

# Integration of Business Modeling and IT Modeling

Girts KARNITIS<sup>a,1</sup>, Janis BICEVSKIS<sup>a</sup> and Jana CERINA-BERZINA<sup>b</sup>

<sup>a</sup>*University of Latvia, Raina 19, LV-1586 Riga, Latvia*

*Email: Girts.Karnitis@lu.lv, Janis.Bicevskis@lu.lv,*

<sup>b</sup>*Datorikas Instituts DIVI, A. Kalnina str. 2-7, LV-1050 Riga, Latvia*

*Email: jana.cerina\_berzina@di.lv*

**Abstract.** The paper covers our research in improving business and IT process modeling. Practical experience shows that universal modeling languages such as UML and BPMN are IT oriented and are difficult to accept by business people. Business models need to use such hard to formalize business terms as legal acts, informal activities descriptions and comments etc. We propose novel approach how to create or choose modeling tools according to model usage. We propose 4 usages for modeling: model to understand business processes, model for business process definition, model as information system specification and model that can be interpreted by computer. All these four models can be perceived as four highly interconnected parts of one single model. We propose to create unique domain specific language for each model. We analyze functions specific for each model type. Authors last six years practical experience for creating and practical introducing related business and workflow oriented information system models are analyzed.

**Keywords.** Business process modeling, domain specific languages, modeling

## Introduction

Business process modeling and information system specification have number of problems irrespective of wide research in a field of Model Driven Software Development (MDS) [1] and Model Driven Architecture (MDA) [2]:

- Universal modeling languages UML and BPMN [3, 4] do not provide language easily understandable by both – IT specialists and business people. IT specialists propose to use UML than can describe operation of IS objects. In such case process part outside the computer stays weakly specified. Therefore business people cannot adequately describe their requirements.
- Model syntax prevails over model's meaning – semantic. Universal modeling languages concentrate on graphical symbols usage laws and do not explain symbols semantic meaning. It's even impossible, because in each particular case semantic may differ. As a result process semantic is hard to put into the models and to read it and models allow different interpretations.

---

<sup>1</sup> Corresponding Author. As. professor, University of Latvia, Raina 19, LV-1586 Riga, Latvia; Email: Girts.Karnitis@lu.lv.

As a result software requirements specifications and job descriptions are created in form of a written document containing illustrative graphs and schemes.

Our novelty approach is to concentrate to model semantic according to specific model usage. We have identified 4 important model usages – informal business description, precise business model that can be used as work instructions for people, business process model that serves as requirement specification for IS development, business process model that is used by IS to operate. In details – we propose to add semantic to each symbol used in diagrams according its usage. Widely known modeling languages, such as UML activity diagrams or BPMN can be adopted to acquire abovementioned goals. But because of specific requirements (link with information in existing IS, consistency checks) we decided to use specific DSL.

Universal modeling languages specify only syntactic structure of diagram and do not specify semantic meaning. We propose assign to activities/tasks not only unique names but also attach informal descriptions defining semantic meaning of each activity/task. In this case diagrams syntax can conform to UML Activity Diagrams or BPMN or domain specific language (DSL), but semantically they are supplemented with these informal description and shows essence of modeled system. It brings us to specific DSL [5] that allows us to understand models in context of specific system diagrams made with universal modeling tool.

This technique is used in number of projects in Latvia. Result shows that proposed approach allows minimize problems with connections between business people and IT specialists. In this paper we explain proposed approach with small process and 4 usage scenarios and 4 different types of models of this process.

The structure of the paper is as follows. Chapter 1 contains four usage scenarios and four different DSLs for these usage scenarios. Chapter 2 contains analysis of properties of proposed DSLs. Chapter 3 contains our experience with usage of these DSLs.

## 1. Types of Models

During analysis and modeling number of organizations we have developed 4 different model usage scenarios. In this chapter we analyze these model usages and show how to add semantics to the models. As a result we get 4 DSLs that allows to model wide spectrum of workflow oriented systems:

1. Business process description – due to graphical illustration of the processes the model contributes to better understanding of the nature of business processes [6].
2. Business model determines definition of a process – formalized language determines process steps, implementation and reports, communicated among process executors, thus creating precise descriptions of functions carried out by the employees. Moreover, process implementation steps are closely linked to the organizational structure – process executors and their activities under respective normative acts. Clearly, IT specialists proposed UML only partly meets the demands of above-mentioned non-formalized descriptions.
3. Business model serves as requirements specification for the IS development – description of business process steps liable to IS support are included in the model as well as all process implementation rules and algorithms.
4. Business process model creates the base of IS operations' description [7]. Such situation is common in recordkeeping and event-oriented systems, where work-

flow is described by graphical diagrams in specialized Business Process Modeling Language (BPML), but software is developed by generating Business Process Execution Language (BPEL) commands from the model in BPML. Thus, we consider the work-flow description language BPML to be domain specific language.

### 1.1. Model as the Process Description

Usually, the initial organizational processes are missing description or descriptions are anchored in legislative and normative acts and instructions. Frequently, these instructions are incomplete or even contradictory and incomprehensible. Process models are created to improve this situation. First step is to create a model that describes process and serves as a descriptive material for organizational process. Users of these models are usually top business managers valuing mainly comprehensibility and simplicity of the model. Here the common principle requires intuitive guess of model's syntax or a minimum training. Main requirement of such models – users without deepest understanding of modeling language must intuitively understand process described in the model. Main components of such model are graphical diagrams illustrating business process activities and their implementation sequence, as well as informal textual descriptions of activities attached to diagrams. Usually, the informal descriptions include references to normative acts, document templates and similar documents. Graphical diagrams and informal descriptions include also other related information such as information about process executor, deadlines and necessary technical resources.

As an example for the rest of the paper we will use registration of enterprises to allow them to perform specific kind of entrepreneurship. A law in Latvia as in many countries stipulates that to carry out some specific kind of entrepreneurship, for example to provide communication services, special license must be received. Figure 1 shows the highest level model for such process. Enterprise must address application to responsible state institution. In this case responsible institution is Public Utilities Commission (PUC). PUC considers application in 15 days and issues license and registers enterprise or rejects application in a case if some requirements are not fulfilled.

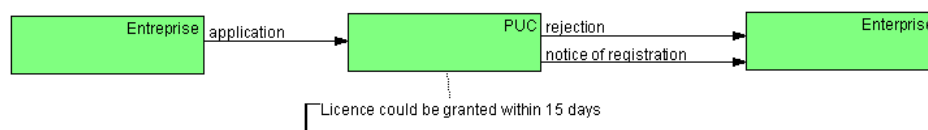
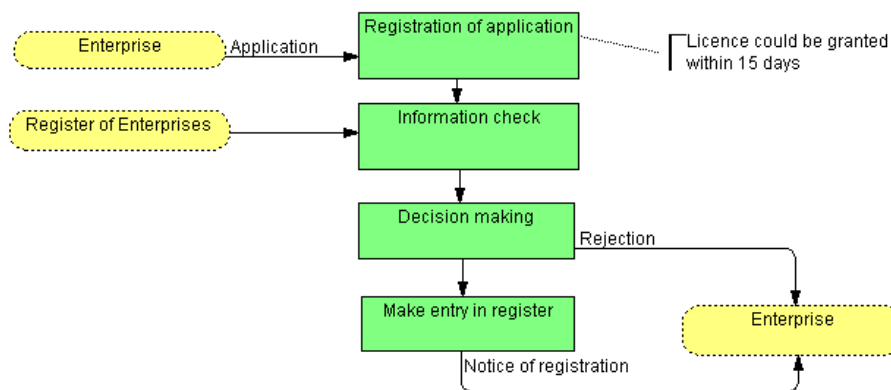


Figure 1. Enterprise registration process model

This simple picture can be formatted according to syntax of different modeling language. Still the meaning of the process can be understood by reading the description in precious paragraph. It means that main goal of modeling language is to bind process semantics with graphical picture. Frequently semantic is described via informal descriptions and normative acts that is not the part of the model itself. Rather process description and normative acts are stored into separate files, different web sites or paper documents. It is hard to use models without attaching these descriptions.

Figure 2 pictures the same process from PUC viewpoint. In this case registration process is divided into 4 consecutive steps – registration of application, enterprise information check, decision making weather to issue license or not, and license registration in register in case license is issued. This model is not in great interest for enterprises, but can be valuable for PUC employees to carry out their jobs. Similarly to previous example, this model is hard to understand without additional information (semantics explanations) how to carry out each step.



**Figure 2.** Enterprise registration process model from PUC viewpoint

The most important - these models are easily comprehensible and well presented, since their main objective is to provide good insight and receive management's approval for publication. However, there is also a main deficiency – these models are rather too abstract and fail to be completely consistent. These models usually have low formalization level. Actually, any graphical tool, such as Microsoft Visio, that provides unlimited choice of symbols, layouts and visualization possibilities, would be suitable for drawing a model of such sophistication level.

Representation of organizational processes by informal schemes, operational descriptions and diagrams is a common modeling approach anchored in process penetration and caused by requirements ISO quality management standards. These models, which are developed for solely the purposes of quality management, contain a certain degree of inaccuracy in order to use them for development of information system operations or monitoring enterprise performance indicators. As a result, the ultimate goal of quality management cannot be met, since there is a lack of objective performance assessment and informal schemes of processes can be used only by limited number of users.

IT specialists can use such model for better understanding of processes and industrial specifics during IS development initial phase. Informal nature of the model constrains its use for IS development and also cannot be used as precise official instructions.

### 1.2. Model for the Definition of Process

The next step of process modeling development would be to develop business process models that define business processes and provide a precise instruction for process execution [1]. This approach sets significantly higher and sophisticated requirements

for model's formalization, precision and consistency than in the case of models developed solely for the purpose of process understanding. These models should have a considerable degree of precision for employees to be able to interpret the model information precisely and execute the process steps correctly. However, the level of precision and formalization sufficient for employee's interpretation might not be sufficient for the implementation of such model in information system or for the automated execution of the model by the information system.

In our example, the executable model contains additional information about the process steps, activities carried out in each, including screenshots of IS forms and reports where information must be entered and can be read during this step. Technically this information can be seen with the corresponding model editor tool that would allow attract the appropriate screen forms or documents to each process step. Criteria for the models in this group - the user has access to all information necessary to process each step in real life.

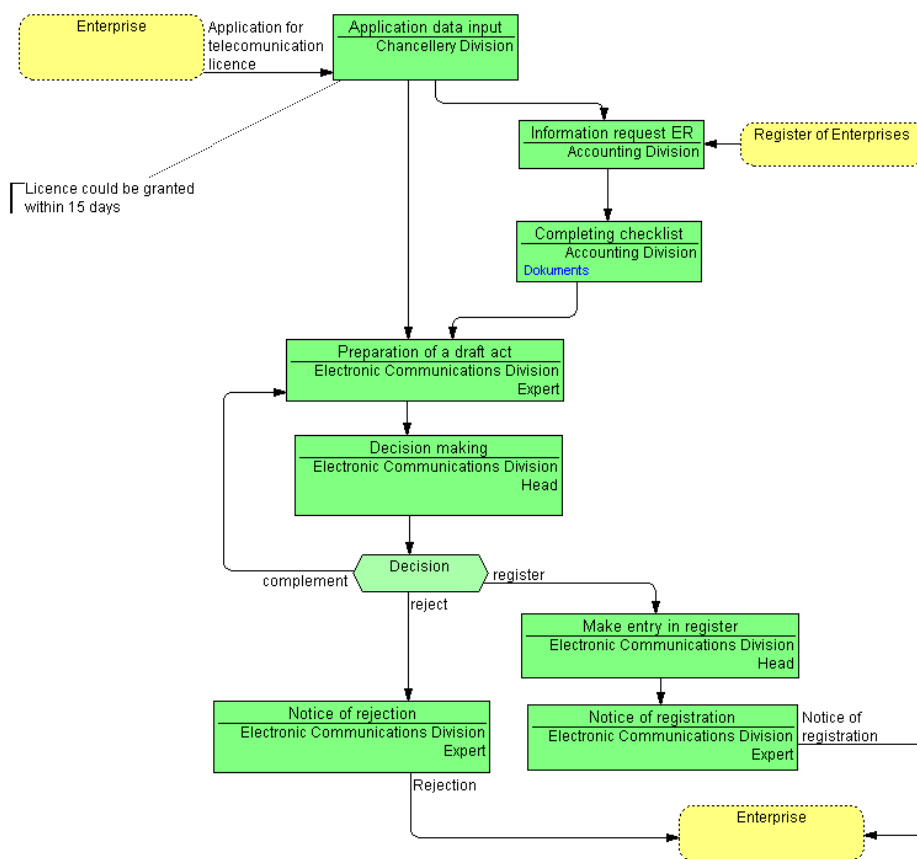


Figure 3. Executable model of enterprise registration

Models that are used to define processes contain graphical diagrams and informal descriptions as well as references (links) to external documents (precise references to paragraphs in normative acts and document templates), references to functions of external information systems (screen forms) and user instruction on filling these forms

(Figure 4), references to objects in external information system (e.g., process steps executor in organizational structure). Model for process definition cannot bear any inconsistency, therefore consistency verification is crucial. Conditions of consistency check-ups should be defined in the same modeling language and check-up should be performed by the modeling tool. It should be emphasized that consistency conditions are domain specific, which means – these conditions are specific to organizational and business requirements of a particular organization.

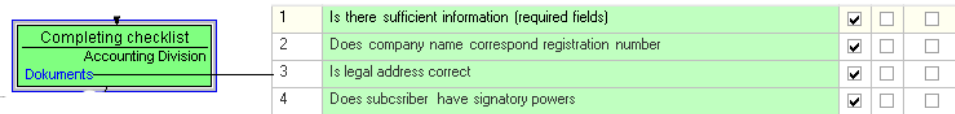


Figure 4. Activity with external link – checklist form

Main users of model are employees, which use the model as an instruction for everyday execution of business processes. Above mentioned model does not contain precise data structures, therefore it cannot be automatically executed and only partially contains information necessary for IS requirements specification.

### 1.3. Model as IS Requirements Specification

Model described in the previous sections were used to define business people view of processes taking place in organization. In abovementioned model business processes are described maximally precise from business point of view so that employees can accomplish them. Business people and also IT people are able to understand language used in such models. However, these definitions and descriptions are not sufficient in order to describe operations of information system – these models must be complemented with data structures and other information required to develop IS. Such models we will call IS diagrams.

Usually IS diagrams are created by the information system analysts. IS diagrams contain descriptions of software screen forms, reports and automated tasks – algorithms. IS diagrams describe conceptual data level, which can be detailed further to the level of physical data model. IS diagrams continue detailing the business process steps from the point where business process diagrams have reached their constraints. For example we take process step “Make entry in register” from previous model in Figure 3 and detail it. We specify this step as two steps and add IS menu items needed to perform this step (Figure 5).

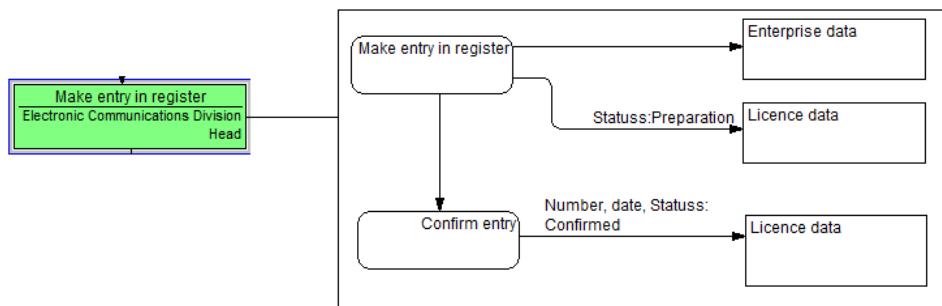


Figure 5. Refinement of “Make entry in register” process step

IS diagrams have several ways of applicability:

- IS diagrams serve as a basis for development of software functional requirements containing data input, data output, data processing and data verification.
- IS diagrams determine, which business process fragments should be implemented in information system and became fully automated and which should remain manual even after information system is introduced. For example, if a business process foresees verification of a signature on submitted document then the employee will perform the verification manually. IS can only verify input values and information consistency.
- IS diagrams allows to make data usability analysis. For, example, if we would like to change processing logic of data object we can create view where we can see all IS activities that use or influence these data. A different view can be created in order to monitor status change of an object in order to indicate activities related to object status and its changes.
- IS diagrams can serve as a basis for creation of test cases for software testing. Use of both, business process model and IS diagrams, can easily solve the problem of software testing completeness - business process model can be used as testing model.

In summary, if business requirements are gradually and successfully developed into IS requirements, organization obtains a very precise description of organization's information system and IT specialists obtain functional requirements for the system implementation. By using and maintaining business process and IS models in a long term organization obtains significant information on impacts and costs of changing or adding new business functions to organization.

#### *1.4. Model as IS Component*

Current trends in information system technological development can be characterized by less technical coding, more precise specification resulting in software development "without programming". Modeling is the integral part of this process.

IT trend, where modeling builds the foundation to systems operations, is represented in workflow-oriented systems. A workflow indicating process steps - activities is developed according to organization's business process. In a workflow significant role is played by executor – user, which has an assigned role in organizational structure. Process steps perform data processing of involved object. Work-flow provides possibility to store data throughout the object's life-cycle. Such concepts and terms as task deadlines, actual status of object, possibility to create documents automatically during workflow execution are very significant in work-flow oriented systems.

There are two specific tasks for models, first, it should be rather simple and comprehensible for employee to understand workflow, second, it should be sophisticated enough in order to contain all information significant for operation of information system. In principal, these are two different views of the same workflow diagram.

Which would be the most appropriate language for creation of workflow description? One possibility is use UML or BPMN. Unfortunately, in the UML and BPMN models because of their universality is difficult to incorporate domain

semantics. In addition, the UML and BPMN are overloaded with many technical features, which are needed for the development of IS in specific cases, but often are redundant for a particular system and is difficult to understand by the business people. Frequently acceptable option is to use a DSL, which allows you to build models for both – business needs and for the IS implementation.

Authors of this article have developed domain specific language BILINGVA [7], with the purpose to describe workflows. Diagrams illustrate object statuses, process steps (events) and sequences of process steps. The language provides possibility to describe also decisions – branching points in processes, define deadlines and execute processes parallel. After a model is created the information stored in diagrams is transported to the information system repository and system's configuration tables. By operating information system its engine interprets information obtained from diagrams automatically, thus ensuring system functionality defined in the model.

Figure 5 contains executable enterprise registration process. Unlike diagram from previous chapter, all activities in this model can be executed by IS. The model does not contain objects that are not covered by IS such as “Enterprise” and “Enterprises register”. The model also does not contain manual operation such as document review. In BILINGVA syntax arrows are activity/task but boxes are object states. Parallel execution also can be shown. After application, registration, enforcement activity has been transferred to the Accounting Division, operating as sub process in parallel with the main process, and shall be executed by an expert of Electronic Division.

Time limit of the process is represented as two events – “begin time control” and “test time control”. In event “begin time control” event “Time to register – 15 days” is stored. IS form shown to user in a case time control rule is infringed is implemented into model interpreter and has not specified in the model. This helps to create user-friendly interface for execution of concrete activities. Such user friendliness is hard to achieve in a case if universal solutions are used.

Diagram in Figure 6 is a user-oriented view, where the notation proposed in [7] is used. To ensure correct work of system process modeler completes attribute set for each process step that is necessary for correct operation of system. For example each step is characterized by the following parameters – does step can make document, if so, than document template must be specified, does some document can be uploaded to system during the step, does specific checklist is fulfilled during the step, step executors roles – manager, responsible employee or anyone. Does responsible employee can be set in this step? In which conditions step can be executed? Step status attribute set describing which departments see this status, status number - to sort statuses in user interface in logical order, status group. Time limit is additionally characterized by following attributes: is time limit in calendar days or working days, is time limit automatically or manually counted, how many days before deadline system issues warning about deadline, which departments are bound to the deadline.

Returning to applicability aspect, a detailed workflow diagram provides framework for system's operations, a special view serves as a user's manual, which can be complemented by explanatory documents referenced to certain steps.

The future perspective includes task – to implement graphical representation of process step execution in business process diagram visible to system user. For example, by providing simplified process model on organization's homepage the customers could track and follow execution status of provided service.



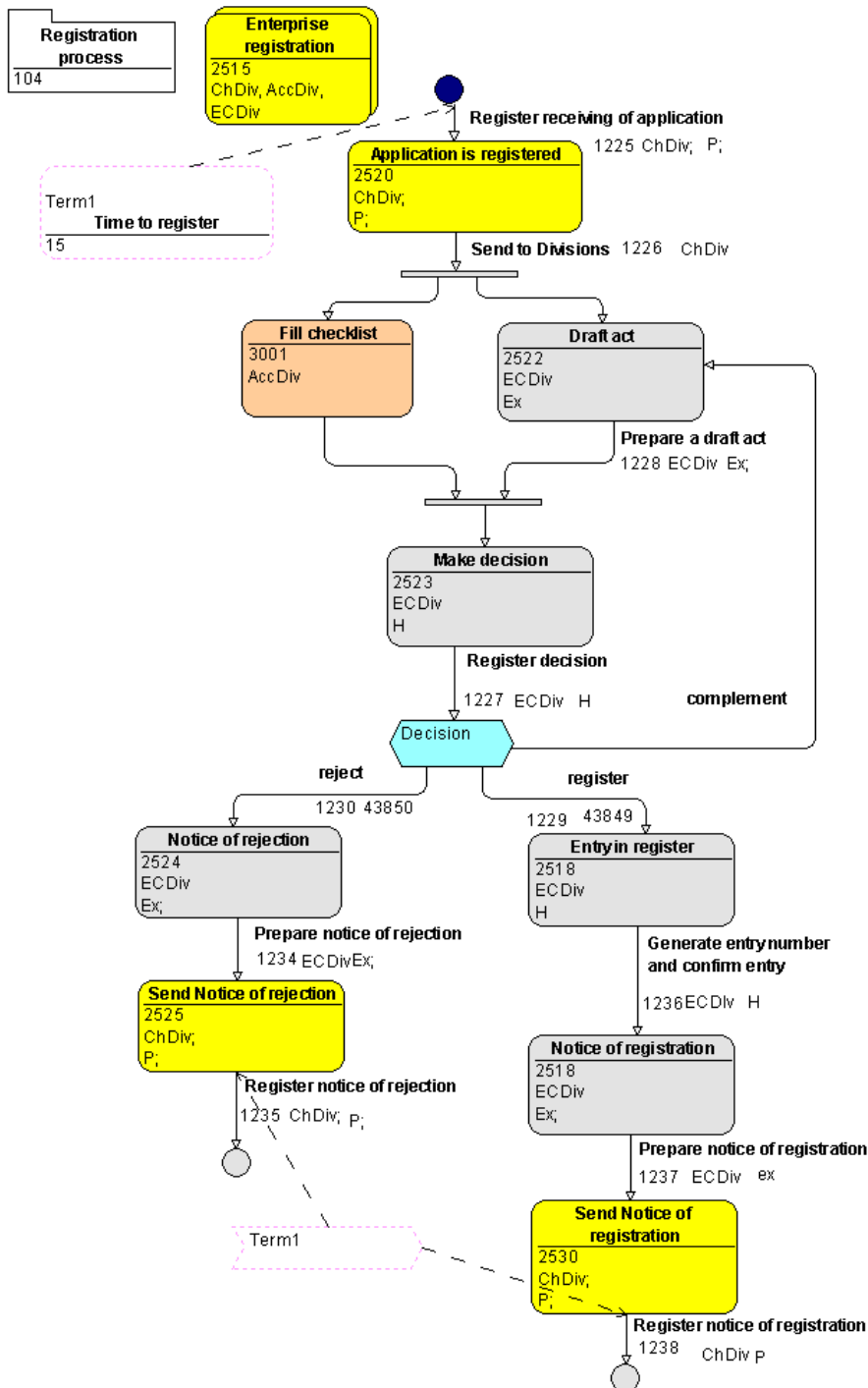


Figure 6. Executable process model

## 2. Components of Model

### 2.1. Models Attributes

Each of four models described above have their own applications and users. Consequently, these models differ in the semantics of elements and attributes of the elements. This fact is presented in Table 1 where some attributes of the process steps of various models are given.

**Table 1.** Attributes of process steps

	Purpose of process	Definition of process	IS requirements specification	IS component
Semantic precision	Very low formalization	high formalization	very high formalization	100% formalization
Syntax precision	Very low formalization	high formalization	very high formalization	100% formalization
Step name	Yes	Yes	Yes	Yes
Step description	Might be	Yes	Yes	Only for human. IS does not need.
Performer	Might be	Yes	Yes	Yes
Time limits	Might be as informal comments	Yes	Yes	Yes, in fully formal form

Documents bound to the model can differ according to the model type:

- Model for purpose of process – process regulating documents are attached as external files. Usually documents are attached to the whole process not to specific process step.
- Model for definition of process –regulatory documents preferable attach to specific process step. IS screenshots, document templates, user manuals and job descriptions are attached to the process steps.
- Model for IS requirements specification –specific chapter regulating process step preferable attach to process step. IS forms that must be fulfilled in this step, description of used data (data model), consistency checks, error messages and other information necessary for requirements specification are attached to the process step.
- Model as IS component – document templates used in this step to generate documents are attached to the process step.

Other attributes can be added to the model according to the domain specific.

### 2.2. Model Views

Model describes the operation of a single object - the organization, its processes and their implementation. Our approach offers to build a single model with common information and add different views to model for different users and different applications. Various pieces of information from the model (different views of the model) are necessary for business representatives, customers, quality service, IT

professionals, and so on. The above mentioned applications show how wide are usage of models. We will mention number of other applications in this chapter, which requires their own solutions.

One way to use model is different views for different applications. Example of view is series of activities performed to accomplish specific service (for example, registration of enterprises to allow them to perform specific kind of entrepreneurship). Another example of a view - the process steps that specific position is interested in - for example, Manager view of the processes, which include not only the steps Manager performs, but also actions that are performed before and after him. Another example of view - the quality system view of the organization's operations, which includes the entire range of activities, but rather superficial, omitting a number of components.

We may say that Manager is interested in some of the process execution to the smallest detail, but in some other process execution is actually not interested at all. By contrast, the quality department is interested in all the processes, but to the highest top level without going into detail.

It is unacceptable that implementation of the above mentioned views would be generated in a separate model, unrelated to the common organizational model. In order to maintain consistency each such view should be implemented as a subset of the overall model. Currently, such views can be created manually and consistency checks can be built into tool. At present, an unresolved problem is the view definition and automated creation from the integrated model.

Moreover, the model can be applied in a further way – service diagrams can be developed based on functionality model. Service diagrams illustrate certain process steps to be carried out in order to provide a particular service to customers. For example, a help-desk inspector at one stop agency provides services to a number customer with various demands. It is impossible for inspector to precisely know by heart every process step of a number of services applicable for each customer case. Therefore the inspector uses process model that includes precise instruction for each service case.

A different reporting possibility is a further way of model applicability, for example, reference of process steps to certain legislative acts. In case of amendments of legislative acts these references easily indicate process steps that should be changed. UML and partly also BPMN can be used as a language to meet the requirements of such models for these languages provide precise definition for process execution. However, missing possibility of adding domain specific information to model makes use of these languages problematic, for example, would it be possible to easily added references to paragraphs of normative acts if a process step is defined in UML or BPMN? Moreover, since the consistencies usually are domain specific, the use of universal modeling languages makes keeping of model consistencies problematic. The elimination of this problem would require development of specific tools in order to ensure consistency in model created by universal languages.

### **3. Real Life Experience**

Authors has several years of experience in the use of DSL in seven organizations in all four analyzed above applications. Usually each case consisted of three steps. The first step was DSL definition, typically through business process modeling together with the company's business professionals. Usually new DSL was based on some already

existing DSL and a new definition seemed to add extra attributes and attach domain-specific input / output documents. DSL definition virtually at the same time meant the definition of modeling tool. That was made possible by using a DSL tool building platform [8, 9, 10]. In the third step, the business model was developed using newly created DSL and modeling tool. This model was then transferred to a business-specific application. Accumulated experience has shown this approach strength:

- Each new DSL definition reveals new requirements for organizational model, such as model's elements linkage to the sector-specific document flow. Identified requirements can be incorporated into DSL and model.
- General purpose modeling language cannot describe all the sector specific requirements, for example, it's hard to access to non-formal process descriptions stored in the enterprise IS. Similarly, if universal modeling language is used then it is hard to link document samples available in company IS or website to the process descriptions.
- It was especially surprising to discover very positive attitude of users towards graphical specifications which users preferred to use as information system exploitation manual for it contained more precise information in a more comprehensible way, thus replacing old-fashioned user manuals.

Modeling technology is a new type of activity for business people. It takes time and the gradual acquisition of modeling techniques. At first, business people are able to create informal models (informal process description models) and only after some time they are able to move to more formal models (formal models of process definition). In our case adoption of modeling technique and models in different organizations took 2-4 years.

#### **4. Conclusions and Proposals**

The experience of DSL's use allows make such main conclusions. Strength of DSL:

- DSL allows create single unified model within organization for all four different usage scenarios mentioned in this paper and allows different views to the model.
- Users can easily understand meaning of models and use models if business process semantic is bind to model objects. Syntax specific usually does no matter so much.
- DSL building is one of easiest way to bind semantic of specific domain to model.
- DSL definition and modeling tool definition platform plays one of mayor role for DSL usage. It is practically impossible to implement in real life methodology we describe without such platform. Platform we use [8,9,10] ensures DSL definition and modeling tool definition in relatively small time frame (approximately 3 month).
- Survey shows that end users definitely prefer graphical diagrams instead of traditional text documents. Experience shows that one graphical diagram can contain the same amount of information as 10 pages of textual instructions.
- Proposed process modeling methodology is examined in workflow type systems. For other types of systems the situation may differ.

Recognizing the benefits we also must admit problems arising from the usage of DSL:

- Definition of DSL and development of modeling tools requires involving high qualification specialists.
- Enterprise specific DSL development and business process definition is individual as enterprise specific IS development. Clients must support not broadly recognized DSL technologies, in order to develop and maintain DSL, modeling tool and company models.
- DSL defined for needs of one company is hard or impossible use in other company even if the companies' profiles are very similar. Experience so far demonstrates, that DSL for each enterprise contains nuances specific for each enterprise. Previously developed DSLs can be used in the very beginning of modeling and help to recognize specific of new domain.

We agree to the supporters of MDSO about growing role of models in IS development process [11] and we plan to study modeling and usage of DSL as one of method for company business process arranging and IS developing more closely in near future.

### Acknowledgement

The work is supported by a European Social Fund Project No. 2009/0216/1DP/1.1.1.2.0/09 /APIA/VIAA/044.

### References

- [1] T. Stahl and M. Markus, *Model-Driven Software Development*, John Wiley & Sons, 2006.
- [2] *MDA Guide Version 1.0.1*. OMG. Available from: <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [3] *BPMN – Business Process Model and Notation*. OMG. Available from: <http://www.bpmn.org>.
- [4] *UML - Unified Modeling Language*. OMG. Available from: <http://www.uml.org>.
- [5] J. Bicevskis, J. Cerina-Berzina, G. Karnitis, L. Lace, I. Medvedis, and S. Nesterovs, Practitioners view on domain specific business process modeling. In: J. Barzdins, M. Kirikova, editors, *Databases and Information Systems VI*, Selected papers from 9th International Baltic Conference DB&IS 2010, IOS Press, 2011, 169-182.
- [6] A. Schleicher, High-level modelling of development processes. In: B. Scholz-Reiter, H. D. Stahlmann, A. Nethe, editors, *Process Modelling*. Springer, 1999, 57–73.
- [7] J. Cerina-Berzina, J. Bicevskis, and G. Karnitis, Information systems development based on visual domain specific language BiLingva. In: T. Szmuc, M. Szyrka, J. Zendulka, editors, *Advances in Software Engineering Techniques*, Lecture Notes in Computer Science **7054** (2011), Springer, Berlin, 124-135.
- [8] J. Barzdins, A. Zarins, K. Cerans, A. Kalnins, E. Rencis, L. Lace, R. Liepins, and A. Sprogis, GrTP: transformation based graphical tool building platform. In: *Proceedings of the MoDELS 2007 Workshop on Model Driven Development of Advanced User Interfaces*, Nashville, Tennessee, USA, October 1, 2007. Available from: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-297/>.
- [9] J. Barzdins, S. Kozlovics, and E. Rencis, The transformation-driven architecture. In: *Proceedings of DSM'08 Workshop of OOPSLA 2008*, Nashville, USA, 2008, 60-63.
- [10] J. Barzdins, A. Zarins, K. Cerans, M. Grasmanis, A. Kalnins, E. Rencis, L. Lace, R. Liepins, A. Sprogis, and A. Zarins, Domain specific languages for business process management: a case study. In: *Proceedings of DSM'09 Workshop of OOPSLA 2009*, Orlando, USA, 2009. Available from: <http://www.dsmforum.org/events/DSM09/>.
- [11] L. Cao, B. Ramesh, and M. Rossi, Are domain-specific models easier to maintain than UML models? *IEEE Software* **26**(4) (2009), 19-21.