Systematic Modeling Process of System Behavior

Radek Silhavy, Petr Silhavy, and Zdenka Prokopova

***Abstract*—** In this articles we focus on the system behavioral modeling. We discuss systematic modeling approach for the system modeling. For the proposed modeling approach, the system engineering techniques are adopted. The role of the SysML basic principles and diagrams is discussed.

***Keywords*—**System engineering, system modeling, behavioral modeling, sysml, uml

I. Introduction

THE System engineering [1] is understood as complex discipline for the system design and analysis of the system. The system engineering as a discipline cover a broad number of subject. It can focused on military systems [7], robotics [8,9] or management systems [10].

The system is for the purpose of the system engineering defined as a set of the components which are interconnected and provides the group of emergent properties [1]. These emergent properties are derivate from the properties of the system components, but new dimension is added by the system integration and by creative approach [11].

A system engineering processes are about preparing [2] generic stakeholder goals, requirements, system design, evaluation of alternative system designs, allocation of functional requirements, system verification. All of these activities lead to creation of the balanced system.

The system development generic process is described by waterfall model [1]. The process should be composed of:

1) Requirements Engineering Definition and Elicitation.
2) System Design.
3) Sub-system Development.
4) System Integration.
5) System Installation.
6) System Evolution.
7) System Decommissioning.

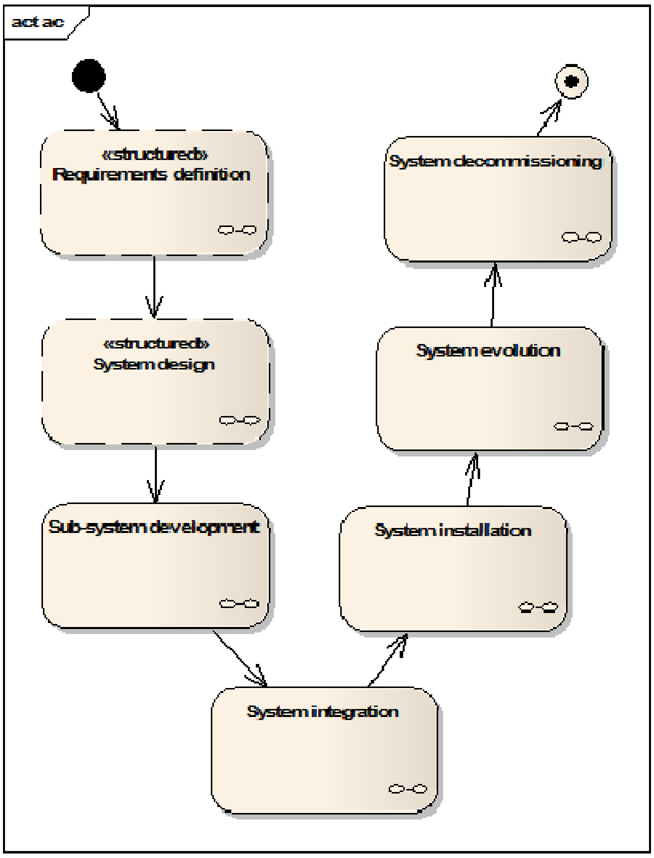

Fig 1: Waterfall Model of the System Engineering

The system requirements engineering general name for the specific sets of the software engineering techniques, which is used at the beginning of the software cycle. The purpose of these techniques is to discover stakeholder's needs.

The requirements gathering process in system engineering has these basics steps, which are graphically described on the next figure.

Manuscript received June 20, 2011. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Research Plan No. MSM 7088352102.

R. Silhavy, Tomas Bata University in Zlin, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic (phone: +420 57 603 5015; e-mail: rsilhavy@fai.utb.cz).

P. Silhavy, Tomas Bata University in Zlin, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic (phone: +420 57 603 5015. e-mail: psilhavy@fai.utb.cz).

Z. Prokopová, Tomas Bata University in Zlin, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic (phone: +420 57 603 5011. e-mail: prokopova@fai.utb.cz).

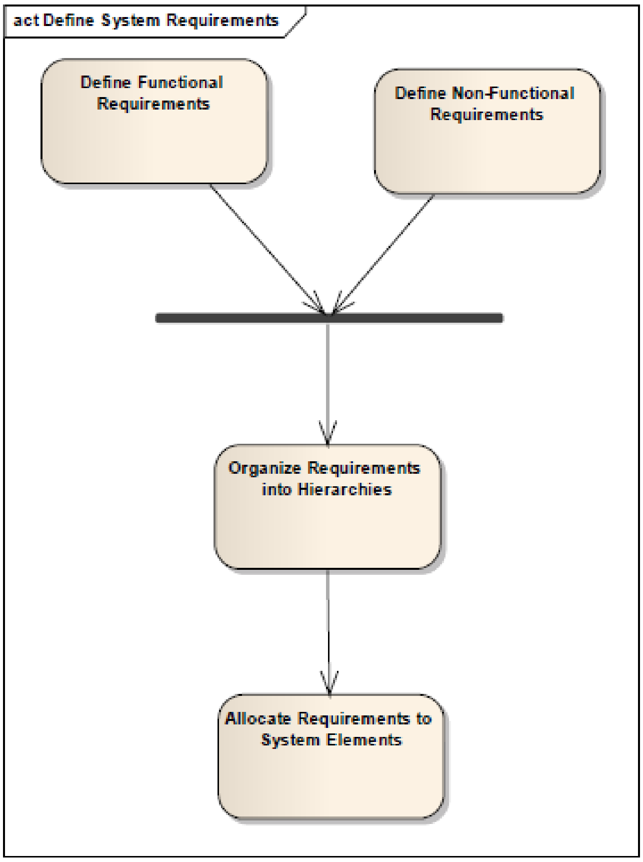

Fig 2: Requirements Definition Model

The requirement [1] is a description of the functionality or condition which stakeholders define for the system. After the first round of requirements gathering is talked about the raw or abstract requirements. These raw requirements are list of functionality or condition for the proposed software, which is unanalyzed yet. The most important in this phase is to establish the project goals, which should be achieved.

The next group of the requirements is non-functional requirements. The purpose of these requirements is familiarized system designers with problem domain and conditions in the domain. Well-known examples of these are reliability, performance, safety or security. These non-functional requirements are critical in the system evaluation very often.

The last part of the requirements gathering is so-called system characteristics. The system characteristics are commonly prepared in negative way. It means, that system design specifies what irrelevant system behavior is.

The system design phase deals with association of the system requirements to individual sub-system, more specific to system components. The set of the system requirements is studied and individual requirement is associated to proposed component. One of the most common techniques, which could be applied in this process, is to sub-system design first. Than system designers is able to associated selected requirements to the specific subsystem. The process of the sub-systems identification and requirements association is interactive and is commonly modeled in form of spiral.

In the system modeling phase is a system architecture designed. This activity is based on sub-system (or component) design. The system is modeled as group of interconnected blocks, where connection indicates data flow or other form of dependency. The main task is to create more concrete definition of the sub-systems, which were set-up in the system design phase.

The sub-system development phase works with sub-system or system components implementation. The sub-systems are implemented in parallel. This is because of sub-systems in the system engineering is not only hardware/software, but also should be create by civil engineering.

Next step in the system engineering life-cycle is the system integration. The implemented subsystem are integrated into the system [1]. The integration is an incremental process. The sub-system are integrated, when their implementation is finished. This approach is time-less consuming, then legacy approach, when all components were implemented together.

The system evolution and system decommission are the final phase of the system lifecycle. A system designer should care about system improvements during its production time. They also should take care about times, when system should be prepared for out-of-service elaboration.

The main task of the software engineering is a system development from very beginning (requirements elicitation) to system decommission.

II. Problem Definition

In the system engineering lifecycle, which were described above, basic ideas of the system engineering were concluded. In the system engineering are used well-known techniques for modeling. Probably the most common is block diagramming. This model is used for system design and for sub-system analyze. Secondly data flow diagrams are common for definition of data and process designing.

Today state-of-art in research of graphical documentation of the system development is System Modeling Language (SysML). The SysML is a graphical modeling language [2], which is derivate from Unified Modeling Language (UML). UML is an industry standard in the scope of the software engineering. SysML is a extension of UML, this two basic technique shared basic principles and some types of diagrams are used in both. The SysML take important role in the system engineering, because its usability in all phase of software engineering process.

The modeling approach which is offered in this article is based on the combination of the most valuable principles, which can be found in the system engineering and software engineering.

III. System Modeling Language

SysML is relatively young modeling language. Its history is written from 2001, when Systems Engineering Domain Special

Interest Group were set-up [3]. Today, there is 1.2 version, from 2010 valid.

A. Language Overview

The modeling project support analysis, specification, design, verification and validation of systems. SysML therefore support all phases of the system engineering lifecycle.

The system components should be described by structural composition, interconnection and classification. Secondly by function-based approach this is based on messages between objects. This aspect should respect to constrains, which are derived from physical system structure or from performance properties. Important role takes association of system functionality to each of the system components.

In the SysML nine of diagrams are of recognized [2].

Requirements diagram is used for graphical interpretation of requirements and their connection to other requirements and to other elements and entities in project – such test cases, use cases. For those how are familiars to UML, this diagram is new in SysML and have never be used in UML.

Activity diagram is based on UML activity diagram; there is slightly different usage of it. Basically is used for modeling of flows of actions based on the availability of inputs, outputs or control. Transformation of actions is also modeled by activity diagram in SysML.

Sequence diagram is used for representing of message flow between objects.

State machine diagram presents set of state of the modeled entity and events which generated message upon which entity state is switched.

Use Case diagram is used for the system function description. This model is composed of the actors, which are external entity and of the use cases. The use cases represent system functions or algorithms. Each of the use case has to realize one of the requirements as minimum.

Block definition diagram is used instead of the class model in UML. The purpose of the block definition is to model system structure.

Internal block diagram is similar to UML composite structure. The main idea of this diagram is to model internal structure of the each individual part of the proposed block. Very important here is modeling of interface and communication between block´s entities.

Parametric diagram is SysML origin and have no predecessor in UML principles. Is used for modeling system parameters and constrains, should be used for critical – hazard system states.

Package diagram is useful for model organizing. Model elements should clustered by its stereotypes. Packages also should be used for creation of a large project structure.

IV. Proposed Behavioral Modeling Process

In the picture 3, there is a diagram of the behavioral modeling process. It is adopted the principles which are used in the system engineering and in the software engineering. The basic stages of the modeling:

1. Requirements modeling
2. Use Case Modeling, Finite State Behavior Model
3. Allocation of UseCase to Requirements, Allocation Requirements to States.
4. Modeling Interactions for Use Cases.

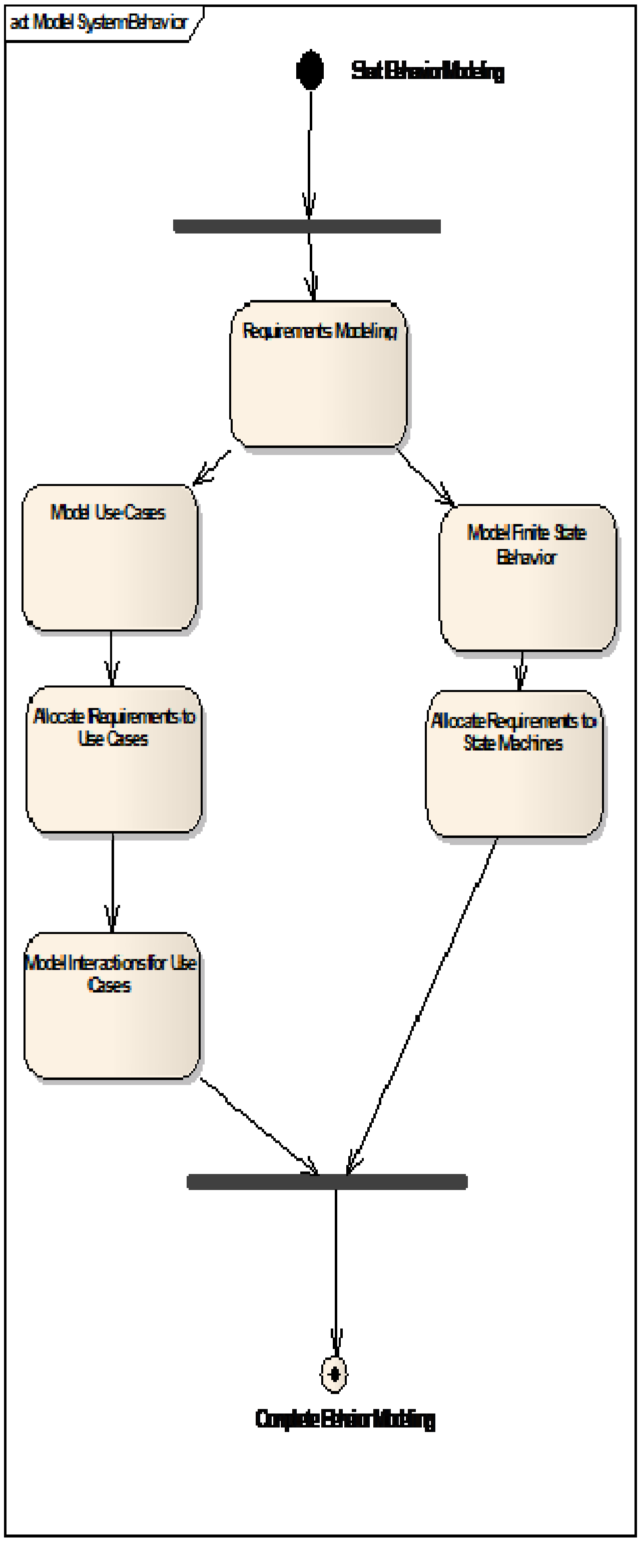

Fig 3: Behavioral Modeling Process

Steps 2 and 3 contain parallel actions. The first activity in proposed modeling schema is a requirements modeling. The requirements modeling are a very important phase. In this stage can several methods be used.

A. Requirements modeling

The selected methods of the system requirements gathering are described in this chapter. There are many research methods, which were adopted or modified for requirements gathering or elicitation [5].

The Interviews; very common in either in the scope of system and software engineering [1, 5]. The interview is origin for social science research. It is human based activity. The requirements engineer has to be able to discuss with stakeholders or with future system users.

The interview has three basic types; structured, semi-structured and un-structured. The most valuable for the requirements definition is the un-structured interview. Contrastingly, un-structured interview have to be held by some experienced engineer. The structured interview commonly leads to missed some of important requirements. The set of questions is not well prepared and structured.

As well-working compromise; the semi structured interview, where basic set of the question is prepared and used.

The brainstorming is very useful addition to the semi-structured interview [5]. If there are more then on stakeholders, the whole group is questioned in same time. The answers and discussions of the all group are noticed.

The laddering [5] is a technique, which is used as part of the brainstorming. It allows moderating the debate and brings hierarchical structures of the stakeholder (or stakeholders) answers.

The questionnaires; is probably the most important impersonal method. It is very useful in preliminary [5] requirements gathering. The questionnaires lack in discovering new facts or dimensions of the proposed system. Therefore have to be prepared by an experienced requirement engineer with huge knowledge of the problem domain.

The task analysis; an activity which is based on task decomposition [5,6]. The top-level tasks are designed first and then by top-down approach all sub-task are derivate and described. In the task analysis, users' tasks and system's tasks are at same level of importance. These tasks can be documented in form user or system stories, which were already described in the chapter 3. The individual task should be resulted in more than one story, even further in more than one requirement.

The observation; a method, which comes to system engineering from the ethnography and other sociological research. The observation allows observing users in their own environment. This method is valid for human-centric system design. For observation are valuable hidden cameras, with respect to privacy. Users very often change their behavioral, if they are informed about observation.

The prototyping is one of the most valuable solutions for requirements capturing. The prototyping has no value for early phase of the requirements engineering. It allows determining very concrete and detailed requirements, in the time, when introductory requirements are already collected.

B. Use Case Modeling

The Use Case is description of the activity of the action in the system. The use case model in the system engineering consist of the actors, use case and associations. Use case are written in form of the scenario. The scenario represents sequence of the steps, which represents using of the system by an actor.

C. Finite State Model

The Finite State Model represents the system states. In proposed methodology used it for modeling systems states and correlation between the states and the use cases model.

The operations, which are modeled in the finite state model, represent the actor's activity.

D. Allocation Requirements to Use Cases and to Finite State

The system designer point –of-view is to observe, which requirements are realized by which use cases. The goal is to create the 1:1 relationship between the use case and the requirement.

Similar situation is in the field of the finite state machine. The system designer have to attach each individual requirement to an individual state.

These association will lead to the systematic system design, which behavioral modeling is accurate.

E. Modeling Interactions of Use Cases

In proposed methodology, interaction are modeled by using activity diagrams.

V. Sample project of behavioral modeling

In this chapter will be introduced sample projects, which contains example model of each type of the diagrams. For purpose of this article authors adopted project which was prepared by SparxSystems as sample project in SysML language [4].

In this sample is described development of the audioPlayer. The main task of the project team was to offer a solution which was successfully in usability and which offers appropriate functionality.

The behavioral modeling is a description (Figure 3) how the proposed system will interact with the actors and with entitles which is out of boundary of the system.

The first step in the creation of the system behavioral model is the requirements gathering.

In the figure can be seen model of the requirements. The diagram illustrates hierarchical structure of the requirements. On the top of the tree the Specification package can be seen. Other parts are interconnected by containment association. The specification is clustered into sever groups. Groups are User Friendliness, Durability, Performance and Media Capacity.

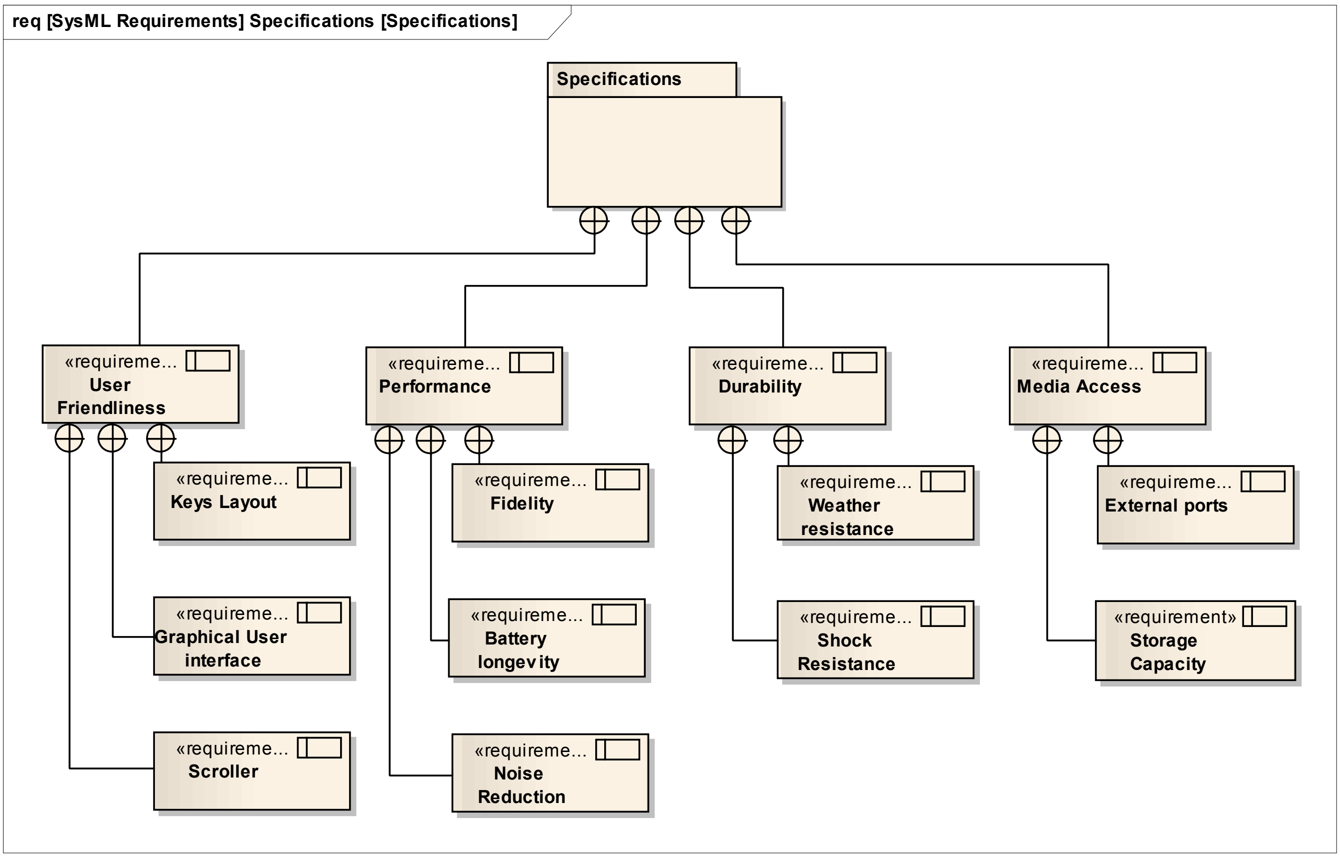

Fig 4: Requirements Model

The user friendliness group defines set of the requirements, which deal with quality of service of the audio player. Keys Layout, Graphical User Interface and Scroller are primary requirements, which take important role in user satisfaction.

When the requirements model is finished, next step is use case model. In the use case model there is an algorithmic definition of the actors' activity in the system. In the figure system boundary, actor and use cases can be seen. The boundary is named as Playlist Maintenance, because this model describes only activities of the listener, which are available as maintain the playlist. Inside of the each individual use case is scenario description. The scenario is a sequence of steps, which describes a track, should be downloaded. In this Use Case model include association is used. The "include" association is used for situation, when the use case contains other use case for achieving its goal.

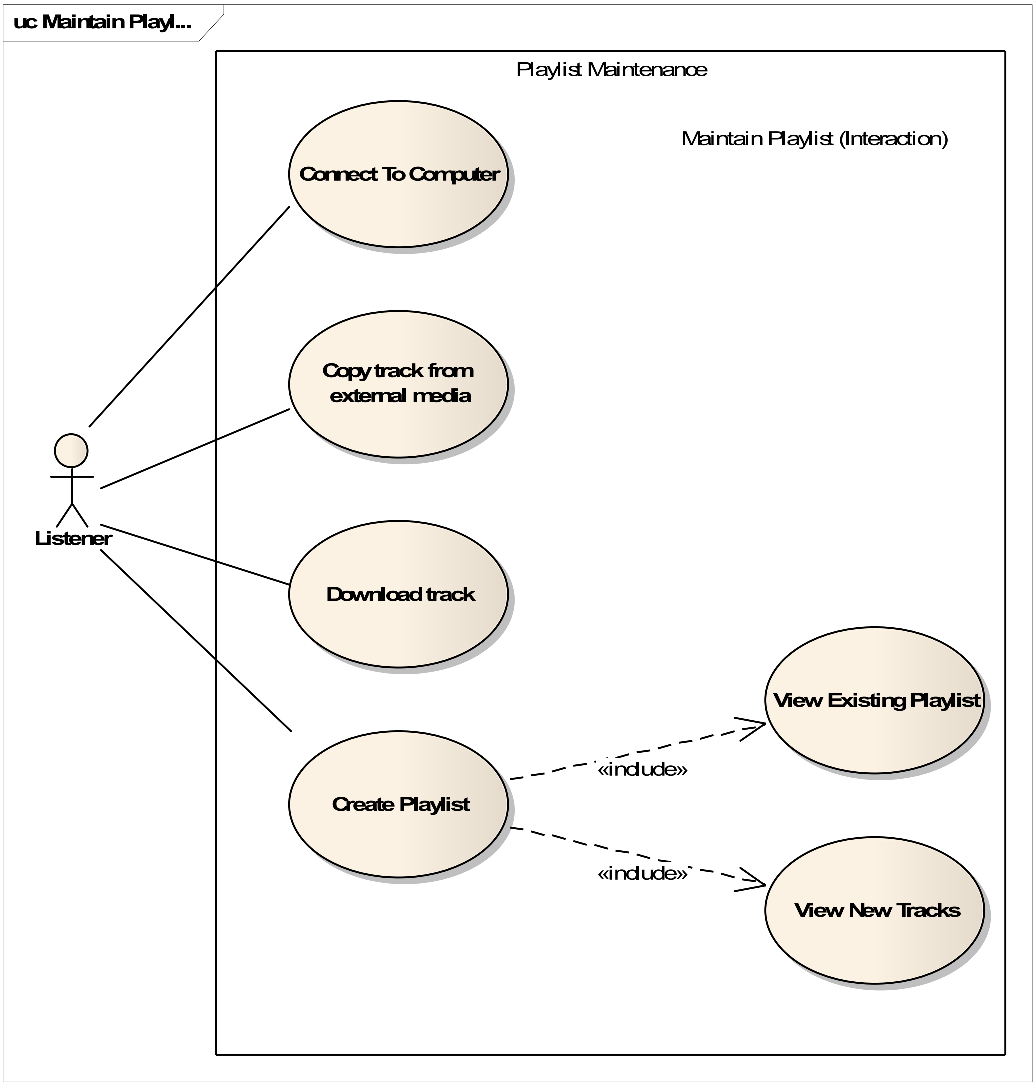

Fig 5: Use Case Model

The Use Case models (Figure. 5) are derivate from requirements. The Use Cases are commonly prepared for individual system blocks. This is rigorous connection between the requirement and the use case. This connection should be modeled as realization in the use case model or can be documented in form of the responsibility matrix.

The interaction in form of sequence diagram is useful for modeling overview of operations. In the Fig 5 can be seen overview of diagram, which describes all possible activity of the actor, called Listener. There are used fragments, which used for referencing individual activity descriptions. These are named as ref. The alt abbreviation is used for conditions. In this context the viewnewtracks can be executed only if the playlist exists.

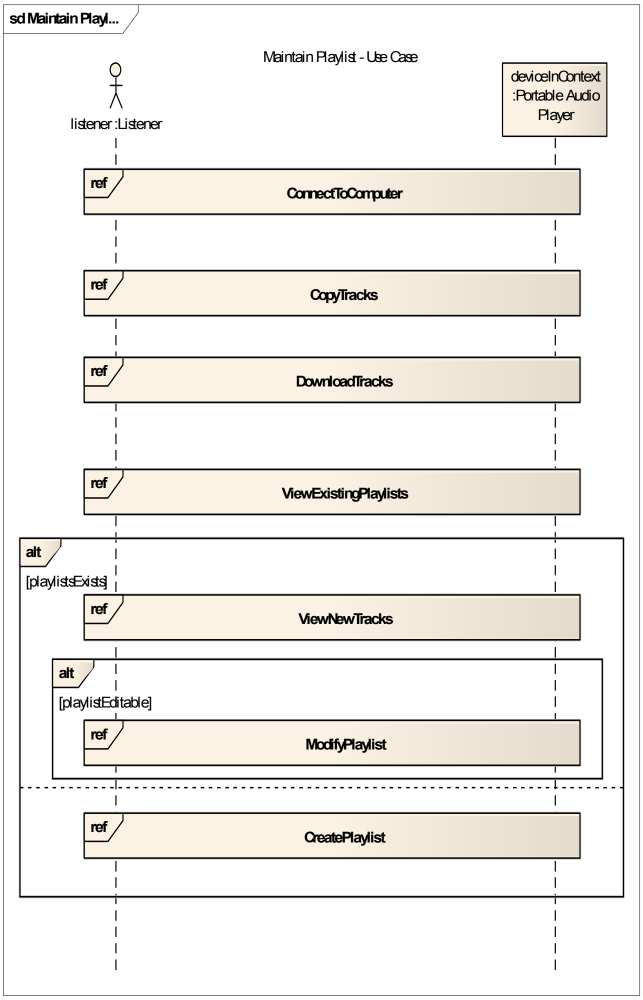

Fig 6: Sequence Diagram for Playlist Maintaining

The last view of the system can be done by state machine diagram. The most important noticeable fact here is, that by state machine diagram is modeled same system in the different view only.

The Figure 7 shows modified version of the state machine diagram. It does not present only two basic states – idle and connected. The state connected is describes in form of the activity diagram. In the connected state tracks could be copied and downloaded. These two operations are in a parallel section, which is created by using fork/join artifacts.

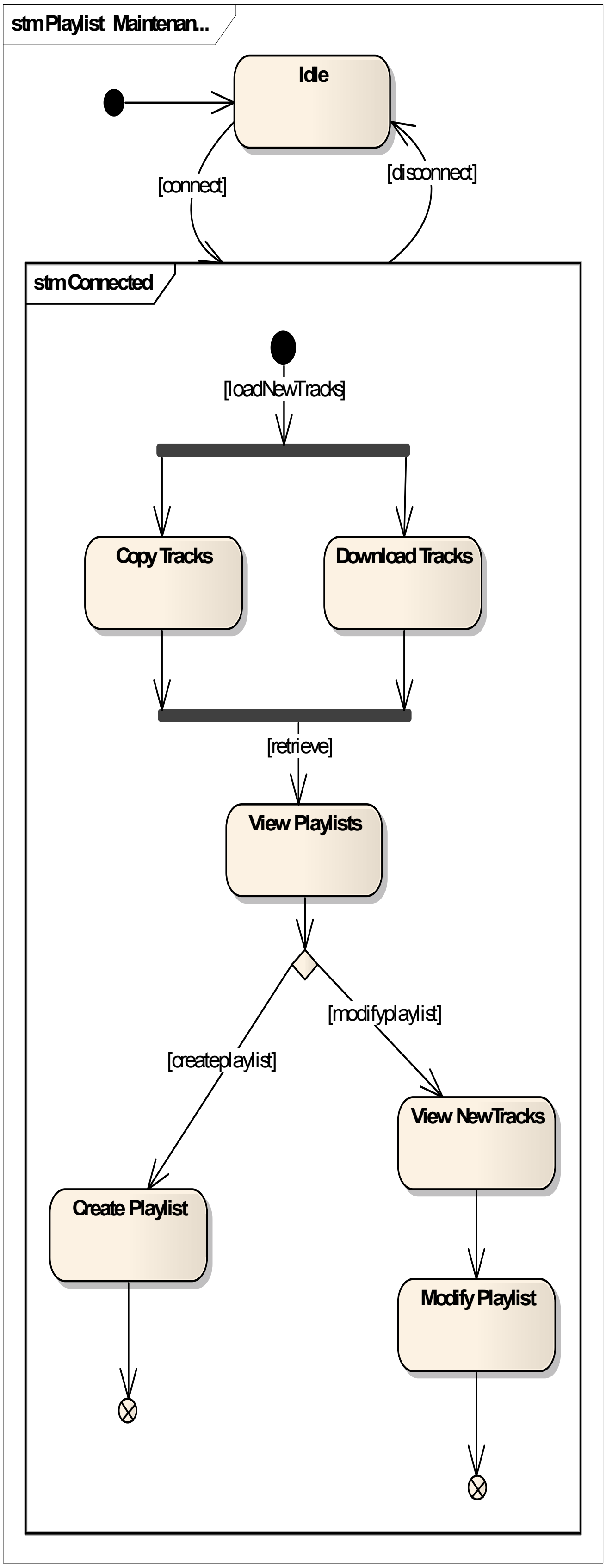

Fig 7: State Machine Diagram with Activity

VI. Conclusion

The idea of the contribution was to introduce System Modeling Language for modeling system behavior. The system engineering were described and connection between system modeling language diagrams and the system engineering phases were illustrated.

For the example purpose part of the audio player model were presented.

The modeling project support analysis, specification, design, verification and validation of systems. SysML therefore support all phases of the system engineering lifecycle.

The SysML uses number of diagram, which allow to a system designer model the proposed system in many views.

The system behavioral modeling deals with requirements engineering, use cases elicitation and state modeling of the system.

Further research in system modeling is focused on the improvement of the simulation environment and to the model based development, which probably the most important in the development of the system engineering.

Acknowledgment

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Research Plan No. MSM 7088352102.

References

[1] SOMMERVILLE, Ian. Software Engineering. Eight Edition. Harlow : Pearson Education Limited, 2007. 824 s. ISBN 978-0-321-31379-9.

[2] FRIENDENTHAL, Sanford; MOORE, Alan; STEINER, Rick. A Practical Guide to SysML : The Systems Modeling Language. USA : Morgan Kaufmann, 2009. 5769 s. ISBN 978-0123786074.

[3] Object Management Group. Systems Engineering Domain Special Interest Group [online]. 2007-2011 [cit. 2011-03-20]. Available on WWW: <http://syseng.omg.org/>.

[4] ROSENBERG, Doug; MANCARELLA, Sam. Embedded Systems Development using SysML. [s.l.] : Sparx Systems Ply Ltd, ICONIYX, 2010. 68 s.

[5] Engineering and Managing Software Requirements. Berlin: Springer Berlin Heidelberg, 2005. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools, s. 19-49. ISBN 978-3-540-28244-0.

[6] Carlshamre P, Karlsson J.A usability-oriented approach to requirements engineering. In: Proceedings of the 2nd International conference on Requirements Engineer-ing, April 15–18, Colorado Springs, CO. 1996.

[7] Balla, J. 2011, "Dynamics of mounted automatic cannon on track vehicle", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 5, no. 3, pp. 423-432.

[8] Ouarda, H. 2011, "Cognitive tasks behavior of intelligent autonomous mobile robots", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 5, no. 3, pp. 610-619.

[9] Tokarz, K. & Manger, C. 2011, "Geometric and rough set approach to inverse kinematics for arm manipulator", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 5, no. 1, pp. 1-8.

[10] Nan, M.S., Nicolescu, C., Jula, D., Bolovan, C., Voicu, G.V. & Petre, G. 2011, "Practical aspects regarding spare parts reliability evaluation within an integrated management system", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 5, no. 2, pp. 238-246.

[11] Loginov, D. 2011, "Possibilities of modeling the creative part of engineering design process using the synergetic approach", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 5, no. 1, pp. 95-104.

Radek Silhavy was born in Vsetín in 1980. He received a B.Sc. (2004), M.Sc. (2006), and Ph.D. (2009) in engineering informatics from Faculty of Applied Informatics, Tomas Bata University in Zlin. He is a senior lecturer and researcher at the Computer and Communication Systems Department. His Ph.D. research was on the verification of the distributed schema for the electronic voting system.

Major research interests are software engineering, empirical software engineering and system engineering.

Petr Silhavy was born in Vsetín in 1980. He received a B.Sc. (2004), M.Sc. (2006), and Ph.D. (2009) in engineering informatics from Faculty of Applied Informatics, Tomas Bata University in Zlin. He is a senior lecturer and researcher at the Computer and Communication Systems Department. His Ph.D. research was on the electronic communication and services in a medical information systems.

Major research interests are data mining, database systems and web-based services.

Zdenka Prokopova was born in Rimavská Sobota, Slovak Republic in 1965. She graduated from Slovak Technical University in 1988, with a master's degree in automatic control theory. Doctor's degree she has received in technical cybernetics in 1993 from the same university.

She worked as assistant at Slovak Technical University from 1988 to 1993. During years 1993-1995 she worked as programmer of database systems in Datalock business firm. From 1995 to 2000 she worked on position lecturer at Brno University of Technology. Since 2001 she has been at Tomas Bata University in Zlin, Faculty of Applied Informatics. She presently holds the position of associating professor at the Department of Computer and Communication Systems. Her research activities include programming and application of database systems, mathematical modeling, computer simulation and control of technological systems.