ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2014. Vol. 1. No. 1. 11–22

Business process modelling using ontological task models

Y. Burov

Department of Information Systems and Networks, Lviv Polytechnic National University,
79012, Lviv, Bandery st. 12, Ukraine
e-mail: eugeneburov01@gmail.com

Received January 26.2014: accepted February 20.2014

Abstract. Knowledge-based technologies form a basis for creation of flexible intelligent enterprise. Methods of business process modelling using ontological tasks models allow to create an active modelling system to capture and reuse knowledge patterns in form of ontological models while preserving overall consistency via common ontology usage. High-level business process ontology is presented. Models for business and decision making operations along with their representation in system are specified. Business process modelling tool using task models is described.
Key words: ontology, business process, modelling, task model, intelligent enterprise.

INTRODUCTION

The concept of business process is widely used for analysis and enhancement of enterprise economics. Despite the fact that the term has been firs used in 60's, considerable attention to the study and modelling of business processes has been given to only in 90's of the twentieth century. One of the earliest works in this area [1] applied methods used in technical systems design and analysis to business processes.

Business process analysis approach focuses on process structure, dependencies between operations, resources rather than functions and procedures as earlier approaches. It strives to enhance business process as whole. Improving complex business processes is seen as a major reserve for increasing the efficiency of the enterprise. This view is reflected in the approach to the restructuring of business processes (Business Process Reengineering - BPR) [2], which was popular in the 90's and currently exists in the form of business processes management (BPM) [3].

Today, BPM is considered as the method of the flexible adaptation of business processes to customer needs. BPM attempts to continuously improve processes in the enterprise.

Current approaches to the evaluation and certification of enterprise manufacturing process (ISO 9000, CMMI) is largely based on the analysis and evaluation of business processes quality for certified company [4].

Currently the problem of analyzing and documenting business processes is already sufficiently resolved. Methodological approaches to document business processes, formal languages and software tools for modelling are developed and used [5,6]. Approaches to measure enterprise activity using multiple criteria are researched [7].

There are many languages for building business models that differ in purpose and notation, approach, the possibility of constructing a code from models [3], such as UML, BPMN, BPEL.

In addition to formal languages, modeling tools are becoming increasingly popular. The examples, of such systems are SAP Business Objects [8], MS Business Intelligence [9], Software AG ARIS [10]. These systems allow analyze and obtain value from corporate databases, existing CRM and ERP systems, evaluate the performance of individual operations and business processes, provide decision-makers all the necessary information.

Thus, the main trend in the business processes management is the transition from purely manual documenting, their structures analysis and decision-making to computer-assisted solutions aimed at in-depth study. However the business process simulation tools usually are not reflecting business situation in real time and don’t support operative business control decisions.

Decision making is an important operation type in business process. It is performed by domain area experts based on their assessment of current business situation. Decision support process include such steps as understanding a problem (situational awareness -SA), designing possible solutions, evaluating and implementing them. Decision support area is often considered separately from business process modelling [11].

Decision making theory includes classical decision making (CDM), behavioral decision theory (BDT), judgement and decision making (JDM), organizational decision making (ODM) and naturalistic decision making (NDM) [12].

Classical decision support methodologies are considering decision making as a 'choice' process, when expert selects one optimal according to selected criteria solution from a set of available solutions. However there is no guarantee that selected solution will be feasible [12]. Classic approaches has been successfully applied to solve structured decision problems with defined goals, conflict resolution, computational complexity, and requiring optimization.

Naturalistic decision-making requires (NDM) proficient decision-makers and is based on matching patterns of current business situation with patterns derived from experience. If two patterns match, than solution from experience can be applied. NDM usually provide feasible solutions and works for unstructured or semi-structured problems, problems with uncertainty and ambiguity [12].

A large part of decision making follows established patterns and can be automated if those patterns are revealed and specified formally. The high rate of change in business environment requires the building of flexible business processes which are able to react and adapt to changes. This can be achieved by building active business modelling systems where changes in business environment and processes are reflected and processed in real time in order to generate viable solutions.

Active modelling systems for different domains are now developed, starting from conceptual models [13]. On the other hand, the importance of providing real-time analysis and recommendations is formulated in the concept of second generation of business intelligence systems (BI 2.0). It requires including business analytics and decision making to be included in business process and when appropriate to be executed automatically [14].

One of the important assets of modern enterprise is its corporate knowledge. It is often incorporated in numerous loosely coupled documents, existing as employee's experience, stated or implied business rules. This knowledge currently is managed poorly, it is not structured and often at least partly, lost. The effectiveness and flexibility of enterprise can be substantially increased if corporate knowledge will be elucidated, formally specified and documented, made accessible to employees and used as a basis of construction of intelligent software and business process management systems.

The importance of knowledge based systems is currently well understood. Quinn [15] stresses that successful companies will derive their competitive edge from highly developed knowledge assets and core service-based competencies, creating knowledge-based intelligent enterprises.

The commonly used approach for building knowledge-based systems is using ontology, defined by Gruber [16] as "explicit specification of a conceptualization". Ontology can be viewed as a hierarchy of concepts and relations from a specified domain. The usefulness of ontology can be augmented by specifying additionally some constraints and rules. Thus, the ontology provides a structured vocabulary for describing domain along with knowledge about general constraints and rules in domain.

The efforts to build knowledge-based system in business modelling domain resulted in creation of business ontology frameworks [17].

In [18] the use of ontology is perceived as a solution to a problem of lack of specified semantics in communication between humans and systems, which creates communication issues. Proposed enterprise architecture for intelligent enterprise uses three levels of ontologies describing business terms, architecture components and their relations. [19] uses ontology as a part of Enterprise 2.0 architecture to support social, open and adaptive views and stimulate flexibility, adaptability and innovation. In [20] is proposed an approach for construction of ontology-based semantic metrics from the thesaurus of a given domain.

Knowledge based approach is used not only in enterprise integration or business architecture modeling, but also for business intelligence and decision support. In [21] is shown that future of business lies in business intelligence systems that can make decisions, rather than producing reports or simply managing physical assets.

Authors [22] propose to redesign business intelligence solutions using approaches from semantic web development. So, they will be able analyze, process and synthesize solutions based on data meaning.

Knowledge based technologies are used on various stages and tasks of decision making process. For example, [23] research situation awareness in terms of OWL ontology, allowing to specify situations in a common language with computer handling semantics. In [24] ontology is used to provide for data integration and interoperability between different industries, facilitating decision-making process.

Despite a considerable contributed research effort to develop intelligent enterprise architecture and technologies, several areas and problems still are unclear and require a further investigation. For example, in order to implement real-time business intelligence and process modelling, knowledge in form of repeating patterns need to be captured, formalized, included in system and reused. Ontology as hierarchy of concepts does not support the usage of active, reusable patterns and actions.

In this paper we explore the approach of using ontology and ontological models for building active intelligent business process modelling systems.

This article has several parts. In first part we describe the current state of business process modelling and highlight the trends of intelligent enterprise creation. We show that knowledge processing technologies form a basis to build an intelligent enterprise with highly flexible business processes. In second part we present an approach to use ontology and task models in business process modelling. In third part a business process modelling ontology and basic models are described with examples from the domain of software development. Last part is dedicated to the description of software modelling tool prototype implementing the proposed approach.

KNOWLEDGE REPRESENTATION AND MODELLING SYSTEM STRUCTURE

At present, the main research in ontology modelling is focused on declarative ontologies - domain ontologies, and general ontologies [25]. While general ontologies may be useful for some classes of problems, such as automatic translation, building text digests, medical diagnostics etc, we believe that in order to capture and formalize repeating patterns in decision making or business processes some smaller unit should be selected as a base for knowledge elucidation, formalization and processing. Such smaller, self-contained contained unit in business modelling is business operation or task.

In order to formalize knowledge within the scope of particular task, special type of ontologies (task ontologies) are built. Historically, the tasks ontologies have been developed as a result of scientific analysis of tasks (task analysis). Methods of task analysis are used to define and formalize all factors that affect or are used in the process of solving the problem by an expert or executing specified business operation. Such methods are widely used for designing interfaces of computer programs, in expert systems, decision support systems [26,27].

Task analysis is focused on the analysis and specification of the components of common tasks, determination of its structure and constraints. This allows the expert to better understand the problem, identify possible errors and omissions. Expert can simulate the process of problem solving and task execution and is able to evaluate the results of simulation in order to gain and pass the knowledge to other experts.

The area of task analysis has experienced a significant change with the advent of ontologies. It was proposed to use task ontologies to formalize the concepts and relations for the any given task [28].

Unlike other types of ontologies, such as general or domain ontology, task ontology

- is created for some class of tasks;
- the concept of task goal is important and its formalization mandatory
- the concept of action is introduced [29] in the context of task execution;
- task ontology modelling environment provides execution (or simulation) of actions;

Tasks ontologies research area is closely related to conceptual modelling, because in the process of building of task ontology expert actually creates a formalized conceptual model for task [30]. An important aspect of both conceptual and ontological modelling is the interaction with domain expert who creates and validates an ontology.

In the process of tasks ontology research were implemented simulation environments allowing users to create and execute ontological models for specific classes of tasks. The most advanced of these environments is CLEPE (Conceptual level programming environment) [25]. However, available research is focused on studying tasks ontologies for different tasks separately.

In intelligent enterprise multiple interacting tasks should be performed. In order to make possible this interaction and knowledge reuse, a common for all tasks ontology should exist, describing all common concepts, relations and constraints for enterprise domain. Thus, task ontologies in intelligent enterprise should be built based on concepts of common enterprise ontology. If task analysis reveals that some concept or relation in task representation is missing in common ontology, then this missing element should be included in ontology.

On the other hand, task ontologies are useful as basis for capturing knowledge patterns, action taking and performing business operations. Those ontologies can also include task specific knowledge, which for various reasons was not incorporated in common ontology. In order to differentiate self-contained task ontologies from task ontologies built using common domain ontology, we will call the last one as "ontological model".

Intelligent, knowledge-based business process modelling system has at its core a knowledge base. The introduction of ontological models adds an ontological models repository to well known structure of knowledge-base [31]. On figure 1 is shown the structure of modelling system which uses ontological models.

Knowledge base contains a common ontology, the repository of ontological models, information base. Information base stores information about instances of types (facts) defined by ontology entities and relations. It also stores models initialized with facts (fact-models). Ontological models serve as templates to create fact-models, providing reusable pattern knowledge about ways to execute tasks. Fact models contain current working data pertinent to task execution. Thus, fact-models could be considered as task in process of execution. When the execution of task is completed, corresponding fact-model is removed from information base and archived.

When new facts or fact-models are created or modified corresponding constraints from common ontology and models are checked and enforced, providing data consistency.

Complex tasks are executed as result of combined models execution – model can call another model if some subtask needs to be done.

The functioning of modelling system is supported by a number of services. Model Execution Manager monitors external business events, creates, initializes fact-models and supports their execution. Service Broker supports the interaction between models. Information provider implements the search of facts according to given criteria.

In the process of fact-models execution real-world business operations are initiated as commands sent to external services. These services for example, could be implemented within network or local operational system or as a web services built according to the requirements of SOA.

BUSINESS PROCESS ONTOLOGY AND ONTOLOGICAL MODELS

Business process ontology is providing a common set of concepts and relations used for model implementation. The specific set of entities and relations specified in ontology in general case depends on selected domain and ontology competence. Thus, in [17] is proposed an ontology for modelling such important aspects of business architecture as produced product properties, customer relationship management, financial and value flows.

In our work we focused on operational aspects of business process which is prime candidate for automation – business process implementation layer [17]. We built a top layer of business process modelling ontology. It will be later enlarged with components describing specific business domains. As example we mostly use the domains of software development and high school education. As a base we used BPMN language modified and enlarged with some additional objects. Most important entities of business process modelling ontology are shown in Table 1 and relations – in Table 2.

Business process model specifies logical order of operations which are performed in business process. Those operations exist as separate entity types in ontology. We defined such entity types as BusinessOperation, BusinessEvent and Decision types. All those entities inherit from BusinessObject ontology entity.

BusinessOperation entity corresponds to specific stage in business process, when a specified change in business environment occurs and pre-planned result obtained. BusinessOperation has start and completion times properties. With it are associated a metric of success and references to procedures or models for success evaluation.

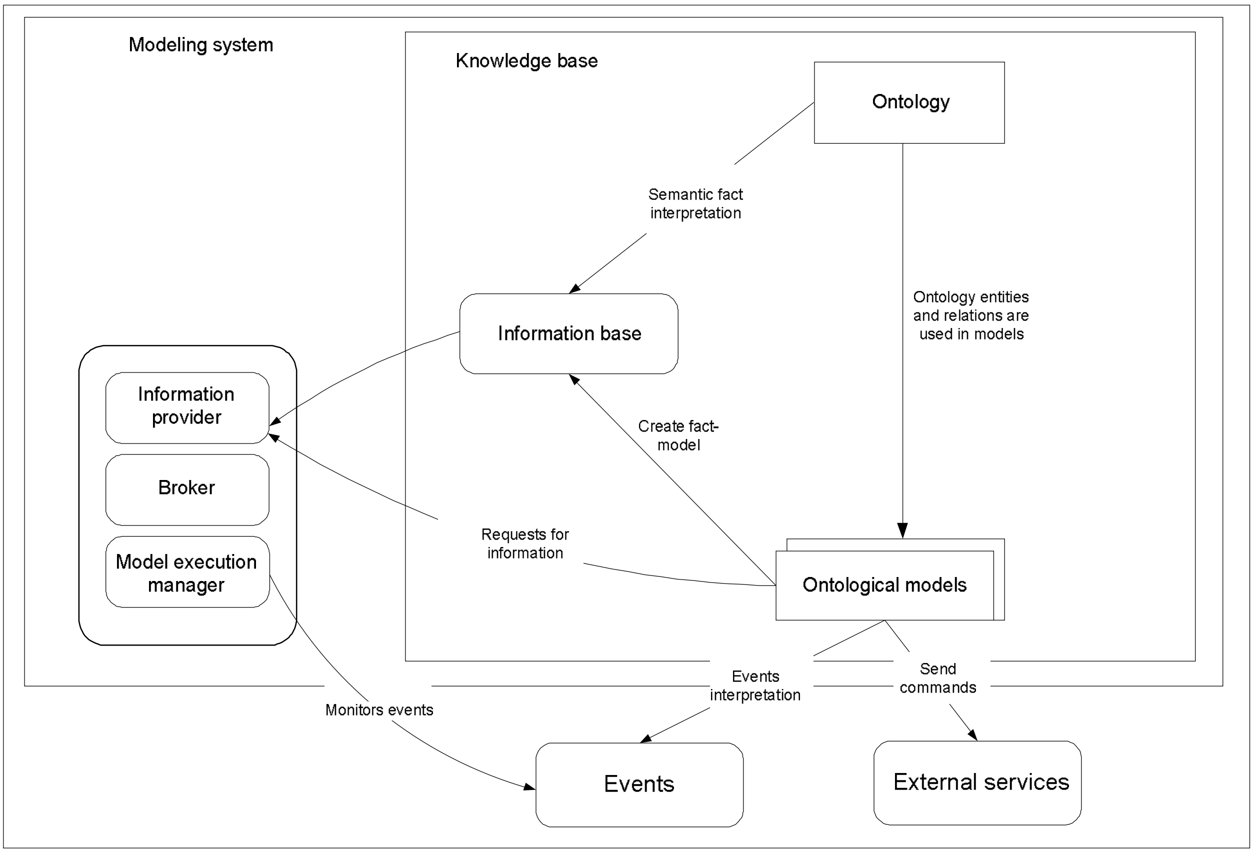

Fig. 1. Structure of modelling system

Table 1. Entities of business process modeling ontology

#	Entity name	Description
Organizations and persons		
1	Person	Human person
2	Organization	Unit performing business processes
3	Actor	The role in a model associated usually with persons or systems
4	Position	Specifies rights and requirements associated with position
Resources		
5	Resource	Something needed for business process execution
6	Resource with controlled access	Resource with limited access
7	Material resource	
8	Financial resource	
9	Workforce resource	
Processes and operations		
10	Business process	Process which results in creation of artifact or service important for a customer
11	Business operation	Limited in time part of business process having expected result
12	Competence	Permanent access rights set
13	Decision	Decision making operation
14	Search for information	Stored search with specified search criteria
Events		
15	Business event	General business event
16	Meeting	
Artifacts		
17	Artifact	Generic artifact, such as document, software program etc
18	Document	Generic document
19	Contract	Contract as a document
20	Email message	
21	Software product	
Commands and services		
22	Service	Service, accepting commands from modeling system
23	Command	Particular command
24	Script	The list of commands
25	Model	
Rules and constraints		
26	Business rule	Organization-wide rule
27	Corporate constraint	Constraint defined by corporation
28	Constraint from law	Constraint which stems from a law
29	Best practice	Optional recommendation
Additional		
30	Annotation	Optional detailed description

Table 2. Relations of business process modeling ontology

#	Relation name	Description
1	Is part	Entity is a part of another entity
2	Is subclass	Entity can be considered as some, more general entity
3	Is after	Defines an order of business operation execution
4	Associate	Generic association
5	Use	Define resources used in business operations
6	Have access	Used in access control models
7	Own	Ownership relation
8	Initialize	Specifies the starting of execution of some model

If necessary, derived from BusinessOperation entities can be created, which expand the basic set of attributes, taking in consideration the specifics of business operation. For example, business operation Lecture has additional attributes specifying the name of lecturer, planned audience, subject and lecture room number.

Unlike business process model, BusinessOperation does not allow further decomposition. However, with BusinessOperation usually is associated ontological model describing how operation is executed, associated entities, resources, relations and operations. For example, with operation CopyFile is associated a model with such entities (roles) as RemoteLocation, LocalLocation, CopyMethod.

Depending from business operation content, ontology designer creates additional constraints to be met by the operation. These constraints are:

• pre-conditions - a list of conditions necessary for the start of operation

• restrictions on the maximum (or minimum) allowable execution time

• post-conditions - a list of conditions that must be true after the operation, to be classified as successful

BusinessEvent entity corresponds to any event, influencing and important enough to be included in model. Differently from business operation, business event does not have durability and is considered one-time object. BusinessEvents can be associated for example, with email arrival, project stage completion or project deadline. In common ontology BusinessEvent entity is derived from more general Event entity. Similarly to BusinessOperation, new entities can be derived from BusinessEvent, adding new information specific to event as attributes.

Decision entity in business process model corresponds to situation during process execution when there is a need to assess business environment and take decision, initiating new business operations. Such a situation usually occurs when some business operation has completed, or some business event occurs. Decision can be made either manually by designated employee or automatically, by model. In case if decision should be made by employee, model references the competence (requirements) to employee to be entitled to make this type of decisions. Alternatively, if decision is made by a model, Decision entity refers to a decision model. This model includes situation signature, and specification of actions. Situation signature is a list of assertions over facts from information base which should hold for the specified situation. If situation is detected, than actions are performed. For example, if in process of automatic software installing an error is detected, system administrator is notified by email.

Entities in business process model are linked with relations to form a whole. Among those relation types we will select relations which form the logical sequence of business operations. Those relations are IsAfter and Initialize. Other relation types, having informative function are Associate, Use.

Relation IsAfter links BusinessOperation with another one, or with Decision operation. The instance of this relation is created only if its starting business operation is completed. The availability of IsAfter relation instance allows start execution of a next BusinessOperation.

Initialize relation links Decision operation with one of several possible BusinessOperations selected as result of decision making process.

Associate relation links model with other models (such as normative or reference models), or artifacts used by model. Relation Use links model with resource used in process of model execution. This relation can have constraints describing the allowed amount of resource usage.

The domain expert works with modelling system using both textual and graphical business process model representation. Graphical representation allows model designer see business process as a whole. For a graphical representation of in the modelling system we used the notation of BPMN. The example of business process of automated software testing is shown on figure 2 To simplify the example, some operations (such as software product uninstall) in the figure are not shown.

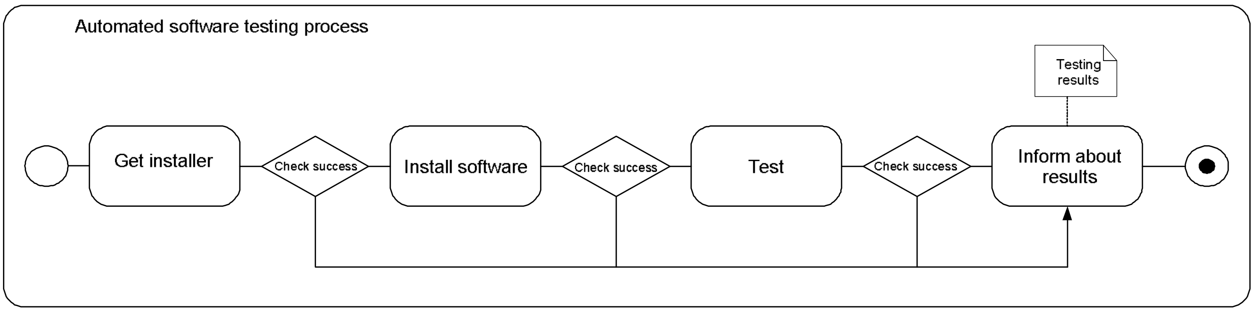

Fig. 2. Graphical representation of automated software testing business process

Model specification is stored in xml format. This format allows store both data and metadata required for model processing into a single file. Below is an example of automated software testing business process model specification.

```
<Model>
	<ModelMetadata>
		<GeneralInfo>
			<ModelId> id </ModelId>
			<ModelType> BusinessProcessModel </ModelType>
			<ModelName>OvernightAutomatedTestingModel</ModelName>
			<OntologyURI> www.acme.org/AutomatedTestingOntology</OntologyURI>
			<ModelRepositoryURI>www.acme.org/ModelRepository</ModelRepositoryURI>
		</GeneralInfo>
		<ActivationInfo>
			<Condition>
```

```
                <ConditionBd> Was not active during</ConditionBd>
                <ConditionParameter>15 min<ConditionParameter>
            </Condition>
            <StartState>InstallerChecking</StartState>
        </ActivationInfo>
    </ModelMetadata>
    <ModelBody>
        <Operations>
            <Operation>
                <OperationName>CheckInstallerAvailabilty<OperationName>
                <OperationModel>ModelId1<OperationModel>
            </Operation>
            <Operation>
                <OperationName>GetInstaller<OperationName>
                <OperationModel>ModelI2d<OperationModel>
            </Operation>
            <Operation>
                <OperationName>Install<OperationName>
                <OperationModel>ModelId3<OperationModel>
            </Operation>
            <Operation>
                <OperationName>Test<OperationName>
                <OperationModel>ModelId4<OperationModel>
            </Operation>
            <Operation>
                <OperationName>UnInstall<OperationName>
                <OperationModel>ModelId5<OperationModel>
            </Operation>
            <Operation>
                <OperationName>FinishAndClean<OperationName>
                <OperationModel>ModelId5<OperationModel>
            </Operation>
            <Operation>
                <OperationName>InformByEmail<OperationName>
                <OperationModel>ModelId6<OperationModel>
            </Operation>
        </Operations>
        <Decision>
            <DecisionCallType>BetweenOperations</DecisionCallType>
            <DecisionModel>DecisionModelId</DecisionModel>
        </Decision>
        <ProcessFlow>
            <Signature>
                <Condition>
                    <IB_Entity Type="Test_Status">ReadyForTesting<IB_Entity>
                </Condition>
                <Execute>
                    <SetIB_InstanceValue
Type="Test_Status">CheckingInstallerAvailability<SetIB_InstanceValue>
                    <ExecuteModel>CheckInstallerAvailabilty<ExecuteModel>
                </Execute>
            </Signature>
        ......
        </ProcessFlow>
    </ModelBody>
</Model>
```

Code describing the model consists of sections of metadata, activation, and body of the model. In metadata section are specified the name and type of the model, a reference to an ontology and models repository. Activation section defines how the model is executed. In our example the model is activated after a period of inactivity of 15 minutes. Model body section contains subsections of operations, decision making and flow control. In operations subsection are specified the operations of current model and references to the relevant models, describing operations. In decision making subsection is specified how to activate models of decision making (at the beginning and between operations), and links to decision making model. Decision making model analyzes the state of domain by information base and determines the status of the testing process. Then, data flow control subsection is used to activate the next operation.

BUSINESS OPERATION MODEL

Building a model of business operations allows the expert to identify and analyze in detail all relevant to the operation entities, relations, constraints, calls to external services or other models.

The general model of business operation includes the following elements:

• entities from ontology relevant to the operation

• relations relevant to the operation

• initial state, presented as a set of conditions specified using attribute values of relevant entities and relations

• final state of the operation execution. This is a list which includes both states of successful completion of the operation as well as unsuccessful completion states.

• specify the mechanism of the transition between the initial and final states, such as a set of operations (commands) to external service

• determining the metrics of success of the operation and include a reference to external model which implements the evaluation of those metrics.

The model of business operations can be executed either manually or automatically. In the case of manual execution model:

• before actually starting, it automatically checks all prerequisites for its execution and informs a person assigned to perform operation;

• helps the person who performs an operation to identify and obtain relevant information on all relevant objects and their properties needed;

• imposes restrictions on the process of the operation execution as defined by law and corporate rules and regulations.

• after operation completion system evaluates the effectiveness of its execution and updates the information base with results.

For example, when hiring a person's a model helps HR employee to check information on education, work experience, qualifications, and other conditions needed to fill a specified position, verifies the authenticity of the documents submitted reading data from the database of organizations that have issued documents.

In case, if business operation model is executed automatically, after checking the prerequisites, it initializes operation execution. This execution is performed by calling external services. After the operation completion efficiency of execution is evaluated. As in the case of manual operation the result of the business operation is stored in the information base.

Graphical representation of the business operation model should allow the domain expert to capture and understand all relevant aspects of the operation (Fig.3). Therefore, it should have a small number of items (4-7).

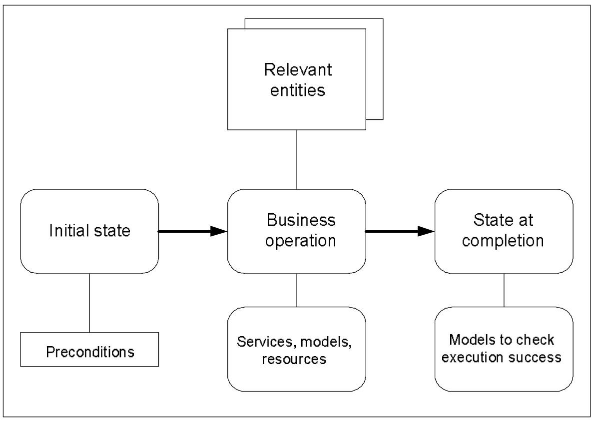

Fig. 3. Graphic representation of business operation model

This representation includes:

• a list of entities and their attributes relevant to the operation,

• preconditions of an operation

• a list of services, resources and models used

• the description of the state of the domain after the operation

• models for evaluating business operation success and execution performance

Textual business operation model is represented in xml-format. An example of textual representation of the generic business operation model is shown below.

```
<Model>
    <ModelMetadata>
        <GeneralInfo>
            <ModelId> id </ModelId>
            <ModelType> BusinessOperationModel</ModelType>
            <ModelName>BO-GenericModel</ModelName>
            <OntologyURI> www.acme.org/AutomatedTestingOntology</OntologyURI>
    <ModelRepositoryURI>www.acme.org/ModelRepository</ModelRepositoryURI>
        </GeneralInfo>
        <ActivationInfo>
            <Condition>     OnCommand</Condition>
            <StartState>CheckPreconditions</StartState>
        </ActivationInfo>
        <Preconditions>
            <Precondition>
            ......
            </Precondition>
        </Preconditions>
    </ModelMetadata>
    <ModelBody>
        <Operation>
            <OperationName>OpName<OperationName>
            <OperationDescription></OperationDescription>
        </Operation>
        <Entities>
            <Entity>
                <Id>EntityId</Id>
                <Attributes>
                    <AttributeId>Id</AttributeId>
                    .......
                <Attributes>
            </Entity>
        </Entities>
        <Resources>
            .....
        </Resources>
        <Services>
            .....
        <Services>
    </ModelBody>
</Model>
```

The preconditions for model execution are shown in metadata model section. In model body section are specified relevant entities and their attributes, references to resources and services.

DECISION MODEL

The purpose of decision model is to identify well known situations in business environment and trigger necessary actions. Decision models are placed in specified places of business process model, typically on business operation completion or when some business event occurs.

Decision model contains entities from ontology relevant for decision making for some known situation. Important components of decision model are situation signature and action specification.

Situation signature specifies a set of conditions which hold in this situation. Mathematically each condition is represented by first-order predicate. For example, “A completed testing exists” situation is specified by:

$$\exists t(type(t) = "Testing" \text{ and } (t.status = "Completed"))$$

An action specification defines the execution of a specific action. For example, such action types are typical:

- Initialization and execution of business-operation model
- External service calling using a specified command
- Complete model execution

Similarly to business operation model execution, decision model can be executed manually or automatically. If model is executed manually, system selects from information base all facts relevant to situation and proposes recommended solutions, with possible parametrical adjustments.

If decision model is executed automatically, system verifies situation signature and initiates corresponding actions if some situation was identified.

SOFTWARE MODELLING SYSTEM PROTOTYPE

In order to study the feasibility and practical usefulness of model-based intelligent system design approach, a software prototype for modelling system was developed. This prototype consists of four components, united in a common application: Ontology Editor, Fact Editor, Model Editor and Modeller.

Accordingly, the main window of modelling environment has tabs for ontology, facts, model editors and modeller (Fig.4). Also are shown the basic constructs used in systems, such as Attribute types and Roles.

Ontological engineer uses the ontology editor for the creation and modification of ontology. This software component has several functions, such as export and import of ontology; creation, modification and deleting of ontology entities; implementation of inheritance; creation, modification and removal of relations; creation, modification and removal of constraints.

Ontologies and other data are stored in xml-format. The system supports exporting and importing ontology and all related data sets (facts, models) as separate modelling contexts.

Fact editor is designed to create and modify facts. Every fact is an instance of some entity in ontology. User creates facts and also specifies their attribute's values. The system checks the constraints associated with the attributes on process of fact validation. Only the fact that was validated successfully will be accepted by system.

Model Editor is a central component of modelling environment. It allows creating, modifying and deleting models, model metadata and constraints, operations. Figure 5 shows an example of simple classification model. The roles ClassifiedObjectRole and Classification are linked by relation Classify. In the process of model execution not only fact itself is used, but also some objects in this fact's context (for example fact's attribute values).

Component Modeller allows to create fact-models based on specified models, initialize them by facts from information base, execute operations in model.

Figure 6 shows an example of simple interval classification problem solving using the modelling environment.

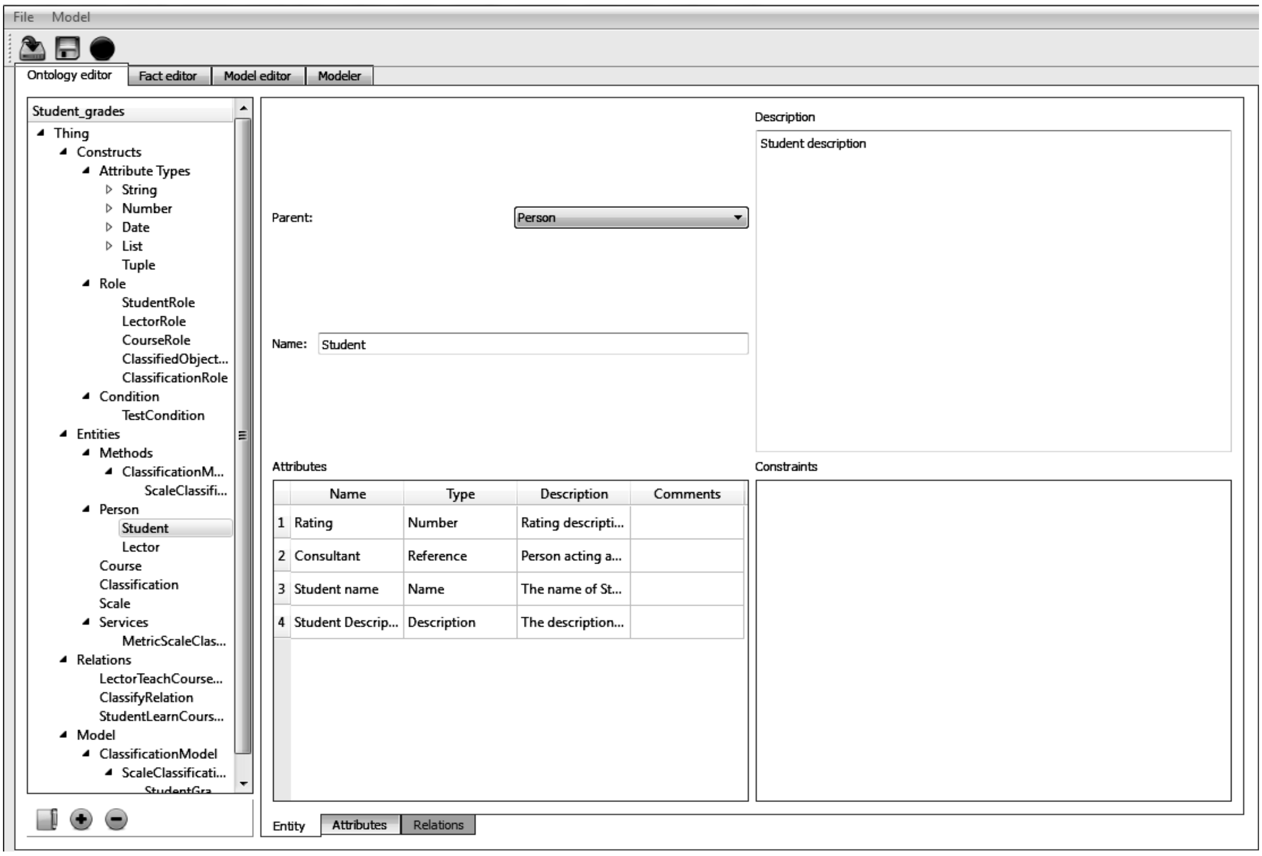

Fig. 4. Main window of modelling environment with ontology editor

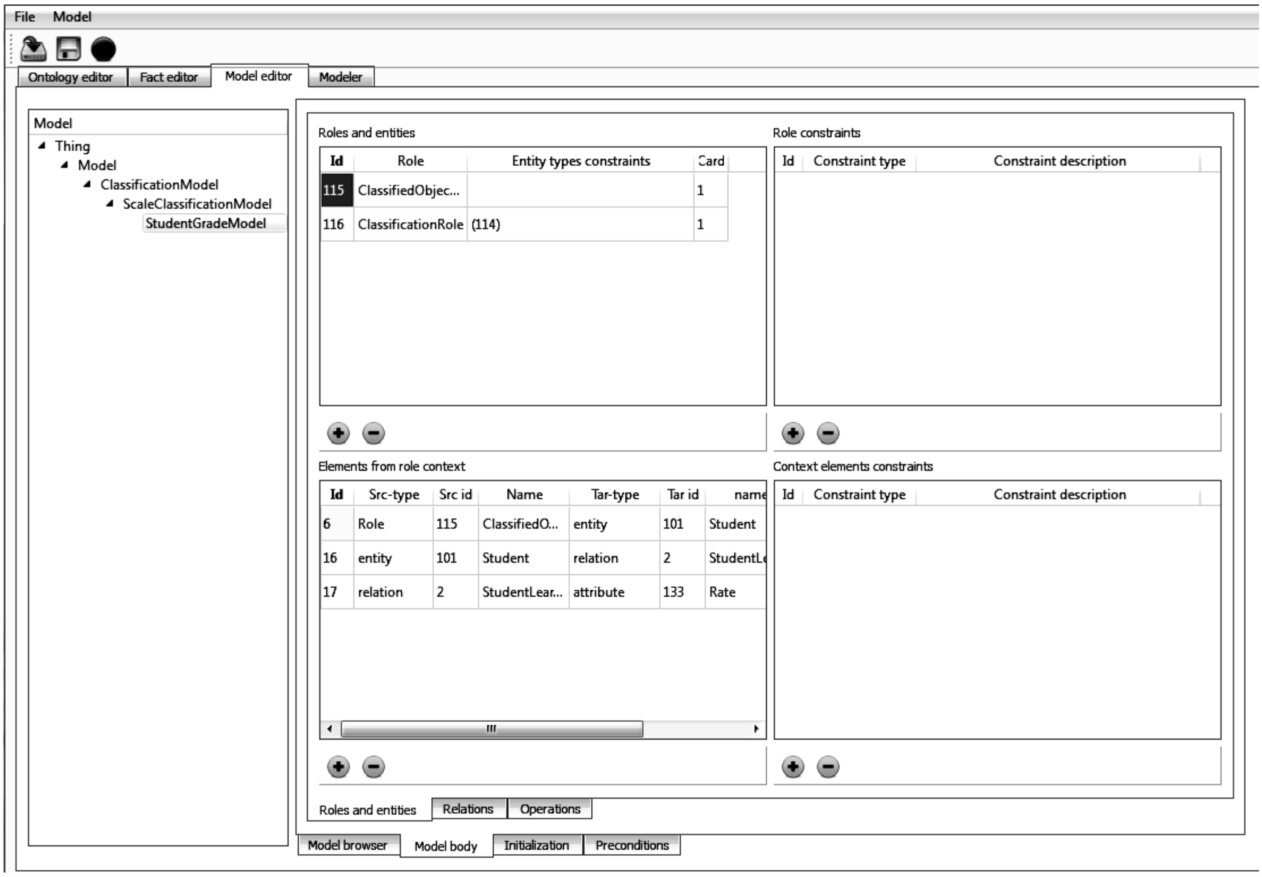

Fig. 5. Model editor

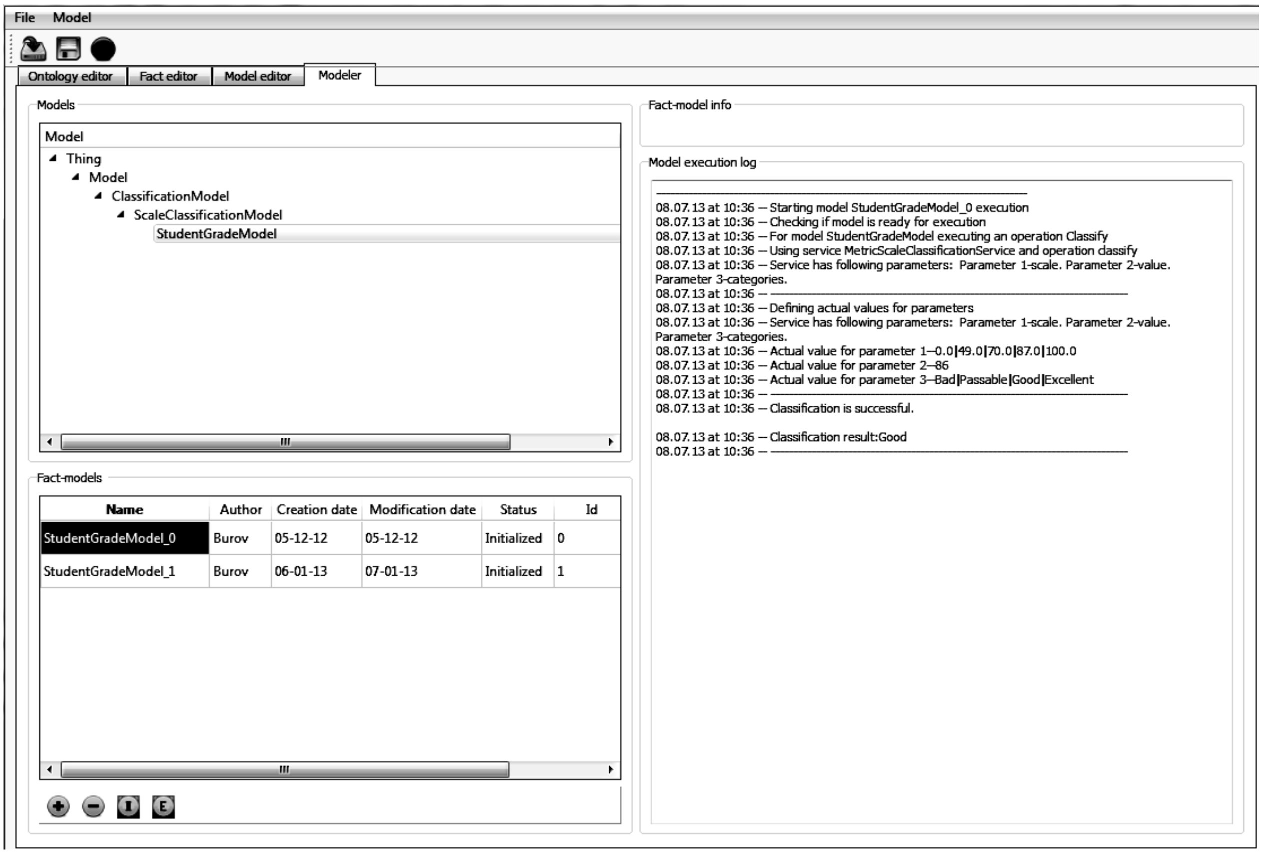

Fig. 6. An example of classification problem resolution

Developed software prototype modelling environment provide necessary flexibility for model modification. For example, the change of classification scale requires only attribute update or creating and using another fact of Scale type. Likewise, if method of classification is changed, then only attribute Method value of Classify relation should be updated.

CONCLUSIONS

Intelligent enterprise paradigm is largely used in order to improve the efficiency and flexibility of business processes. Knowledge-based business process modelling, using ontological tasks model, allows to create common enterprise-wide system of concepts and relations in form of ontology. Typical operations flows, decision patterns are captured in form of ontological models. Knowledge specified in form of ontological models is reused. Corporate standards and regulations are also enforced using corresponding models.

Developed business process modelling ontology along with generic business operation and decision models allows to build complex business process models.

As future work we intend to theoretically research the system created by complex multiple interacting models executed simultaneously, investigate the usefulness of proposed approach during real business processes implementation.

REFERENCES

1. **Williams S. 1967**. Business Process Modeling Improves Administrative Control. Automation, December, 44–50 (USA).
2. **Davenport T. H. and Short J. E. 1990.** The New Industrial Engineering: Information Technology and Business Process Redesign, Sloan Manage. Rev., vol. 31, no. 4, 11–27. (USA)
3. **Weske M. 2007**. Business Process Management: Concepts, Languages, Architectures. Springer-Verlag. (Germany)
4. CMMI Architecture Team. **2007**. Introduction to the Architecture of the CMMI ® Framework (USA).
5. **Mili H. and Jaoude G. B. and Jaoude G. Bou and all. 2010**. Business process modeling languages: Sorting through the alphabet soup. ACM Comput. Surv., vol. 43, no. 1, 1–56 (USA).
6. **Graham B. B. 2004**. Detail process charting: speaking the language of process. John Wiley& sons, 209 (USA).
7. **Kuzmin O. Ye. and Melnyk O. H. and Shpak N. O. and Mukan O. V. 2012** The concept of creation and use of the polycriterial diagnostics systems of enterprise activity, Econtechmod. An international quarterly journal on economics in technology, new technologies and modelling processes. Vol. I, No 4, 23-28 (Poland)
8. **Howson C. 2006**. Business Objects XI: the complete reference. McGraw-Hill, 650.
9. **Larson B. 2008**. Delivering Business Intelligence with Microsoft SQL Server 2008. McGraw-Hill Osborne Media (USA).
10. **Davis R. 2008**. ARIS Design Platform. Advanced Process Modelling and Administration. Springer-Verlag, 416, (Germany).
11. **Lehmann T. and Karcher A. 2008**, Decision support and situational awareness. Intell. Decis. Technol., vol. 2, no. 1, 21–31 (USA).
12. **Niu L. and Lu J. and Zhang G. 2009**. Cognition-Driven Decision Support for Business Intelligence: Models Springer, 237. (Germany).
13. **Hutchison D. and Mitchell J. C. 2007**. Active Conceptual Modeling of Learning, vol. 4512. Berlin, Heidelberg: Springer Berlin Heidelberg, 243 (Germany).
14. **Ranjan J. 2008**. Traditional Business Intelligence vis-a-vis real-time Business Intelligence. International Journal of Information and Communication Technology, vol. 1, no. 3/4. 298 (Germany).
15. **Quinn J. B. 2005**. The intelligent enterprise a new paradigm. Academy of Management Executive, vol. 19, no. 4. 109–121 (USA).
16. **Gruber T. R. 1993**. A translation approach to portable ontology specifications. Knowl. Acquis., vol. 5, no. 2, 199–220 (USA).
17. **Osterwalder A. 2004**. The Business Model Ontology-a proposition in a design science approach. Citeseer. (USA)
18. **Kang D. and Lee J. and Choi S. and Kim K. 2010**. An ontology-based Enterprise Architecture. Expert Systems with Applications, vol. 37, no. 2. 1456–1464 (USA).
19. **Mangione G. R. and Miranda S. and Paolozzi S. and Pierri A. and Ritrovato P. and S. Salerno. 2009** Ontology-Based System for Enterprise 2.0. Ninth Int. Conf. Intell. Syst. Des. Appl (USA).
20. **Lytvyn V. and Semotuyk O. and Moroz O. 2013**. Definition of the semantic metrics on the basis of thesaurus of subject area. Econtechmod. An international quarterly journal on economics in technology, new technologies and modelling processes. Vol. II, No 4, 47-52 (Poland)
21. **Michalewicz Z. and Michalewicz M. 2008**. Machine intelligence, adaptive business intelligence, and natural intelligence. IEEE Comput. Intell. Mag., vol. 3, no. 1 (USA).
22. **Airinei D.-A. and Dinu Berta 2012**. Semantic Business Intelligence - a New Generation of Business Intelligence. Infomatica Econ., vol. 16, no. 2, 72–80 (USA).
23. **Kokar B**. **2009**. Ontology-based situation awareness. Int. J. Inf. Fusion, no. 10(1), 83–98. (USA)
24. **Miah S. J. and Gammack J. and Kerr D. 2007**. Ontology Development for Context-Sensitive Decision Support. Third Int. Conf. Semant. Knowl. Grid (SKG 2007) (USA).
25. **Ikeda M. and Seta K. and Kakusho O. and Mizoguchi R. 1998**. Task ontology: Ontology for building conceptual problem solving models, Environment, 126–133 (USA).
26. **Raubal M. and Kuhn W. 2004**. Ontology-based task simulation, Spat. Cogn. Comput., vol. 4, no. 1, 15–37 (USA).
27. **Johnson P. and Johnson H. and Waddington R. and Shouls A. 1988**. Task-Related Knowledge Structures□: Analysis , Modelling and Application, Knowl. Creat. Diffus. Util., 35–62. (USA).
28. **Van Welie M. and Van Der Veer G. C. and Eliëns A. 1998.** An Ontology for Task World Models, Methods, vol. 98, 57–70 (USA).
29. **Taylor P. and Raubal M. and Kuhn W. 2004**. Ontology-Based Task Simulation, Spat. Cogn. Comput., vol. 4, no. 917247301, 15–37 (USA).
30. **Seta K. and Koyama K. and Hayashi Y. and Ikeda M. 2006**. Building ontologies for conceptual model management, WSEAS Trans. Inf. Sci. Appl., vol. 3, no. 3, 546–553 (USA).
31. **Olive A. 2007**. Conceptional modeling of information systems. Springer Berlin Heidelberg, 471 (Germany).