. Some documents may be involved in a file, which is a virtual catalogue for collecting documents. For system specification we used standard UML notation, which description and modelling guidelines could be found in Ref. [5].

The presented example is composed of different BR. For example document type may be one of the enumeration document types: a letter, an invoice, a memorandum, etc. To collect and check different BR classification scheme is required. To understand the nature of BR and the categories into which they fall classification scheme is presented in the GUIDE Business Rules Project report, see Ref. [4]. Though there are a lot of proposals how to describe and classify business rules in business rules research community, however GUIDE presents a formal approach for identifying and articulating the rules that define the structure and control the operation of an enterprise; moreover the classification proposed by this project involves others presented in Ref. [1],[3],[15]

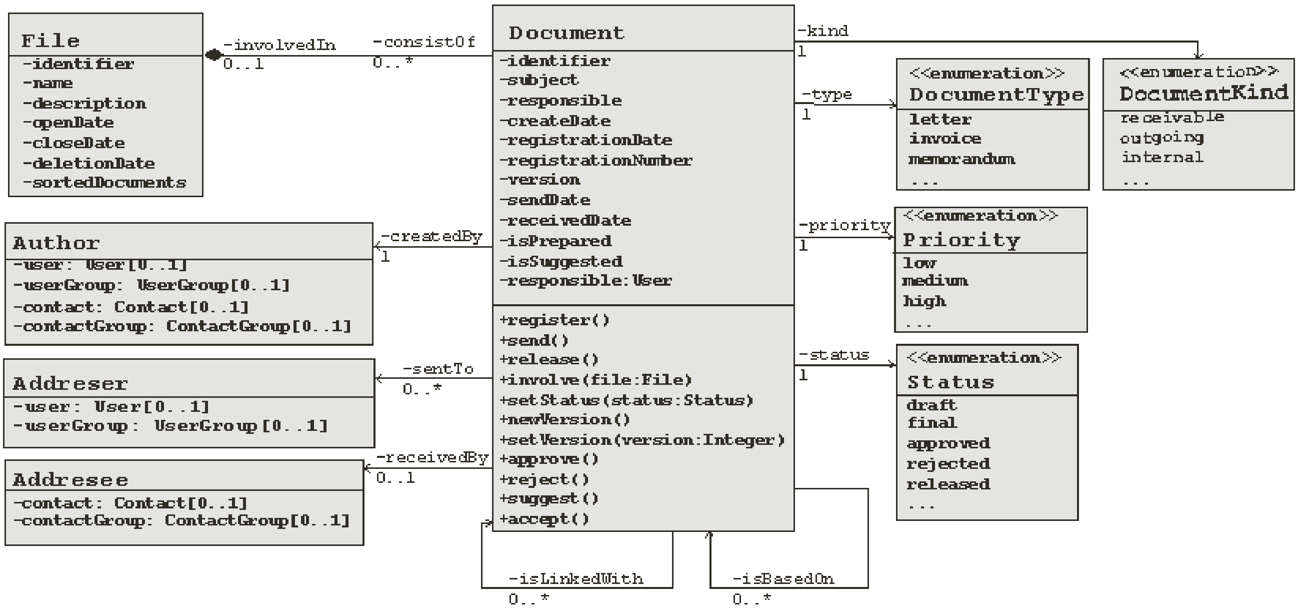


Fig. 1. Document management system: class diagram

3 Realisation of tasks using Object Constraint Language

In this section, considering the example discussed above, we will check extended list of action assertion type rules. This list involves the following types of BR: i) instance verifiers; ii) type verifiers; iii) sequence verifiers; iv) position verifiers; v) functional evaluators; vi) comparative evaluators; vii) calculators; viii) update controllers; and ix) timing controllers. Possibilities to model these types of rules using the UML/OCL will be discussed in the following subsections.

3.1 Instance verifiers

Each rule describes the effect of a correspondent (constraining object) upon an anchor (constrained object) [15]. Instance verifiers pertain to individual instances or occurrences of correspondent object classes [4]. Instance verifiers type includes the following subtypes: i) mandatory constraint (requiring occurrence of some object), ii) limited constraint (constraining number of population of object), iii) restricted (involving recursive structures), iv) pre-existing (requiring occurrence of some corresponded object class to exist before anchor object class and existence at the moment of rule check) and v) antecedent (requiring occurrence of correspondent object class to exist before anchor object class and it is not important or it still exists during check of rule).

Instance verifiers require possibility to manipulate a population of corresponded object class. The manipulation of collections of objects is very common in object-oriented systems. The OCL support different operations for collections which may be adopted by expressing this type of BR. All operations for collections are denoted in the OCL expressions using an arrow; the operation following the arrow is applied to the collection before the arrow [6].

Though most rules of this type could be modelled using standard UML notation, e.g. mandatory or limiting constraint as cardinality of association between classes or cardinality of object attribute, there is possibility to express them using the OCL. For example, the rule stating that *„Every document must have a responsible user assigned*" can be expressed as cardinality of attribute (or association) „responsible" or following the OCL expression as presented in the example 1.

```
// example 1
context: Document
inv: self. responsible->not Empty()
```

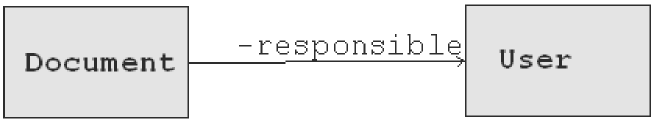


Fig. 2. Mandatory constraint rule: class diagram

The OCL expression is stated as invariant in the context of document. An invariant is a constraint that should be true for an object during its complete lifetime [7]. Thence this rule states while document exists it must have responsible user assigned to it. In the same manner it could be modelled limited constraint, e.g. *„A file must not have more that 1000 document involved in*" as presented in the example 2.

```
// example 2
context: File
inv:self.consists Of -> size()<1000
```

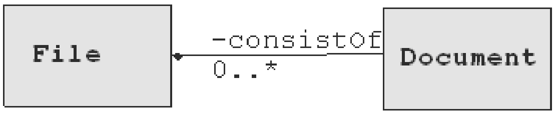


Fig. 3. Limited constraint rule: class diagram

The invariant above states that the size of the collection (number of all elements) of involved in file document objects must be less than or equal to 1000. Similar operation *count*() could be used to check the number of occurrences of some object in the related collection.

For restriction of recursive structures it should introduce recursive associations in class model. In the considering example recursive associations are modelled as document class attributes "isBasedOn" and "isLinkedWith". Suppose rule stating that "*A document cannot be based on itself*". In this case operation *reject*() could be involved as presented in the example 3.

```
// example 3
context:Document
inv:self.isBasedOn->reject(self)
```

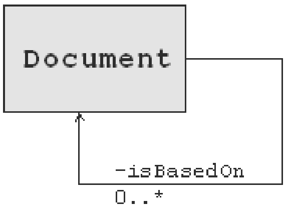


Fig. 4. Restricted constraint rule: class diagram

Operation *reject*() in this expression is used to state that in associated with document object collection „isBasedOn" cannot be document object itself. Similar operation *rejectAll*() could be used to except a collection of objects from related collection. On the contrary, operations *include*() and *includeAll*() could be used to preserve an object or object collection respectively to be in related collection.

Consider pre-existing constraint from involved example: „*Every outgoing document must be linked with some document*". The OCL expression for this example could be modelled in the following way as presented in the example 4.

```
// example 4
context Document
inv: self. kind=Document Kind::outgoing
implies self.isLinkedWith->notEmpty()
```

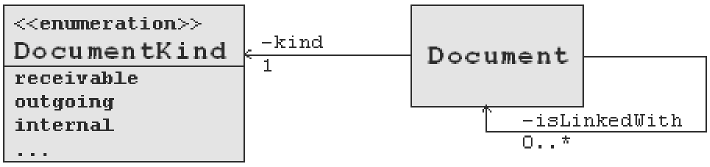


Fig. 5. Pre-existing constraint rule: class diagram

The expression above states that the fact that "*document is kind of outgoing*" implies the collection of objects "isLinkedWith" to be not empty. It follows thence that while the document is kind of "outgoing" it must be linked with the other document.

3.2 Type verifiers

Type verifiers control the creation of multiple instances in various object classes [4]. Type verifiers control occurrence of objects in object classes and may be one of four types: i) mutual (requiring that correspondent objects exist simultaneously), ii) mutually exclusive (requiring that no more than one correspondent object exist simultaneously), iii) mutually dependent (requiring that either one instance of every correspondent object class exists, or that no instances of any correspondent object class exist) and iv) mutually prohibited (requiring that at least one of the correspondent object classes has no instances).

For this type of rule comparison of collections should be involved. Combination of operations for collection with logical operator (OCL support *and*, *or*, *xor* operators) could be used to express most of this type of a rule. Consider following the rule "*Document can be created only by one of following: a user, a users group, a contact or a contacts group*". This rule could be referred to mutually exclusive type of type verifier. Depending on example model, the document is created by some author which can be a user, a user group, a contact or a contact group. The following OCL expression mutually excludes candidate subjects to author as presented in the example 5.

```
// example 5
context Document
inv:
 self.createdBy.user->notEmpty() and
(self.createdBy.userGroup -> isEmpty() and
 self.createdBy.contact -> isEmpty() and
 self.createdBy.contactGroup -> isEmpty() )
 or
 self.createdBy.userGroup -> notEmpty() and
(self.createdBy.user -> isEmpty() and
 self.createdBy.contact -> isEmpty() and
 self.createdBy.contactGroup -> isEmpty())
 or
 self.createdBy.contact -> notEmpty() and
(self.createdBy.user -> isEmpty() and
 self.createdBy.userGroup -> isEmpty() and
 self.createdBy.contactGroup -> isEmpty())
 or
 self.createdBy.contactGroup -> notEmpty() and
(self.createdBy.user -> isEmpty() and
 self.createdBy.userGroup -> isEmpty() and
 self.createdBy.contact -> isEmpty())
```

The invariant above states that one of the collections must not be empty while others must. Other types of type verifiers could be expressed in the same manner.. Consider mutually prohibiting rule: "Internal document cannot be received by or send to any subject(s)". Corresponding the OCL expression could be modelled as presented in the example 6.

```
// example 6
context Document
inv: self.kind = DocumentKind::internal implies
self.sentTo -> isEmpty() and self.receivedBy -> isEmpty()
```

The invariant above involves implication operator and comparison of the collections. It states that document of kind "internal" implies associated collections of objects *sentTo* and *receivedBy* to be empty. In the same manner it could be possible to express more of this type BR.

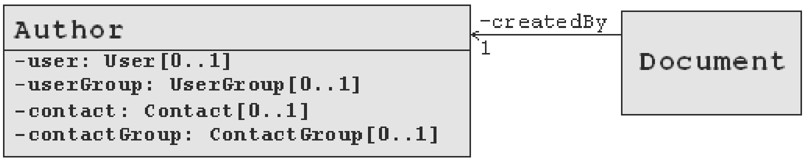


Fig. 6. Mutually exclusive constraint rule: class diagram

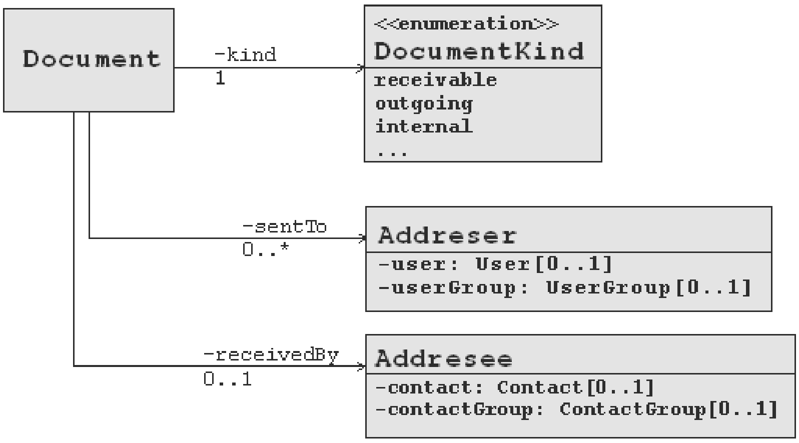


Fig. 7. Mutually prohibited constraint rule: class diagram

3.3 Sequence verifiers

According to Ref. [4], sequence verifiers control changes in object state. If object may be in multiple states then these rules determine the sequence in which instances of that object class may assume those states. Sequence verifier may be of one of the following types: i) initialling (requiring some state on object initialization); ii) forward (requiring transition to a higher state of object); iii) progressive and retrogressive (requiring transition to a next higher or lower state of object accordingly); iv) re-initializing (defining that when object moves to a lower state it should be moved to initial state first) and v) cyclical (defining that object can be moved to a lower state before it moves to highest state and vice versa).

This type of rules can not be fully defined in the OCL because the OCL is a declarative language, thus can not be used to define actions. The best way to express changes of object states in the UML is state machine diagrams [8]. The UML state machine diagrams describe the behaviour of a class over time of the states and transitions of a single object progressing through its lifetime. In the UML state machine diagrams the OCL expressions may be used in a number of ways – see Ref. [7], but commonly used are guards on states transitions and restrictions on states. The guard is condition on transition in a state machine diagram that must be met to change a state of object. Usually, restrictions on states are restrictions on values of links and attributes when an object is in a certain state. Using the UML state machines to define a sequence of object states and the OCL to express restrictions on model elements is possible to express most of the rules of the sequence verifiers type.

Referring to considering example, document lifecycle could be explained as state machine diagram as presented in Fig. 8.

The process could be described as follows. A new document saved in system is being prepared for release. After acceptance it may be suggested for release. After suggestion for release document may be approved or rejected. If a document is approved then it could be released, otherwise if a document is rejected the new version of document must be created referring to the cause of rejection. As sequence verifiers require, the sequence of object states is defined (by state machine diagram) in a model. The mentioned-above guard for transition is described as transition from the state *draft* to the state *final* which is an entry point to release process by the UML notation meaning. The restriction is enclosed in brackets and denotes that transition is possible only if document is prepared (by checking Boolean type attribute *isPrepared* value). Restriction on a state is described as *releaseCandidate* state denoting that document can be in this state only if it is suggested (by checking Boolean type attribute *isSuggested* value).

Considering the sequence verifier subtypes, the control of object states could be preserved by operations on any the OCL instance *oclInState*() or *oclInState* (str:StateName). Supposing the following rule: "*Every newly created internal document is draft until it is accepted that document is prepared for release*". Firstly, the rule denotes that initial state where document must be in is a draft. Corresponding the OCL expression is presented in the example 7.

```
// example 7
context Document::accept()
pre: self.oclInState(draft)
post: self.isPrepared=true and self.oclInState(final)
```

The OCL expression above restricts an operation *accept*() denoting that before execution of the operation document must be in a draft state and then it must be prepared in a final state. Thus control of changes of object states using the OCL could be applied to the other sequence verifiers.

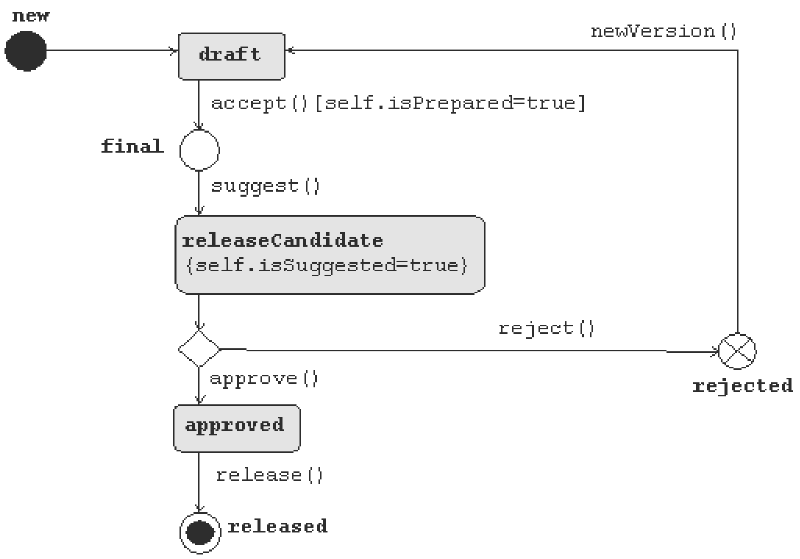


Fig. 8. States of document: state machines diagram

3.4 Position selectors

Position selectors pertain to a value, either in a value sequence or in a chronological sequence [4]. This type of rules is of two types: i) positioned – the lowest and the highest, and ii) chronological – the oldest and the newest. Supposing the rule: "*Document version must increase by one on set of version*". Respective OCL expression could be modelled as presented in the example 8.

// example 8

context Document::setVersion(version: Integer)

pre: --none

post: version=@pre + 1

The expression above denotes that after execution of operation *setVersion*() document attribute version must increase by one. It is expressed using keyword @pre which refers to an attribute value at the start of operation.

Another way of expressing position selectors could be usage of the OCL loop operations. Different iterator (loop operation) types may be chosen depending on business rule nature. For example, the rule mentioned above could be expressed using iterator *forAll*() as presented in the example 9.

// example 9

context Document

inv: self.allInstances->forAll(doc1,doc2 | doc1 <> doc2 implies doc1.version = doc2.version + 1)

Iterator *forAll()* returns true if all elements of collection used expression are true. In this case the operator *forAll()* is used to denote, that every different element of collection attribute's version value should be greater by one. Thus expression of positioned type of position selectors could be applied to the other business rules. However, the chronological sequence rules require additional constructs of model because the OCL does not support date or time data types. Therefore there should be involved some utility elements to support chronological sequence of elements of the collection.

3.5 Functional evaluators

Functional evaluators take care of sequence in which instances of object class are defined [4]. There are the following types of functional evaluators: i) unique (requiring unique sequence of values), ii) ascending and descending (requiring sorting of values in sequence), iii) non-renewable (requiring that any given value of the correspondent object, if used more than once, may be used only in strictly successive instances of the anchor object), iv) patterned (requiring that successive instances of the anchor object class be assigned in a specified sequence, or tests for that condition).

As it was discussed in the previous section control or modification on sequence of elements could be preserved using iterators. For example, unique values evaluator could be the following rule: "*Document registration number must be unique*". Respective OCL expressions are presented in example 10.

// example 10

context Document

inv: self.allInstances -> isUnique
(doc | doc.registrationNumber)

context Document

inv: self.allInstances -> forAll
(doc1, doc2 | doc1 <> doc2 implies
doc1.registrationNumber <> doc1.registrationNumber)

The expressions above are identical because the first one uses operator *isUnique*() which is true if every document has a unique registration number and the second one is true if every pair of different document objects, registration numbers is not the same.

Sorting out the collection elements could be modelled using *sortedBy*() which simply sorts out collection's element in ascending mode. For example, the rule „*Document registration number must be in ascending order within a file*" could be expressed by defining attribute sorted Documents of collection type in context of file (example 11).

```
// example 11
context File
def: sortedDocuments: Sequence(Document) = self.
consistOf->sortedBy( registrationNumber )
```

The expression above collects sorted elements of documents collection into sequence. The sequence as well as other type of the OCL collections defines a sequence and ordering the elements of collection. Therefore, if it is required that collection elements are served out in some pattern operations *asSequence*(), *asBag*(), *asSet*(), *asOrderedSet*() could be applied, as it is described in Ref. [7]. Combining mentioned operators with iterators for collections could produce sequences of elements arranged in predefined way, thus, providing a possibility to express complex rules.

3.6 Comparative evaluators and calculators

Comparative evaluators describe comparisons between pairs of instances of object classes. The comparisons may be 'equal to', 'not-equal-to', 'greater-than', 'greater-than-or-equal-to', 'less-than', or 'less-than-or-equal-to', and so forth [4]. The OCL supports different logical operators for this purpose and some of them were discussed above. It should be noticed that the OCL is strongly typed language, therefore compared objects or values should be carefully chosen.

Calculators involve any standard computation, e.g. sum, subtract, max, min, med, etc. Calculations in the OCL are possible between two data types: integer and real. Calculators may be expressed using standard OCL calculating operations (summary, subtraction, division, modulus, etc.) but as well as in the case of comparative evaluators conformance of types is required.

3.7 Update controllers

Update controllers prescribe whether updates to a database may occur and may be of the following types: i) frozen (requiring existence of anchor object to make some operation to correspondent object); ii) frozen to users (the same as above, but restricted to specific list of users); iii) enabled (existence of every instance of anchor object enables operations on correspondent object), and iv) enabled with reversal (when anchor object is deleted the state of correspondent object is reversed).

The OCL is the constraint language therefore every expression in some way may confine an operation to a database. In object-oriented systems object attributes values are set or got by some defined operation. Therefore, if it is required to restrict some changes of value then pre-and-post conditions for operations could by applied. For example, restriction on change of value may be used for post condition for an operation denoting that value must be the same as at start of operation by introducing *@pre* keyword as it was shown in the previous examples. Certainly, if it is required to relate constraints on operations with some users, e.g. restrict execution of operation or change of value to specific users, additional constraint should be introduced into the model to check whether operation is executed by the related user or not.

Consider the following rule "*Only assistant can register internal documents*" requiring an assistant role to execute the operation *register*(). One of possible expressions of this rule may be introduction of a new operation into document class. The operation, e.g. *checkRole*(event: String, role: Role) could be of Boolean return type checking whether some event is made up by some role. Additionally, it should be mentioned that in Ref. [18] much attention is made on modelling access control with the UML/OCL, where suggestions on modelling such type of rules are proposed.

3.8 Timing controllers

Timing controllers prescribe tests for the length of time that instances of correspondent objects have (or have not) existed [4]. The OCL does not support the possibility to express time-based constraints which could confine objects states according to time. Therefore, additional elements (such as time utility) could be introduced to model or extension to the OCL could be provided as suggested in Ref. [19],[20].

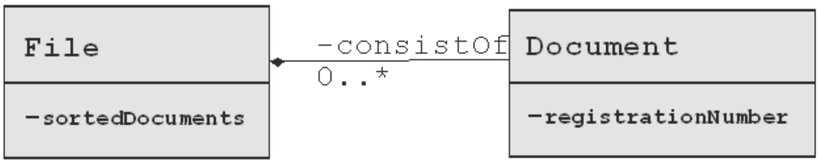


Fig. 9. Functional evaluator rule: class diagram

Conclusions

In this paper, research on possibilities to express different types of action assertion business rules with the UML/ OCL was made. It was established that the combination of the UML and the OCL is expressible enough to capture most of action assertion type rules, both structural and behavioural.

The OCL supports operations on collections of the UML model elements, therefore restrictions on population of object instances could be expressed as it is required by instance verifiers.

The OCL provides calculative and logical operators, which may be used to express functional evaluators. The combination of operators and operations for collections may be used to express type verifiers.

The OCL has pre and post conditions on operations of classes, therefore restrictions on data manipulation could be applied in many different ways.

The OCL provides possibility to query the UML model; therefore, the OCL may be used to express derived values. Supported collection data types enable to predefine required structure of the collection; in addition, iterators enable to make a more complex structure of the collection.

The OCL supports only limited set of data types, therefore there is no possibility to express BR requiring comparison of variables with constants (e.g. it is not possible to compare dates).

The OCL does not provide a possibility to express time-based constraints which could constraint objects states according to time, therefore suspending model with additional constructs is required, or extension for the OCL should be provided.

Acknowledgements

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

References

[1] Morgan T. Business Rules and Information Systems: Aligning IT with Business Goals. Addison Wesley. 2002, pp. 384

[2] Hay D.C., Requirements Analysis: From Business Views to Architecture. Prentice Hall, 2002

[3] Von Halle B. Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley and Sons. 2002, pp. 495

[4] Hay D.C. et al. Defining Business Rules - What Are They Really?//GUIDE project. 2001, available from: [http://www.businessrulesgroup.org/first_paper/br01c0.html], retrieved 2009 01 10

[5] Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, Second Edition. Addison Wesley. 2005.

[6] OMG. OCL 2.0 specification. 2005. Available from: [http://www.omg.org/docs/ptc/05-06-06.pdf], retrieved 2009 01 15

[7] Warmer J., Kleppe A. Object Constraint Language, The: Getting Your Models Ready for MDA, Second Edition. Addison Wesley. 2003, pp. 347

[8] Eriksson H.E., Penker M., Business Modelling with UML: Business Patents at Work. John Wiley and Sons. 2000.

[9] Zimbrao G. et al. Enforcement of Business Rules in Relational Databases Using Constraints//UMFG Database Group. 2007, available from: [http://www.lbd.dcc.ufmg.br:8080/sbbd2003/artigos/paper010.pdf], retrieved 2008 12 20

[10] Sosunovas S., Vasilecas O. Verslo taisyklių modeliavimas OCL kalba (in Lithuanian) // *Lietuvos matematikos rinkinys.* Vilnius. 2004, pp. 369-376

[11] OMG. Semantics of Business Vocabulary and Business Rules, v 1.0. 2008, available from: [http://www.omg.org/docs/formal/08-01-02.pdf], retrieved 2008 10 15

[12] Lebedys E., Vasilecas O. Verslo taisyklių modeliavimo kalbų analizė (in Lithuanian) // "Informacinės Technologijos'2004". Technologija. 2004, pp. 487-494

[13] Herbst H. et al. The Specification of Business Rules: A Comparison of Selected Methodologies. Methods and Associated Tools for the Information System Life Cycle, Amsterdam et al.: Elsevier 1994, pp. 29-46.

[14] MoReq2, Model Requirements Specification for the Management of Electronic Records, available from: [http://www.moreq2.eu/], retrieved 2008 11 21

[15] Ross G.R., Principles of the Business Rule Approach. Addison Wesley. 2003, pp. 352

[16] Warmer J., Kleppe, A., Cook, S., Informal Formality? The Object Constraint Language and its application in the metamodel. in Proc. International Conference on the Unified Modelling Language (UML), Springer-Verlag. 1999.

[17] Warmer J., Kleppe, A., OCL : The Constraint Language of the UML. Journal of Object-Oriented Programming, 12(1):10–13. 1999.

[18] Ahn G.J., Shin M.E., Role-Based authorization constraints specification using Object Constraint Language// Enabling Technologies: Infrastructure for Collaborative Enterprises, Proceedings tenth IEEE international Workshops. 2001, pp. 157-162

[19] Flake S. Temporal OCL extensions for specification of real-time constraints. In Workshop Specification and Validation of (SVERTS'03). 2003.

[20] Hamie A., Mitchel R., Howse J. Time-Based Constraints in the Object Constraint Language. School of Computing and Mathematical Sciences, University of Brighton. 2006, available from : [http://www.biro.brighton.ac.uk/biro], retrieved 2008 12 14

AN OPEN ISSUES IN BUSINESS RULES – BASED INFORMATION SYSTEM DEVELOPMENT

Ruta Dubauskaitė*, Olegas Vasilecas
Department of Information Systems, Vilnius Gediminas Technical University,
Sauletekio al. 11, LT-10223, Vilnius, Lithuania

Received Januar 10, 2009, accepted August 21, 2009

ABSTRACT. All rules are not usually elicited from business system, sometimes they are damaged during transformation from business level to program system or implementation level in information system development process. Hence, in this paper we propose an idea how to improve the process of creating the correct set of business rules. It is suggested to use ontology to avoid incorrectness. Besides, researches state that using templates of business rules can enable us to define rules in more precise way. It is proposed to rely on the integrated framework of business rules elicitation, which came after detailed analysis of business rules and their elicitation process. The main activities in business rules elicitation process on business, information and program systems levels are defined in the framework, and the actual open issues are formulated. Provided suggestions shows trends in evolution of tools that facilitate activities concerned to using the rules.

Keywords. Business rules, ontology, templates of business rules, transformation of business rules.

Short title. Business Rules Based ISD.

**Corresponding author, email: rutad@isl.vgtu.lt, phone: +370 5 2744860*

Introduction

The environment of enterprises changes constantly and changes appear in the business. Business logic (it is often expressed by rules) also changes. On-line response to changes is one of the factors of enterprise competitive advantages. Information system is important for performing enterprise activities successfully, because relevant information is saved in it, which is necessary for performing business process, it supports decisions that are important for the activity. Business is performed according to rules. They make an important and integral part of each information system by expressing constraints on concepts, their interpretation, and/or relationships of a domain. Therefore, it is relevant to pay some special attention to business rules in developing the information systems.

All rules are not usually elicited from business system. Sometimes they are damaged during transformation from business level to program system or implementation level in information system development process. For example, if rules are not defined clearly they can be interpreted differently. Besides, sometimes representatives of business, analysts and designers use different concepts for defining the same objects. Hence, in this paper the solutions 'how to avoid creating incorrect set of rules' are offered.

This work covers the areas of how to improve the process of business rules elicitation from defining business rules in business system to their implementation in the software. Section 1 represents an overview of related works, business rules and their elicitation process is investigated. Section 2 concentrates on the suggested framework of business rules elicitation. Section 3 provides open issues in business rules information system development.

1. Overview of Related Works

1.1 The Analysis of Business Rules

Information systems are developed to support activities that are performed by an enterprise. Information system is a part of business system. Software systems are developed to support information system. Processes of information system are performed according to some rules. Many authors have emphasized importance of business rules [1], [2], their usage in developing the information system [3], [4] modelling [5] [6], [7] and realization [8], [9]. Therefore, it is relevant to pay special attention to business rules in developing the information systems.

Business rule is a statement that constrains or defines some aspects of particular business at the business system level [10]. Rules are represented declaratively (informally) at business system level [9–12]. The following sentence is presented as a typical example: a customer can buy more than credit limit permits. Business rules can be defined differently depending on context.

Business rule from the perspective of information system is a statement that defines how data should be processed and represented using some rule-based language [12], [13]. Rules of information processing are represented explicitly and can be interpreted in one way. These rules could be labelled as formal rules [9]. The example of this rule according to Hay [13] is represented in the following sentence: 'Total Value' of an ORDER can not be greater than the 'Credit Limit' of a CUSTOMER.

Business rule from the perspective of software system is a statement transformed to the executable rule – like trigger, constraint or fragment of application.

All three enterprise abstraction levels (business, information and software system) should be correctly integrated. The lower level system is constrained by the higher-level system. All changes in the higher level should be reflected at the lower level system correctly. The activities of enterprise should be modelled gradually from business system level to software system level. The life cycle of business rule is presented by Bajec and Krisper [14]. The consistent transformation of business rules enables common conceptualization of all systems. Hence, the following definition of business rules elicitation process is used here. Business rules elicitation process is a set of integrated activities that are performed to define business rules at business system level and to transform them to executable rules gradually.

1.2 The Definition, modelling and implementation of business rules

The first step of rules elicitation could be named as a rule definition. According to Ref. [1], rules definition process is usually complicated because i) often business rules are not expressed in explicit way, ii) there are different rules sources, for example, vocal language, laws, documents, domain ontology and etc. Business rules can be elicited in different ways, the main of them are document analysis and dialogue with enterprise workers. Besides, sometimes the representatives of business, analysts and designers use different concepts for defining the same objects, because the chance is that incorrectness appears during the rules elicitation. The researches state that every term, used in business should be defined clearly, because very often the same term may have several meanings [1], [15]. The vocabulary of terms as well as ontology helps to solve the problem of creating incorrect set of rules. The main disadvantage of this approach is that it allows reducing negative impact of only one factor (incorrectness because of different terms) from all factors that determine incorrectness in a set of rules.

The second step of rules elicitation could be named as rule modelling. Defined rules can be represented in different ways, for example, sentences of natural language, decision tables, diagrams, sentence of formal language etc. [9],[16]. As usually, three levels of the different representation are defined according to Ref. [17].

1. *Informal rules*. The rules are represented by semantics and syntax of natural language.
2. *Technical rules*. The UML (Unified Modelling Language) [18], the ORM (Object Role Modelling) [19] could by entitled as the typical examples of graphic modelling languages. The rules are represented by logical constructions of graphic modelling languages including: i) templates of business rules; ii) decision trees; iii) decision tables etc.
3. *Formal rules*. The rules are represented by logical constructions of the formal languages. The OCL (Object Constraint Language)) [20] represents a typical example from group of formal languages. The main behaviour of the formal language is the automatic processing.

Rules represented by natural language are easily understandable by people, but they can be interpreted variously. Von Halle et al [3], [9] suggest using predefined template for rules inputting. But some other authors, in particular, von Halle et al (Ref. [4], [9]) do not provide information, how to transform these rules into executable rules, for example, SQL triggers. Only formal rules can be transformed to executable code [16]. Hence, it is an open issue, what language should be selected in order to create more accurate set of rules. All above-mentioned languages are not standard for rules modelling. During the analysis a clearly defined methodology of rules elicitation from business system level to software system level has not been found.

Not all modelled rules are realized in software system. A part of them is presented in the text document. Another part of rules can be implemented in program code, database, using technical software [1], using business rules management systems [21].

1.3 The Results of Analysis

The analysis of related works has shown that the problem of creation incorrect set of rules is arising due to several circumstances: i) sometimes representatives of business, analysts and designers use different concepts for defining the same objects; ii) non-valid interpretation because the rules that are represented by natural language can be interpreted variously.

Ross et all [1], Appleton et al [14] suggest using common vocabulary or ontology for business representatives and creators of information system. The main disadvantage of this approach is that it allows reducing negative impact only of one factor (incorrectness because of different terms) from all factors that determine incorrectness in a set of rules.

Von Halle et al [3], [9] suggests using predefined template for rules inputting, which would allow defining rules in one way (one possible interpretation). But mentioned authors do not provide method for realization rules, inputted using templates, it is not clear how to transform them to executable rules, for example SQL triggers. Hence, the researches provide various ways for solving problem of creating incorrect set of rules, but these approaches are not completed and integrated.

The framework of business rules elicitation, which have came after detailed analysis of business rules and their elicitation process, is proposed in Ref [3]. The suggested framework integrates approaches of different authors. The main activities in business rules elicitation process on business, information and program systems levels are defined in the framework, and the actual open issues are formulated.

2. The framework of business rules elicitation

Analysis of business rules and their elicitation process shows that rules elicitation process is defined differently depending on different points of view. For example, to an analyst of information system rules elicitation means defining rules from conversations with business representatives, documents and ontology. These rules are captured by natural language. Hence from the analyst perspective the rules elicitation process is rules definition activity, which input is source of rules (vocal language, documents, ontology) and result is a set of rules (statements of natural languages). Therefore the three aspects (input, activity and result) describe a point of view of information system analyst in detail.

An analyst, an architect, and a designer can analyze rules elicitation process differently. Table 1 describes business rules elicitation process from different points of view and aspects. Hence, an information system analyst starts the process of rules elicitation from rules definition. He/she defines business rules by analyzing the documents of enterprise, domain ontology, talking with enterprise employers. Using pre-defined rules templates analyst input defined rules to computerized system (see Table 2). During this process the specification of business rules is created, rules are represented by partially formal language, for example, structured English language.

More information about the classification of rules and rules classes templates are provided in von Hale's work [3]. An architect transforms partially formal rules to formal rules that can be automatically transformed to executable code.

Table 1. The framework of business rules elicitation

View point	Aspects			
	Input sources for definition of rules the rule	Definition of rules	Result of definition	Typical sample of language for representing
Analyst	Sentences of vocal language, documents (laws, orders), domain ontology, etc.	Semantic and syntax of vocal language, templates of documents, etc.	Constructions of natural language	English, German, French, Lithuanian
	Sentences of natural systemized language	using predefined template for rules inputting	Specification of rules - partially formal language	Structured English language
Architect	Specification of rules	Modelling of rules	Graphic model or formal language	UML diagram, OCL
Designer	Model of rules	Transformation of rules	Implemented rules, executable code	SQL, JAVA, C++

Table 2. Templates of rules

No.	Template	Example
1.	**<term 1>** IS A **<term 2>**	**Bus** IS A **vehicle**
2.	**<term 1>** MUST NOT BE IN LIST **<a, b, c>)**	**The status of items on offer** MUST NOT BE IN LIST **<sold, not brought>**)
3.	**<term>** IS COMPUTED AS **<formula>**	**Total sum** IS COMPUDED AS **sum1 + sum2 + sum3**

Rules elicitation process is represented graphically in the Fig. 1. Successive performing of all activities allows implementing more correct set of rules. However, defined activities are rather recommended than mandatory. Before choosing one or another the approach of business rules system development every enterprise should take into account various factors, for example, critical importance of developed system (if system is critically important), objectives, time and budget resources, qualification of employers, tools end etc.

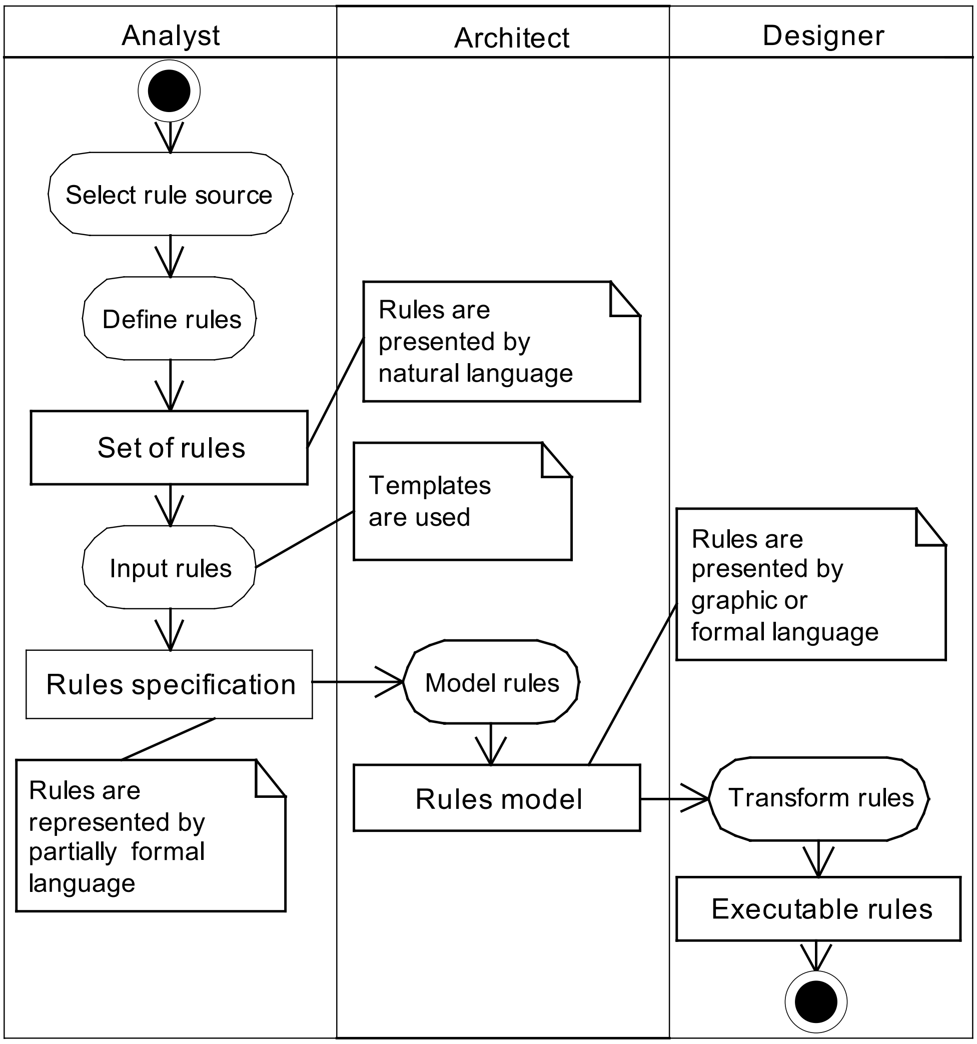


Fig. 1. Process of rules elicitation

3. Open issues in business rules – based information system development

The suggested framework, the aspects and views of business rules elicitation process help to define open issues in business rules - based on system development they are as follows.

1. Work with domain ontology, for example, automated generation initial set of rules from selected domain ontology.
2. Work with templates. Template should enable to input various types of rules.
3. Extension of the selected modelling language with special symbols for business rules modelling.
4. Automatic transformation rules of information system to executable rules (rules of software system).
5. The main issue is to integrate and improve approaches, methods that are suggested by various researches on above-mentioned issues.

CASE (Computer-Aided Software Engineering) tool, which facilitates all rule elicitation activities (generation rules form ontology, rules inputting through templates, automated generation of executable rules etc.) during the analysis of MagicDraw [22], Power Designer [23] and others tools have not been found. Hence, solutions of defined issues and their realization in CASE tool enable to create rule-based systems easier and more qualitative.

Conclusion

The analysis of related works on business rules elicitation has shown that there is no standard, which would define activities of business rules elicitation process. It can be understood differently depending on the point of view. Authors propose to integrate these views into framework by aspects. Suggested framework provides requirements which while implemented would help to create more correct set of rules. For example, information system analyst should use predefined templates for inputting rules, it allows to interpret inputted rules in one way.

Open issues in business rules – based on information system development have been formulated after investigating the suggested framework of rules elicitation. The main issue is to integrate and improve approaches, methods that are suggested by various researches. Solutions of defined issues and their realization in CASE tool enable to create rule-based systems easier and more qualitative.

Acknowledgements

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

References

[1] Ross R. G. Principles of the Business Rule Approach. Addison Wesley, 2003.
[2] Ross R. G. Business Rule Concepts. Business Rule Solutions Inc., 1998.
[3] von Halle B. Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons, 2002.
[4] Morgan T. Business Rules and Information Systems: Aligning IT with Business Goals. Addison-Wesley, 2002.
[5] Herbst H. Business Rule-Oriented Conceptual Modelling. Physica-Verlag, 1997.
[6] OMG. Semantics of Business Vocabulary and Business Rules (SBVR), 2006, Available at [http://www.omg.org/docs/dtc/06-03-02.pdf], retrieved 2009 03 19.
[7] Gurarino N. Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy, IOS Press,1998, 3-15.
[8] Date C. J. An Introduction to Database Systems. Addison Wesley Longman, 2000.
[9] Chisholm M. How to Build a Business Rules Engine. Morgan Kaufman Publishers, 2004.
[10] Zachman J. A., Sowa J. F. Extending and formalizing the framework for information systems architecture. IBM Systems Journal, IBM Publication, vol. 31(3), 1992, 590-616.
[11] Perrin O., Godart C. An approach to implement contracts as trusted intermediaries. In: Shan, M. C. (ed.): Proc. of the First International Workshop on Electronic Contracting (WEC'04), IEEE Xplore, 2004, 71-78.
[12] Lebedys E., Vasilecas O. Analysis of business rules modelling languages. In Proc. of the IT'2004, Lithuania, Technologija, 2004, 487-494.
[13] Hay D. C. Requirements Analysis: From Business Views to Architecture. Prentice Hall PTR, 2003.
[14] Bajec M., Krisper M. Issues and challenges in business rule-based information systems development. In Bartmann D. , Rajola F., Kallinkos J. (eds.). Proc. of Information systems in a rapidly changing economy, Regensburg: Institute for Management of Information Systems, 2005, 1-12
[15] Appleton D. S. Business Rules: The Missing Link. Datamation 30, 1984, 145-150. Available at [http://www.dacom.com/News/publications/busrules.asp], retrieved 2008 08 08 .
[16] Booch G., Rumbaugh J., Jacobson I. The Unified Modelling Language User Guide. Addison-Wesley, 2000.
[17] Vasilecas O., Saulis A. Informacinių sistemų kūrimo metodai ir technikos (The Methods and Techniques of Information System Development). Paskaitų medžiaga, Lithuania, 2007, 110 - 183.
[18] OMG. OMG Unified Modeling Language (OMG UML), Infrastructure, v2.1.2 , OMG Document: formal/2007-11-04, 2007. Available at [http://www.omg.org/cgi-bin/doc?formal/07-11-03], retrieved 2008 09 11.
[19] Halpin T. Object-Role Modeling (ORM/NIAM). In Handbook on Architectures of Information Systems. Springer-Verlag, 1998, 81–102.
[20] OMG. UML 2.0 OCL Specification, 2003. Available at [http://www.omg.org/docs/ptc/03-10-14.pdf], retrieved 2008 09 20.
[21] Vasilecas, O., Avdejenkov, V. Business Rules Applying to Credit Management. In K. Elleithy (ed), Advances and Innovations in Systems, Computing Sciences and Software Engineering, Springer Netherlands, 2007, 481-483.
[22] No Magic. Magic Draw tool. Available at [http://www.magicdraw.com/], retrieved 2009 03 24.
[23] Sybase. Power Designer tool. Available at: [http://www.sybase.com/products/modelingdevelopment/powerdesigner], retrieved 2009 03 24.

VILNIUS BUSINESS COLLEGE

EXPERIENCES IN TEACHING INTERNATIONAL BUSINESS WITH BUSINESS SIMULATION GAME

Leena Laari-Muinonen*, Kirsi Viskari
International Business Programme, Faculty of Business,
Saimaa University of Applied Sciences,
Pohjolankatu 23, FI-53100 Lappeenranta, Finland

Received Februar 28, 2009 , accepted August 21, 2009

Abstract: The purpose of this article is to introduce the experiences gained in teaching a business simulation game in a web-based learning environment via a multi-cultural pilot course. In the pilot course the students were instructed to actively gather their comments and thoughts and formulate a report discussing their learning experiences. After the pilot course the teachers' experiences of e-learning were collected, and the student feedback was analyzed. The business simulation game proved to be a good learning model for reaching a comprehensive understanding of strategic business decision-making. The pilot course also demonstrated that the basic elements of e-learning like flexibility, sense of community and usage of the modern technology are important. In the multi-cultural groups the formation of a common working culture is important for gaining the benefits of e-learning.

Keywords: business teaching, simulation game, e-learning

Short name: Teaching business with a simulation game

** Corresponding author, email: leena.laari-muinonen@saimia.fi*

Introduction

Business studies consist of a variety of subjects, all touching one corner of the business functions; accounting, marketing etc. In order to give the students a comprehensive picture of how these areas influence one another as well as the total company or organization, simulation games have been found as a useful method. In order to gain the best learning experience of a simulation game, the students should already possess understanding of each of the business areas at least to some extent.

The web-based technologies of today offer a wide range of alternative tools and learning environment choices to support teaching and learning. Simulation games utilize information technology in order to give a realistic picture of the operational environment, and in combination with web based architecture they virtualise the learning environment even more. This article discusses the usage of a web based business simulation game as the learning environment for an international group of business administration students. The theoretical part includes perspectives of web based learning environments and the challenges of virtual teams. It highlights various perspectives of virtual learning environment, such as the technology, ability to reach the students, work loads, flexibility, tasks, feedback, supervising, and sense of community and learning results. In the empirical part we discuss the experiences gained from a pilot course and also the prospects for the future enhancements.

1. Teaching framework

1.1. Technology

Learning and teaching environment possibilities have developed dramatically with the introduction and rapid expansion of internet based tools and systems. The student base has changed; there are more and more students who return to school to enhance their knowledge, and students who cannot attend the traditional classroom-related learning environment. Fig. 1 represents the perspectives of virtual learning environments [1] in condensed form. Their demand for the education channels provides not only the basic need for virtual learning but it is also an increasing educational channel for the traditional students. [2]

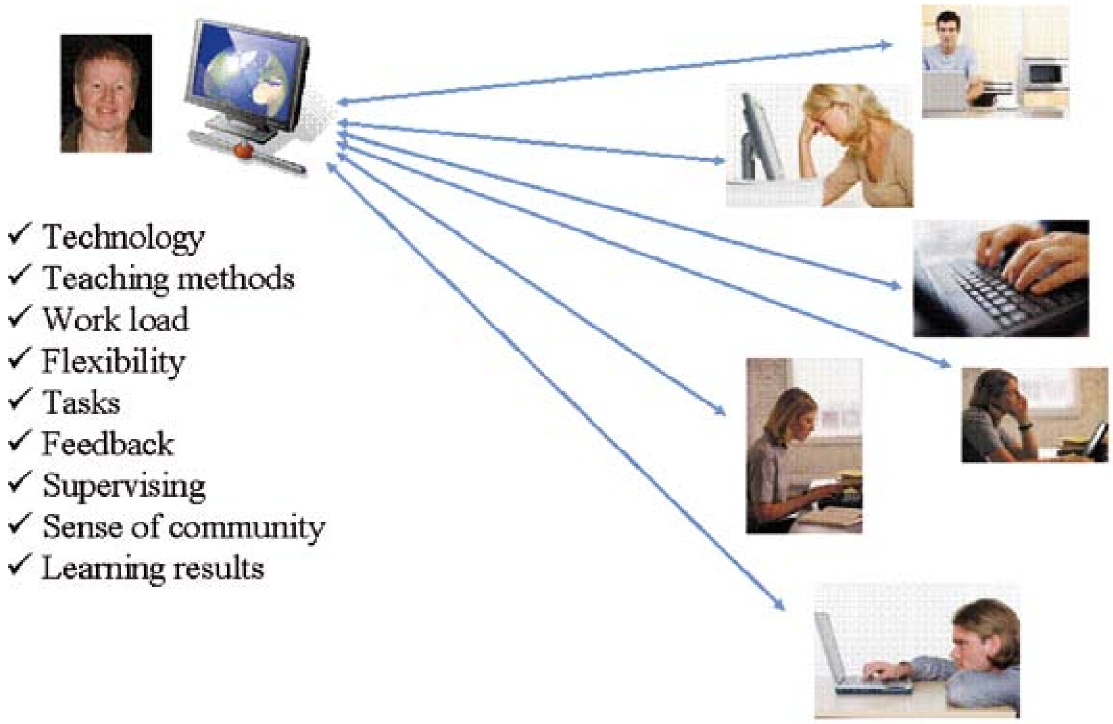


Fig. 1. Perspectives of virtual learning environments. Adapted from Ref. [1]

Also other competencies should be provided to ensure that the members get over the first threshold of operating in a virtual environment. These competencies include the knowledge of the needed technological tools and systems, but not to be forgotten that the technology is not enough to ensure good learning experiences in the virtual teams. Educational organizations should remember not to over-emphasize the technology, but remember the methodological approaches required to fully utilize the technological tools. [3]

The wide range of tools that can be utilized in the web based learning environment offers not only alternatives but also become challenges for the teachers. In order to select the most suitable tools the teacher needs to be familiar with them and to know their advantages and disadvantages. Wrong tool choice can be harmful to the learning. The teacher also needs to make sure that the students possess the usage knowledge of the selected tools. [4]

1.2. Teaching methods

Virtual learning environments offer new dimensions for studying in groups and societies that are not physically in the same location. Teachers of today can no longer rely on their traditional one-way teaching methods, as the world of learning and teaching has changed to be an interactive society. Delivering the learning message is not enough for the students, and the concept of learning has fast evolved to be a collaborative, two-way process, integrating the students more into the learning situations. This constructive theory implies that the learning takes place in the interaction between the students, the teacher and whole environment, and is claimed to provide a deeper level of learning than the traditional one-way learning message transfers. [5] [2]

Inclusion of modern technological tools and virtual, web based systems and learning environments creates challenges for the teachers, and the teaching environment is becoming more and more complex. Also the requirements of the students representing the younger generation are higher in regards of the alternatives, freedom of choice of place and time as well as the level of convenience of studying. In addition to specifying the learning objectives from an academic perspective, the teachers of today need to include the perspective of team-based, collaborative learning in their courses. [3] [5] [2]

1.3. Work load

The virtual learning environment is often perceived as free, easy and entertaining. Especially those students who are familiar with the most current web technologies, usually tend to have a shorter concentration span and they prefer visuals over the written information. Also the ability to choose the time and place to study may sound misleadingly easy, when instead it requires more self-discipline and proactive input than the passive information-pushing mode. Collective knowledge building is based on the activity of the students and study teams, which may cause higher work load than traditional individual learning with ready-made materials. [4]

Work load nature differs in the virtual learning from traditional setting. Instead of studying the given material, the students need to search for the materials and information, generate input based on their findings and collectively build learning materials. Problem-based and resource-based methodologies have paved the way, but did not include the aspect of student-built content. [4]

The work load of the teacher has the same challenges as that of the students. It is a false belief that a virtual course demands less working hours than a traditionally organized course with lectures. Careful preparation in designing the course structure, activities, timelines, feedback, tasks etc. may be very demanding, especially if the teacher is not familiar with virtual teaching. Actually the course design is far more important than the selection of tools and technologies. The approach should be student-oriented instead of teacher-oriented, which turns the traditional setting upside down. Self-discipline is required also from the teacher, and the students demand constant virtual presence of the teacher in different interactive forms. [4]

1.4. Flexibility

Web based education environments promote anytime, anywhere learning possibilities. An increasing number of students (especially the ones working at the same time) welcomes this opportunity to personally decide and plan how and when to study. Time and place related flexibility also brings challenges for the team-building and collectiveness as well as requires self-organizing skills from the students. These issues have been discussed in other chapters of this article, and have close relationship with the flexibility.

There is another aspect to flexibility; when knowledge is built collectively as a result of the input of all participants, the final outcome can be very different from course to course. The teacher can influence this via constructive feedback, but also has to accept the fact that the learning results cannot be specified beforehand as strictly as when using traditional teaching approaches.

The results from the pilot course showed that the more the students worked in groups, the better results they obtained. During the group discussions the collective understanding of the business strategies developed faster than with students working alone. However, the pilot course included students who were not able to participate in live group discussions. For them the flexible structure offered a virtual possibility to be an active member of the course.

2. Organizing and operating the simulation game

2.1. Tasks

Being an active and connected member requires self-discipline, and this should be emphasized by the arranging organization in the very beginning. The virtual teams need clear definitions on what is expected from them in the areas of roles, responsibilities and rules for the team and to each of its members. Well-specified tasks, activities and timelines support the teams' progress, and constant feedback is required from the teacher to monitor and encourage the teams. Special attention should be paid to those students who are the least interactive in order to make them more active members of their team. [3]

It is not enough to define the tasks related to the content and substance of the course. In a virtual learning environment the communication between students and teams is a vital form of generating knowledge and enhancing learning. Therefore the teacher needs to include a clear specification for the communication in the form of tasks including timelines. It is also important to organize the course materials in a logical and clear way in order to avoid confusion. [3] [6]

2.2. Feedback

As learning in a virtual environment is an interactive process, feedback from the teacher cannot be over-emphasized. The feedback should encourage the hesitant students, guide students searching for the way to proceed and help the students feel they can complete their tasks. Successful feedback reduces: i) the distance between the students and the teacher when it is not too formal and rigid; ii) mistrust and insecurity when it is supportive and positively-formed; iii) the feeling of remoteness and distance when it is frequent and prompt [6].

In an e-learning environment the feedback should be seen more as an interactive task to enhance learning. The learning does not involve the students, but in a student-oriented methodology with collective knowledge building, the teacher is also a learner. Another important issue to discuss is the feedback between students. The teacher should encourage the team members to give feedback to the input of their fellow students in a constructive way. As the knowledge is built from the collective input, a discussion-oriented communication should be emphasized. [4]

2.3. Supervising

Supervising a virtual student group requires at least as much attention from the teacher as a group studying in a traditional classroom environment. Supervising is not only a task for the teacher, but also for each student. With constant communication the learners can monitor the progress of other students and teams. [3]

With continuous supervising the teacher also shows respect and consideration towards the students, their progress and achievements. This kind of encouragement is necessary for the students to remain motivated. Supervising should also be about cooperation of the teacher and the students instead of being a form of control. [4]

2.4. Sense of community

The physical distance also creates challenges, as the feeling of belonging to a community is important for the participants. The usage of multiple teaching methods and web based systems in combination supports the group building and the sense of belonging. Face-to-face contacts are a very important addition to the virtual study communities, and should be arranged whenever possible and feasible. [3]

In order for the virtual team to be productive, the input from all members is required. The physical distance poses a challenge for keeping the communication alive in a constant mode, especially for those members who do not feel to be a part of the team. This highlights even more the importance of the team building activities and the development of good social ties within the virtual teams and communities. A key task for the teacher is to increase the collective identity instead of encouraging the students to target for individual achievements. [3]

When studying in a virtual team, it is important that the members get to know each other in more ways than just a name or a user id. This could be facilitated by face-to-face situations, but also with informal activities in the virtual communities, such as chats, discussion forums and voice-enabled meetings (Skype, Webex etc). Good team building is seen to enhance the learning and knowledge building inside the teams, and the arranging organization should support this activity by developing competencies in social and technological areas. It is important to consider the interaction between students, as the learner-learner communication is an important source of individual learning. [3]

3. Experriences and learning results from the pilot course

The simulation game consists of a case company operating in global mobile phone markets, so the products are familiar and interesting for the students. Teams of 3-5 students form a company, and the teams compete against each other in a common, global market. Decisions are made in areas like demand (predictions), production, pricing, R&D, logistics and finance. The teams can see how their decisions influence their company's financial situation, but only after each game round is over, the outcome of that round is shown.

The software itself does not require complex skills, but the teachers should of course possess comprehensive knowledge of the business environment and its areas. We also found out that it is useful to play the game at least once through several rounds in order to get acquainted with the playing process. As each round has some specific market conditions, it is easier to guide the students, when the teacher knows what to expect.

Today's students are familiar with web based tools, so the user interface did not require any teaching as such. Adjusting the timing for the game requires some practicing for the teachers, and after the pilot course we realized that in the beginning we should have used more practice rounds, explained the connections between different decision-making areas and gone through the practice round results more thoroughly. The importance of classroom sessions was clearly seen, as it provided an opportunity for the students to get to know the game environment and the other team members. This supports the literature findings, which show that the combination of face-to-face and virtual learning brings the best results.

As a game can never represent the real world perfectly, some compromises have to be accepted. Some decision-making areas may seem as oversimplified, while others may seem as overemphasized. The game in question is a strategic simulation game, so long-term decisions like R&D are highlighted more than daily operations. The variations of students background education and experience brings both a challenge here as well as an opportunity; some students and teams might need more attention in areas that are not familiar for them, while the opportunity lies in the collaborative learning in teams with different knowledge based members.

The idea of the game is that each student makes their own decision set, and the team decides whose decisions are finally accepted as the team's decisions for each round. One of the main outcomes was that the students were able to work in multicultural teams effectively. The competitive aspect of the game gave additional motivation to the student teams. Having different level of activity within the groups gave the teachers a task of motivating the silent individuals and thereby ensuring an equal level of participation of all team members. Some teams worked via physically joint sessions, some only worked via the virtual platform. This did not seem to provide big differences in the results – bigger differences seemed to be between active and passive teams regarding collaborative working.

In the end of the game one team was announced to be the winner according to the specific criteria of the game. However, the students had to analyze their decisions; where they succeeded, what they could have done differently. The analyses was formed into a report and presented in a final seminar. This provided the students with understanding of the corrective actions which would have been needed for improving their results. According to the feedback from the teams, analyzing the results and the impact of the teams' decisions created the biggest learning experiences. It also showed that some teams had a clear strategy throughout the whole game, while others adjusted their decisions based on the previous round's results. Severe set backs of some teams forced the teams to make dramatic changes in their decisions, and the teams found this also an educational situation.

Conclusions

Encouraged by the positive experiences of this simulation game, we will proceed with it. In order to widen the experience we have started co-operation with our partner schools from other countries, and are targeting to a common simulation game with teams from different countries competing in the same virtual market. There are various alternatives in setting up this co-operation – the extreme being a simulation game including teams consisting of participants from different countries. Regardless of the set-up the challenges of working in the virtual learning environment are the same as with the pilot course – the distance and cultural differences of each participating school bring an extra level of complexity.

The experiences have enforced our belief that the importance of face-to-face start should be emphasized. In a co-operational implementation of several schools this means that teachers of each school have to be involved and committed to guide their students and teams to the game.

In order to gain wider understanding of the advantages and disadvantages of different e-learning models, more experience-based research is needed. Comparable studies from various business games and e-learning platforms should be done to get deeper knowledge and more comprehensive view of best practices in this field. The demo version of the business simulation game used in this article can be found from Ref. [7].

References

[1] Laari-Muinonen Leena. Best practices in web based education. Seminar presentation. Lappeenranta, Finland: Saimaa University of Applied Sciences, 2008.

[2] Palloff, Rena M. Lessons from the Cyberspace Classroom : The Realities of Online Teaching. Somerset, NJ, USA: Jossey-Bass, Incorporated Publishers, 2001.

[3] Ferris, S. Pixy(Editor). Teaching and Learning with Virtual Teams. Hershey, PA, USA: Information Science Publishing, 2005.

[4] Mason, Robert and Rennie, Frank. E-Learning and Social Networking Handbook. Resources for Higher Education. New York, NY, USA: Routledge Publishing, 2008.

[5] Aggarwal, Anil(Editor). Web-Based Learning and Teaching Technologies: Opportunities and Challenges.Hershey, PA, USA: Idea Group Publishing, 2000.

[6] Albalooshi, Fawzi(Editor). Virtual education: cases in learning & teaching technologies. Hershey, PA, USA: Idea Group Inc., 2003.

[7] [http://www.cesim.com], retrieved 2009 02 05.

VILNIUS BUSINESS COLLEGE

VALIDITY OF PSYCHOMETRIC TOOLS IN SOCIAL SCIENCES: MEASURING GRIEVING CHILDREN VALUE SYSTEM

Tomas Butvilas*
Department of Education Activities, Mykolas Romeris University,
Ateities Str. 20, LT-08303 Vilnius, Lithuania

Received March 25, 2009 , accepted August 21, 2009

Abstract: This paper deals with some theoretical approaches of a modern definition of research validity and its importance to the other basic research quality level – reliability. It is stated that validity is mostly described as a main prerequisite to make conclusions not only based on research instrumentality and the set of respondents but mainly from the whole research. Also, based on the former research about grieving children socialization, these children value internalization system and the construction of such model are revealed as well. Consequently, grieving children value system is shown through their behavioural patterns: they less than the others internalize openness, believing in self and others, sensitiveness, respect, self-control and activity. Generally, the main socio-moral values, that make the basis of favourable socialization for children (age 7–11), consist of openness, believing in self and others, sensitiveness, respect, self-control, solidarity and activity.

Keywords. Validity, psychometrics, grieving children value system, psychometric tools.

Short title. Validity of psychometric tools.

* Corresponding author, e-mail: tbutvilas@mruni.eu

Introduction

Most of the socialization research is directed towards the individual, as social being development and his/her participation in society. Thus, the socialization is the essential basis of an individual's participation in a particular society, as well as expression, according to which, the relations with surrounding environment and its events are revealed. On the other hand, socialization is quite a difficult psychosocial process. In a fast developing society it has become more difficult for children to socialize adequately. Various social changes (sudden growth of technologies and science, mobility, urbanization, individualization etc.) have an influence on the participants and determinants of socialization [1–3].

The recent researches have shown that children from incomplete families are more prone to the aggressiveness than others from nuclear families [4]. On the other hand, after the loss of one parent children of age 6–7 and even younger suffer from feeling guilt i.e. they usually blame themselves for the things that have happened [5]. Thus, the loss in the family with its negative emotional charge also makes it more difficult for a child to identify himself/herself and consequently limits the value of internalization. As a matter of fact, the loss impedes the socialization processes of such children. Many authors, for example Berns [6] et al, points out the values internalization as the basis of a child's socialization, because he/she seeks to identify with close people and intercept their values. Thereby, a problematic question arises: i) on which methodological basis we could explore such children value system, and ii) which methodological approach to construct such psychometric system would be the most appropriate.

This work is basically devoted a) to present some of theoretical approaches of a modern definition of research validity and its importance to reliability; b) to reveal the grieving children socialization aspect, their behaviour, through these children value internalization system and to show the construction of such model.

1. Literature overview

Psychometric tools validation process. Psychometric instruments have been used to measure and reveal some psychological differences between people for many years. In these days, instruments tend to be slightly more sophisticated and can deliver genuine insights into individual and team capabilities and behaviours. Thus, over 70 % of larger companies are currently using psychometric instruments to gather and interpret vital information to help them to maximize their people's potential. Also if psychometric tools are used quite professionally they can help an organization make effective use of its people and to improve business performance. Actually, psychometric profiling is accepted as a tool that can enhance: managerial decision making during recruitment and selection; individual and team development; managerial development, and organizational change. Putting it in other words, psychometric tools are questionnaires or tests that have some predictive ability, that is, the ability to predict outcomes in the workplace [8].

To become psychometrically valid, the questionnaire or test must go through a rigorous process of validation that is usually carried out by a workplace psychologist who specialises in psychometric evaluation. According to psychometricians Sabesan [9], Anastasi&Urbina [10], examples of psychometric tools are as follows: i) numeric tests; ii) literary tests; iii) colour blindness tests; iv) crime predictor tests; v) personal behavioural styles tests; vi) personal values tests; vii) stress response tests; viii) safe working capability tests etc. They are used as inputs to overall processes. Thus they should have a predictive validity and tolerance score for each of the elements of the test or questionnaire.

Meaning for data reliability. Validity measures how appropriate a test is for a specific purpose. A test may be considered valid for one use and invalid for the other. According to Bitinas et al. [11], there are only two levels that show research quality – validity and reliability. Although validity is in most cases only a desirable pursuit. Actually, as mentioned authors would state, contemporary definition of validity says that this research quality level is mostly described as a pure prerequisite to make conclusions not only based on research instrumentality and the set of respondents but mainly from the whole research.

Also a research paradigm may influence conclusions direction as well [12]. Thus validity is the degree to which evidence and theory support specific interpretations and uses of test scores [13]. Although there are many types of validity, a screening tool should test for content validity and concurrent validity. Content validity tests a developmentally appropriate range of skills. Concurrent validity shows how well a screening tool compares to other similar measures in the market [14]. According to DeVon [15], planning for psychometric testing through design and reducing non-random error in measurement will add to the reliability and validity of instruments and increase the strength of study findings [16]. Underreporting of validity might occur because of small sample size, poor design, or lack of resources. Lack of information on psychometric properties and misapplication of psychometric testing is common in the literature (for instance, Bitinas et al).

2. Grieving children value system on their behaviour level: designing, measuring and validating the instrument

The research object is the grieving children value system and appropriate methodological approaches while validating the psychometric tool/model of these children *Value Internalization.*

The goal of this paper could be expressed as follows: i) to reveal some theoretical basis on psychometrics validation and, on the other hand, ii) to show the modelling as well as analysis processes of these children value system.

Main research method could be entitled as a factor analysis (using inner compatibility degree – Cronbach α) of grieving children internalized values. All calculations were provided using MS SPSS 16.0 program [17] according to mentioned Cronbach α formula:

$$\alpha = \frac{N}{N-1}\left(1 - \frac{\sum_{i=1}^{N}\sigma^2_{Y_i}}{\sigma^2_X}\right), \qquad (1)$$

where *N* is the number of components (items or testlets), σ^2_X represents the variance of the observed total test scores, and σ^2_Y represents the variance of component *i*. Another alternative way to define Cronbach α formula coud be provided by Eq(2):

$$\alpha = \frac{N \cdot \overline{c}}{\overline{v} + (N-1) \cdot \overline{c}}, \qquad (2)$$

where N represents the number of components (items or test lets),   equals the average variance and   is the average of all covariance between the components.

Referring to other works [18-20] and to the data of a pilot study, the model of internalized values was set along with the content of those values and factor analysis. As Darlington [21] points out, factor analysis includes both component analysis and common factor analysis. This method is used to study the patterns of relationship among many dependent variables, with the goal of discovering something about the nature of the independent variables that affect them, even though those independent variables were not measured directly. Thus, answers obtained by factor analysis are necessarily more hypothetical and tentative than true when independent variables are observed directly. Within this research basically the PCA method was used, i.e. there are two main methods of extracting the factors from a set of variables that are given bellow in the 1st table in the form of values and their manifestations: common factor analysis (known as principal factor analysis, PFA) and principal components analysis (PCA) [22]. There was no need for setting up a causal model but simply to reduce a large number of items (29 empirical indications) to a smaller number of underlying latent variables (4 groups of factors, see Table 1). Algorithmically factor analysis using PCA method was used (that is the most common extraction technique).

3. Main results and discussion

M. E. Young [23] within his research on models of internalized values in classroom environment states that internalization can be fostered by discipline, teaching, example or modelling, social reinforcement of appropriate behaviour and arrangement of the environment so that desirable behaviour is naturally elicited. The development of a sense of self, ability to self-regulate, and attachment are important precursors of internalization, continues the author. Generally internalization is based upon the child's accurate perception of the message and either the acceptance or rejection of that message. Internalization of values is observed when acceptable behaviour is generated from intrinsic factors and not from the anticipation of external consequences. Two things are required to internalize values: i) clarity of universal and personal values – one must be convinced that universal values are valid and truly worth pursuing, and also that the personal values are clear and strongly felt; ii) contrary conditionings are neutralized (Self-Transformation , [24]). The conditionings to be neutralized are of two kinds: i) physic-emotional conditionings – those involving habits and emotional reactions, such as fears, resentments etc.; ii) mental conditionings – those molded by cultural values, such as the measurement of success and failure, philosophy of life, etc. They create preferences for lifestyles, modes of action etc. When true clarity is achieved, and conditionings are comprehensively reviewed, then values can be fully integrated into one's life with minimal difficulty.

We have performed research when a pilot study was organized and held back in 2005 while main losses and these children personality characteristics were set (see Table 2). This table, as mentioned above, shows exactly the main factors of grieving children value internalization that evidently create a model of these children desirable value system - 1st factor - communicativeness and cooperation, 2nd factor - self control and self-esteem regulation, 3rd factor - locus of control, 4th factor - empathy. Locus of Control refers to the extent to which individuals believe that they can control events that affect them. Individuals with a high internal locus of control believe that events result primarily from their own behavior and actions.

Empirical indications mean the particular behaviour of grieving children that appears in certain social situations. Values manifestations indicate smaller variables that come out from main values stated in the Table 1.

As we can observe from the table above, factor analysis (using Principal Component's method - PCA - and Varimax rotation with KMO normalization – 0.79, and Bartlett's test – $\chi^2 = 3485.88$ $p < 0.000$) has shown that the majority of those empirical indications are rather important. The inner compatibility's degree of this model is very high as well (Cronbach α = 0.91). Thus, the main internalized values for grieving primary school children would be as follows: openness, believing in self and others, sensitiveness, respect, self-control, solidarity, and activity.

Other researches, such as E. Scott's [25], show that some of the children are reared under the dual influence of primary (parents) and secondary (nursery teachers) socializing agents, and some of them are reared under the influence of primary agents only. In E. Scott's research two groups of children were compared having in mind their internalization of the cultural values of self-reliance, cooperation, and compliance; their integration among these internalized values and their congruity between conformity and internalization with respect to each value. The following conclusions were made where nursery children were found to have greater internalization of self-reliance and cooperation, to display higher value integration and greater congruity with regard to compliance. Intelligence was positively correlated with internalization of each value and with integration. Level of maternal expectations was positively correlated with internalized self-reliance and cooperation. Perceptions of expectations were positively correlated for the same values. Maternal acceptance of the child was positively correlated with internalization of cooperation and compliance and with integration, while it was part of a higher-order interaction effect with maternal expectations and treatment group in the case of internalized self-reliance. Thus in our research the model of grieving children socialization reveals its strengths on the aspects of indicated such children behaviour and main both universal and practical values along with their manifestations.

While comparing the internalized value system of those who have experienced loss in the family and those who did not, some behavioural differences were identified. It was set that grieving children less than the others internalize openness, believing in self and others, sensitiveness, self-control, and activity (see Table 2).

Table 1. Model of internalized values for the primary school children

Values	Their manifestations	Empirical indications	Factors
Open¬ness	Sincerity	1. Shares with others (his/her knowledge, stuff, impressions, feelings) 2. Doesn't cheat	0.63 0.82
	Straightforwardness	3. Doesn't lie (says the truth) 4. Accepts his/her own mistakes	0.75 0.67
Believing in self and others	Self-confidence	5. Doesn't avoid harder tasks 6. Tries again while not succeeded	0.71 0.60
	Relying on others	7. Tells about him/herself 8. Asks for others' opinion	0.52 0.43
Sensiti¬veness	Compassion	9. Shows sympathy (comfort etc.) 10. Makes no harm	0.72 0.88
	Advertency	11. Listens to others 12. Shows their merit	0.80 0.72
Respect	Respect to others	13. Doesn't humiliate 14. Behaves naturally with others	0.52 0.56
	Self-respect	15. Doesn't boast of smth. 16. Keeps the word	0.64 0.76
Self-control	Self-control	17. Finishes task until the end 18. Doesn't burst while not succeeded 19. Controls behaviour	0.56 0.61 0.54
	Patience	20. Doesn't groan in case of misfortune 21. Doesn't bother during the classes	0.58 0.74
Solida¬rity	Forbe¬arance	22. Considers others 23. Doesn't revenge	0.73 0.81
	Peacefulness	24. Agrees with teacher, peers etc. 25. Doesn't use any violence while facing conflicts etc.	0.50 0.68
Activity	Initiative	26. Decides what to do 27. Gets involved into the activity voluntarily	0.76 0.51
	Self-sufficiency	28. Suggests the ways of doing certain things 29. Completes the work voluntarily	0.43 0.52

Boys in this case were affected more than girls by the experienced loss in the family (see fig. 1). The results, as it is shown in table 3, have confirmed that grieving boys less than the others internalize the openness (χ^2 = 84.14 p < 0.000), believing in self and others, sensitiveness (χ^2 = 69.18 p < 0.000), respect, self-control (χ^2 = 33.16 p < 0.03), solidarity (χ^2 = 29.23 p < 0.02), and activity. These values are more obvious within non-grieving children behaviour.

Meanwhile grieving girls behaviour in many cases differs much than boys and other classmates (see fig. 2). The data shows that grieving girls quite more than the others – non-grieving classmates – practically internalize solidarity, respect, activity, and openness.

The significant correlation between these two groups of participants was not found.

Conclusions

1. Validity measures how appropriate a test is for a specific purpose. A test may be considered valid for one use and invalid for the other. Validity is mostly described as a main prerequisite to make conclusions not only based on research instrumentality and the set of respondents

Table 2. Statistically meaningful internalized values based on experienced losses

No.	Parameter	Chi square $\chi 2$	Statistic significance p
1.	Sensitiveness	42.39	< 0.002
2.	Self-control	87.52	< 0.000
3.	Activity	31.67	< 0.005
4.	Openness	84.14	< 0.000
5.	Believing in self and others	69.18	< 0.000
6.	Solidarity	29.23	< 0.02

but mainly on the whole research. Therefore based on the former research about grieving children socialization these children value internalization system and the construction of such model were revealed.

2. The tested and validated psychometric tool/model of Grieving Children Value Internalization helped to identify the main four factors that are socializing and cooperation, self control and self-esteem regulation, locus of control and empathy. This helps to analyze further grieving children socialization on their behaviour basis having in mind the value of internalization as the main individual socialization factor and bridge.

3. The main socio-moral values that make the basis of favourable socialization for children (age 7–11), were distinguished. The list of such values consists of: openness, believing in self and others, sensitiveness, respect, self-control, solidarity and activity. However, it was set that grieving children less than the others internalize openness, believing in self and others, sensitiveness, respect, self-control and activity.

References

[1] Butvilas Tomas. Children (Age 7–11) Who Have Experienced Loss In The Family – Socialization and Educational Assistance Strategies. PhD Theses – Šiauliai University Publishing House, 2006.

[2] Butvilas Tomas. Šeimoje netektį patyrusių vaikų socializacija. Monografija. (in Lithuanian) – Vilnius: Mykolo Romerio universiteto leidykla. 2008.

[3] Gibson&Blumberg. Growing Up: Readings on the Study of Children. – Reading, Mass.: Addison-Wesley. 1978.

[4] Goldman, Linda. Helping the Grieving Child in School. – Phi Delta Kappa Fastbacks. Vol. 460. 2000.

[5] Mishara Brian et al. Helplines and crisis intervention services: Challenges for the future. In: D. Lester (Ed.). Suicide prevention resources for the millennium (pp. 153-171). - Philadelphia, PA: Brunner-Routledge. 2001.

[6] Berns, M. Roberta. Child, Family, School, Community: Socialization and Support. – Wadsworth Publishing Co Inc; International ed. edition. 2009.

[7] [http://www.tms-international.com/News/Pharma_marketing_-_psychometric_tools.pdf], retrieved 19-01-2009.

[8] [http://www.hrworkbench.com/eng/l-psychometric-tools.html], retrieved 19-01-2009.

[9] [http://www.indianmba.com/Faculty_Column/FC276/fc276.html], retrieved 19-01-2009.

[10] Anastasi, Anne and Urbina, Susana. Psychological Testing. Prentice-Hall of India, 7th ed. 2003.

[11] Bitinas Bronislovas et al. Kokybinių tyrimų metodologija (in Lithuanian). – Klaipėda: Socialinių mokslų kolegija. 2008.

[12] Bitinas Bronislovas. Edukologinis tyrimas: sistema ir procesas (in Lithuanian). – Vilnius: Kronta. 2006.

[13] [http://www.psychometrictools.co.uk/validating.php], retrieved 27-01-2009.

[14] [http://www.firstsigns.org/screening/tools/psych.htm], retrieved 20-12-2008.

[15] DeVon et al. A psychometric toolbox for testing validity and reliability. -Journal of Nursing Scholarship, 39(2), 155-164. 2007.

[16] [http://www3.interscience.wiley.com/journal/118508777/abstract?CRETRY=1&SRETRY=0], retrieved 16-12-2008.

[17] MS SPSS 16.0 – SPSS Statistics is a modular, tightly integrated, full-featured software comprised of SPSS Statistics Base and a range of add-on modules: SPSS Advanced Statistics, SPSS Categories™, SPSS Complex Samples™, SPSS Conjoint™, SPSS Data Preparation™, SPSS Exact Tests™, SPSS Neural Networks™, SPSS Missing Values, SPSS Regression, SPSS Custom Tables, SPSS Forecasting, SPSS Decision Trees, and SPSS EZ RFM™ [http://www.spss.com/statistics/], retrieved 09-02-2009.

[18] Klaassen Marcel. Education and Citizenship in a Post-Welfare State. – Curriculum, 17, 2nd December. 1996.

[19] Gecas. Contexts of Socialization. Social Psychology: Sociological Perspectives. – New York: Basic Books. 1981.

[20] Campbell&Stanley. Experimental and Quasi-Experimental Designs For Research. Chicago: Rand McNally College Publishing Company. 1973.

[21] Darlington B. Richard. Factor analysis. [http://www.psych.cornell.edu/Darlington/factor.htm], retrieved 09-02-2009.

[22] [http://faculty.chass.ncsu.edu/garson/PA765/factor.htm], retrieved 26-01-2009.

[23] Young E. Mark. A Classroom Application of Grusec and Goodnow's Discipline Model of Internalization of Values [http://www.questia.com/googleScholar.qst;jsessionid=JKLSnvwyHC1Cw2NLhn87dJKGd6nQ 7N6HhR5Q1d6bvrm3MMvYQBLk!111030713!1530467563?docId=5000309386], retrieved 12-03-2009.

[24] [http://www.selftransformation.net/articles/values.htm], retrieved 14-03-2009.

[25] Social Value Acquisition in Preschool Aged Children III. Internalization of Institutionalized Value Expectations [http://www3.interscience.wiley.com/journal/119700455/abstract?CRETRY=1&SRETRY=0], retrieved 24-03-2009.

VILNIUS BUSINESS COLLEGE

Vilnius Business College
Kalvarijų str. 125, Vilnius, LT-08221, Lithuania
Tel./ fax. +370 5 215 4884, Mob.: +370 616 55422, +370 615 96680