Modelling of Information System Requirements

Jaroslava Kniežová, Ing. PhD
Comenius University
Faculty of Management
Department of Information Systems
Bratislava, Slovakia

Abstract

Exact specification of system requirements is very important factor in system developing process. Any shortage in this step will affect all later works and may lower the quality on the output. Therefore the attention should be focused on the way the requirements are defined and modelled. Usually the methodologies recommend using use case diagrams and UML language for this purpose. This article describes the possible ways to model requirements of information systems and catch more relevant information in diagram.

Building of new information system is a complex process consisting of many steps which have to be done before the final product is prepared for the customer. It is very important to meet customer needs and expectations so as to keep him for future projects. There are more approaches to develop new information system in which different strategy, methodology, modelling techniques or life cycle can be used. At the present time the agile approach and methodologies are very common and recommended especially for developing of some types of information systems. No matter which approach is used, the requirements stand always at the beginning and also at the end of the developing process. Project begins with specification what is going to be built and should end with the evaluation if the customer is satisfied with the product. And both depend on perfect requirements specification. Because if this isn't done well the new information system would not reflect the needs and what's more, the solution provider couldn't recognize it.

Gathering requirements of the information system from the future user

No matter which approach is used in developing process of information systems, it always starts with gathering the requirements. According to these first requirements other project properties are usually set (price, time, etc.). The first form of the discussed requirements is usually text written as notes. And this is the moment at which many developing companies start to develop the system and take no more time to define the requirements and model them. The time shortage is usually the reason and if the project is not large (functionally) or the project is similar to many others the developing team has done in the past (they have experiences) it can be an advantage at the end. To shorten the time means shorten projects costs of course. On the other hand if this happened to show as bad decision, the project result is very bad then. Modelling the systems requirements usually helps to prevent this.

Using UML Use Case Diagram for Modelling Information System Functional Requirements

It could be said that most often and also traditional way to define functional requirements for the new information system is to model them using UML language and its use case diagrams. The requirements must be discussed well with the customer and the big advantage of these diagrams is, that the used symbols are quite easy to understand also for the customer who has no or only little experiences in software developing area. Also customer who doesn't know how the software works is able to read what kind of user will use the modelled information system and also what will such user do with it.

Another advantage of use case diagrams can be the fact that they can be worked up quite quickly. The symbols used in these diagrams are very simple and can be written or rewritten very often. Thanks to this the focus in requirements definition process is mainly on the communication between the customer and solution provider instead of modelling diagrams. This is the very good attribute of these diagrams and therefore they can be used also when agile methodology is applied (more about agile approach can be found in [AGI11].

Use case diagrams are very often used at the present time and certainly every member of solution provider team knows them. This is another reason why to use them. It speeds up modelling what of course is expected from the UML language.

Functional Requirements of the Information System and their Position in the Life Cycle

It was already mentioned that the requirements appear at the beginning and also at the end of the developing process. But the beginning and the end, that means the tasks which are done at the beginning and the end may be different. Applying an agile methodology the process probably starts with requirements and quickly continues to coding, ends with functional software supplied to the customer.

On the other hand, there are other types of developing process, in which the requirements are preceded with defining the business processes, their optimization or changing the process map in some areas at all. In this case the company processes for which the new information system is to be built are evaluated and reconstructed first. The main fact of this case is, that quite big documentation of the company's processes exists before the requirements of the new information system are defined. Starting the developing process with requirements definition based only on the discussion with the customer would be a mistake. This may not cause quality decreasing of the final product, but can take more time than needed because of doing some redundant activities. As the adequate documentation exists the solution provider team should certainly use it as the basis to start from and complete the needed details according to discussion with the customer.

Based on the facts written above it can be said that the requirements are defined at the beginning of the life cycle but in those cases when business processes are mapped or/and reengineered in the customer's company, the information system requirements are not the first step. Therefore there can be defined steps in the life cycle, which the requirements definition should reflect and other steps which should reflect the defined requirements. When the software team keeps the extra-model relationship [NEU07] of <<trace >> stereotype between the business models and the system requirements as the first point and between the requirements and the software models as the second point, this is the good condition for correct defining requirements and also their realisation in the final result. As these two points are the weak place in the project life cycle, keeping the traceability between business diagrams – UML diagrams (requirements) – UML diagrams (software design) is quite important in the system developing process. This fact is also remembered in many methodologies, for example in RUP methodology [IBM11], the workflows: Business Modelling, Requirements, Analysis & Design are defined and the traceability between these three workflows is recommended. The facts described and defined in the Business Modelling workflow should be the basis for modelling in Requirements workflow and also the software should be designed in the next Analysis & Design workflow based on previously defined requirements.

Business Process Modelling as the Basis for the Future Information System

When the company processes are defined the process map is created. Business Process Models are worked up usually in two different points of view [REP12]. The first is to model the processes as they are realised at the present time in the company. Details about processes, which are actually done, are defined. This is the analysis of the processes and is later used as the basis to reorganize the company. This is called also as 'as is' model. The other type of business diagrams display the future processes, i.e. details of the processes which are realised after reorganization or optimization. In these models also the future information system should appear if it is planned to build. This models are called as 'to be' models. It can also be looked at it as requirements of the company, because this kind of business models covers the requirements of the processes done by employees (in the future) and future information system as well.

When the solution provider team starts to build the information system and collects the requirements, previous worked up business models of type 'to be' should be used. This can assure that no previously worked up detail is lost even if another member of the solution provider team does it. The software developer can use the business diagrams as additive source of the requirements and confront it to the requirements specified by the customer. Doing this, another weak place is eliminated as this confrontation prevents bad understanding the same requirement by solution provider and customer.

If the business model is created together with the new information system, it means in the same project or in two projects being realised in the same time (one after the second) another advantage could be taken. It can be remembered on the information system requirements when doing the business model and cover also processes of the future information system to it. If this is done, the traceability can be kept very easily as the processes of

future information system should be transformed to its requirements. As we say that the future information system will do this – it is the requirement on it.

On the picture 1 there is the business process diagram, which defines the process of testing students. There are two subjects who participate on this process and each of them has its activities. Each subject when has finished some of the activities goes to one of his states. So each subject does activity, gets to state and after done other activity, gets to other state. The whole diagram represents one process, so at the end of the process the student has finished and given his test and got other instructions and the teacher has all tests from the students and has given the other instructions to them. This process is displayed using business diagram on the picture 1.

But the new information is supposed to be used in this process in the future. So it would be good to model this diagram also with the details what information system will do in this process, its activities and states. This could be very valuable information for the following steps in developing the new information system.
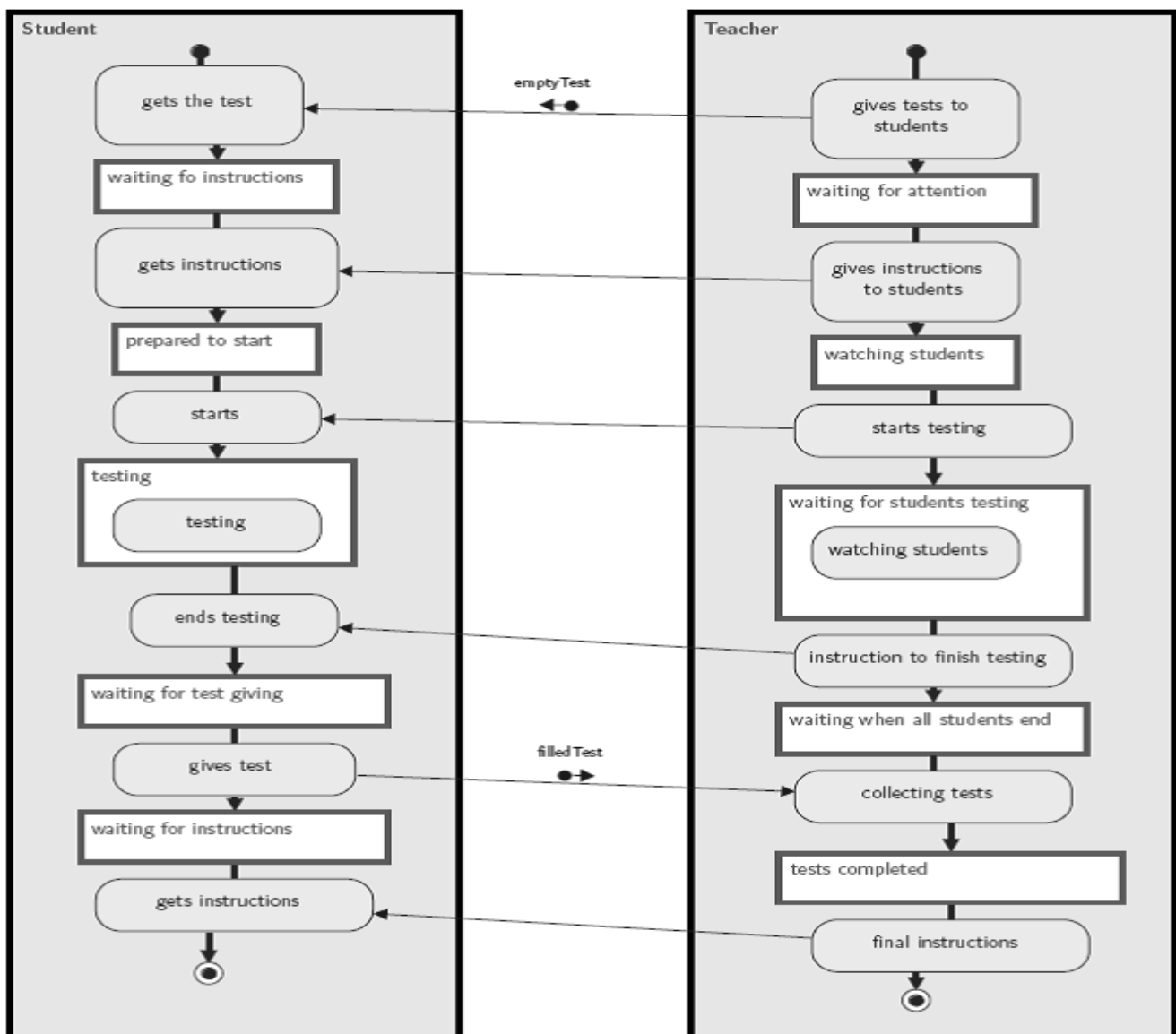

Picture 1 Business Process Diagram

On the picture 2 there is business diagram for the same process, but also the activities of the future information system are covered. It is very easy to define future function of information system within this process and therefore it is very easy to define the requirements depending on it. Also this diagram is very useful when discussing the requirements with the customer because the customer can see what will be done and by who, when testing the students and can easily imagine the way of using information system in this process.

Including activities of information system in the business diagrams is useful step for following design of information system. It is very important to do this in 'to be' model because the business diagram should display the future process and the future (required) activities of the information system. Based on this diagram the software team should define the exact requirements and model them in UML diagram. Use Case diagram with the requirements of the information system for the testing process is displayed on the picture 3. According to the used methodology the modelled requirements should be in the next step written in the matrix of requirements realisation [NEU07] which helps to control if every requirement is covered by the system function (use case). This can prevent situation when all requirements are defined correctly, discussed with the customer but one or more of them are not covered in next software models because it was forgotten during the developing process.
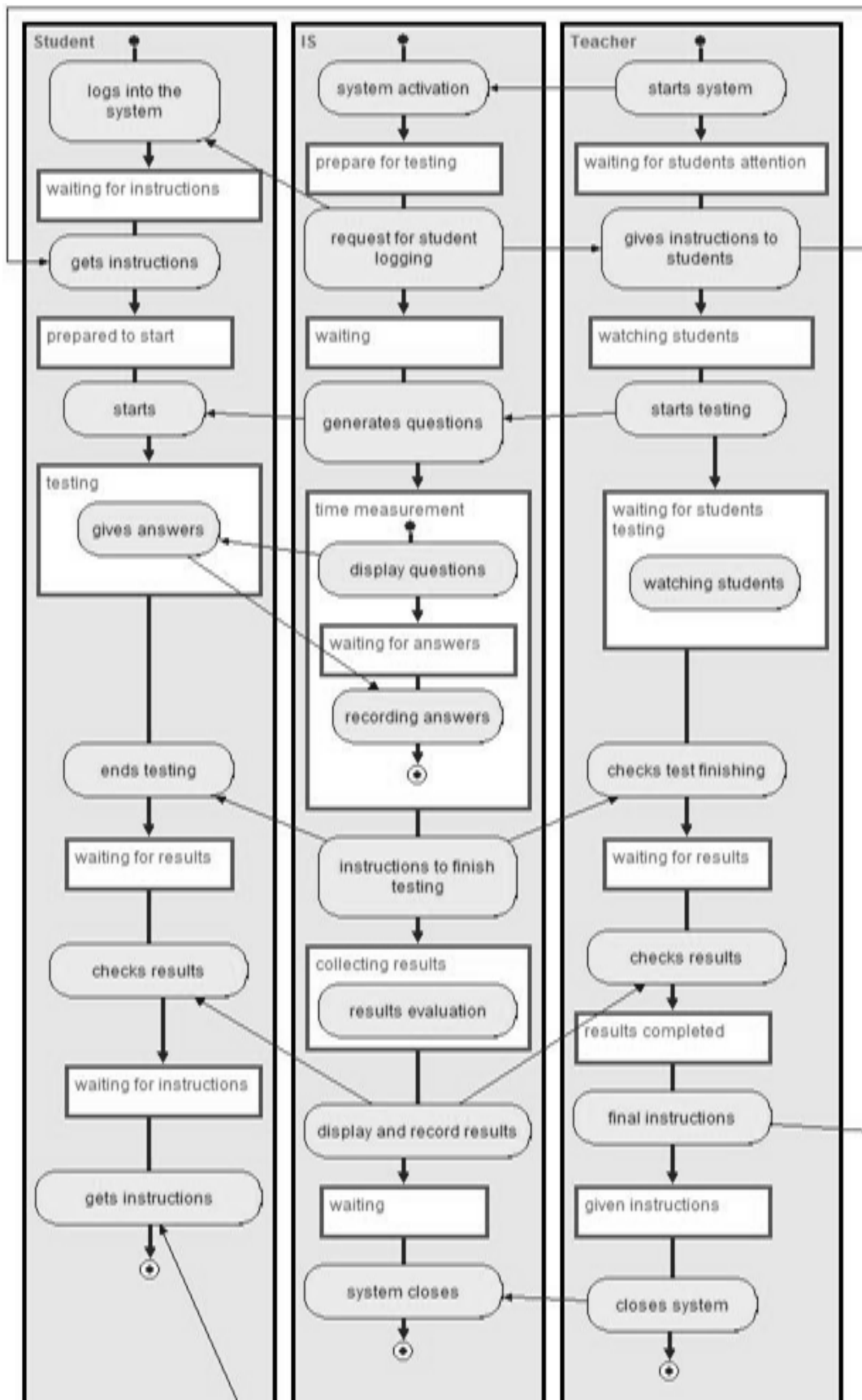
Picture 2 Business Process Diagram with Information System Activities
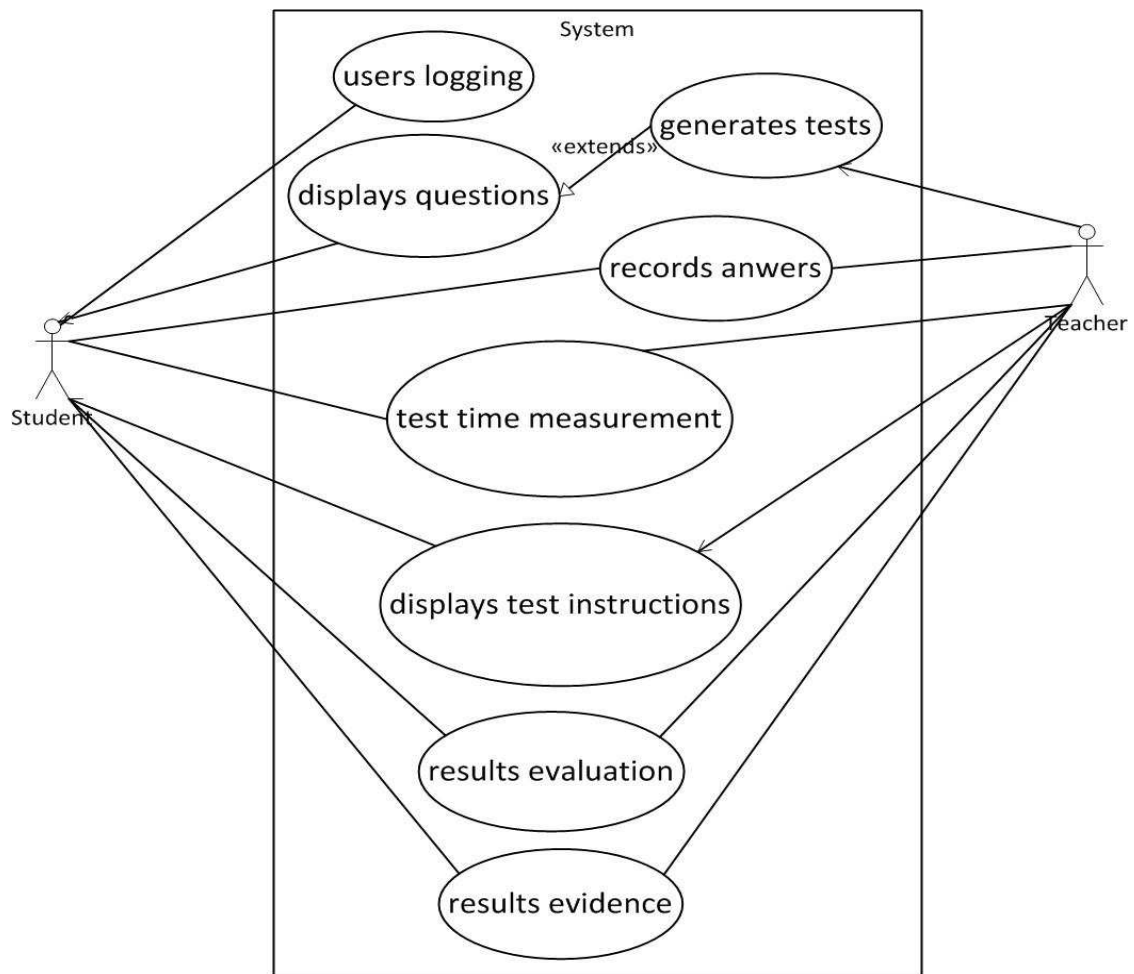
Picture 3 Use Case Diagram – Requirements of the Information System

MDA Architecture and Requirements Modelling

MDA Architecture (Model Driven Architecture) is defined as software developing process of model transformation [NEU07].

CIM – Computer Independent Model: Future information system is modelled on conceptual level. The main functions of future information system are described and also the requirements on the information system are specified. This model is transformed into PIM model.

PIM – Platform Independent Model: Information system on logical level is modelled. The system is modelled in details – its functions, objects, data structures. In this model the platform specific details are not modelled. This model is transformed into PSM model.

PSM – Platform Specific Model: Information system is modelled on physical level with details of used platform and implementation details. This model should be transformed into code. More details about MDA can be find in [TAV13]. This approach of developing system is considered to be the future of developing systems and UML modelling [NEU07].

It is available to realise business model, requirements design and software design with keeping the traceability as described in previous part. The software developers can create business models, requirement models within the CIM model and the software model within the PIM model. The important thing is to keep the <<trace>> relationship between them especially between requirements model and software model as they are in different levels of MDA architecture.

Based on the described facts it can be claimed that developing software using MDA process is not problem for keeping traceability between models. The advantages of defining activities of information system in business diagram and defining its requirements based on them can be fully taken.

Agile approach and Requirements Modelling

At present the majority of software projects are realised using the agile approach. The software companies aim to shorten time spent on information system developing so as to be more competitive and to lower project costs. In this approach the developing team starts to code new information system without having complete models and even requirements of the new information system. It is expected that other (new) requirements the customer creates during the developing process or even at the end of this process. The new version of the system or new part of the system will be developed then and these new requirements will be realised in it.

Business processes are usually not modelled in this approach, so it is not possible to keep traceability between models in this situation. Only basic requirements can be defined and modelled. Matrix of requirements realisation can be done so as to make sure that no one important requirement will not be forgotten. This fact corresponds with the main idea of the agile approach when the models should be created only in very necessary cases [AGI11].

When to Use Business Diagrams for Requirements Specification

As the defining of the requirements is very important step, the developing team should pay attention to it. The requirements should be modelled if the used approach allows it. It can be said that if the agile approach is used, the requirements are not or only minimally modelled. When using other approach (for example MDA, or other type of life cycle of the information system) it is possible to model the requirements.

Using Business Diagrams for the requirements specification brings more details and more information about future system functionality, so if the approach allows modelling, using business diagrams is always good decision for requirements specification. In cases when the business diagrams are done, the developing team should always use them for the requirements specification. In some situation it is possible even to create the business diagrams because of requirements specification needs. Especially situations when some important functionality is designed.

The type of the project of system developing influences the chosen approach and this influences the level of modelling. According to this it can be said that the decision of modelling the requirements and using the business diagrams for it depends on the choice of used approach for developing the new information system.

Conclusion

It is possible to define the advantages of using business process diagram for requirements specification:

-   Keeping traceability between models: Business process diagrams can display the functions of the future information system and therefore defining its requirements is much easier and more correct.

-   Keeping defined information between the developing teams: It can often happen that business analysis is done by other part of the developing team than the software design or even by another solution provider. If these models contain also the participation of the future information system, the other team – doing the requirements definition, has valuable information about the expected functionality of the future information system.

-   Supplying the customer with the information how the future information system will work. As the business diagrams are quite understandable to the customer and contains also the human activities, he can easily control (and discuss) the expected functionality of the future information system

It is possible to make a conclusion that creating business diagrams and using them as the basis for the requirements definition is very useful step, bringing more advantages for the development process of the new information system. The developing team should always do this step when developing the system using such approach in which models of information system are created. In cases when the agile approach for the development process is used, this step is not realised because of minimization of system modelling.

References

[NEU07] Neustadt,I., Arlow, J.: *UML 2 a unifikovaný proces vývoje aplikací*, Computer Press, 2007, ISBN: 8025115038 (in Czech)

[TAV13] Tavač, M., Tavač, V.: DBRE and MDA integration, In: Objekty 2011 proceedings of the 16th international conference on objectoriented Technologies, s.52-65, Žilina, november 2011, ISBN 978-80-554-0432-5

[REP12] Řepa, V.: Procesně řízená organizace, Grada, 2012, ISBN: 9788024741284 (in Czech)

[KNI09] Kniežová, J.: Metódy v softvérovom inžinierstve, EKONÓM Bratislava, 2009, ISBN:978-80-225-2696-8 (in Slovak)

[AGI11] Composite authors: Manifesto for Agile Software Development, available at: http://agilemanifesto.org/

[RAT13] IBM: *UML notation guide*, White Paper, available at: http://www.rational.com/uml/technical_papers/notation_guide.htm

[IBM11] IBM: *RUP best practises*, available at: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf