KAUNO TECHNOLOGIJOS UNIVERSITETAS

Audrius Lopata

Veiklos modeliu grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas

Daktaro disertacijos santrauka

Technologijos mokslai, informatikos inžinerija (07T)

KAUNAS, 2004

Disertacija rengta 2000 – 2004 metais Kauno technologijos universitete.

Mokslinis vadovas:

doc. dr. Saulius GUDAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Informatikos inžinerijos mokslo krypties taryba:

prof. habil. dr. Rimantas ŠEINAUSKAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T) – **pirmininkas;**

prof. habil. dr. Rimvydas SIMUTIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T);

prof. habil. dr. Ignas SKUČAS (Vytauto Didžiojo universitetas, technologijos mokslai, informatikos inžinerija – 07T);

prof. dr. Albertas ČAPLINSKAS (Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T);

doc. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Oficialieji oponentai:

prof. dr. Olegas VASILECAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T);

prof. habil. dr. Vytautas ŠTUIKYS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P).

Disertacija bus ginama viešame informatikos inžinerijos mokslo krypties tarybos posėdyje, kuris ivyks 2005 m. vasario 4 d., 12 val. Kauno technologijos universitete, Disertacijų gynimo saleje (K. Donelaičio g. 73 – 403, Kaunas).

Adresas: K. Donelaičio g. 73, 44029 Kaunas, Lietuva.
Tel.: (8–37) 300042, faksas: (8–37) 324144, el. paštas:mok.grupe@ktu.lt.

Disertacijos santrauka išsiusta 2005 m sausio 4 d.

Su disertacija galima susipažinti Kauno technologijos universiteto bibliotekoje
(K. Donelaičio g. 20, 44239 Kaunas).

Theme significance:

Business process modelling as an integral part of Enterprise modelling has become an essential part of information system (IS) development process. However, the integration of Enterprise modelling techniques into the information systems engineering is still not sufficient.

Presently computerized IS engineering is being developed, new methods of IS engineering are being researched. However, a typical feature of modern computerized IS engineering methods is their empirical nature, because the project models repository of CASE system is composed on the basis of enterprise problem domain. This knowledge is not verified through formalized criteria. The problem domain knowledge acquisition process relies heavily on the analyst and user; therefore it is not clear whether the knowledge about this problem domain is adequate. The human plays the pivotal role in problem domain knowledge acquisition process, and few formalized methods of knowledge acquisition control are taken into consideration.

Another typical characteristics (disadvantage) of present–day computerized IS engineering methods should be also mentioned: design stage models are made in an interactive mode (the designer and CASE tool participate), and only several IS design stage models are partly generated because of an unsufficient enterprise model composition. Currently, in the first stage of IS designing cycle, CASE systems generate a diagram of functional hierarchy according to problem domain model (DFD or WFM), while in the last stage of IS designing cycle, program code (prototype of user interface) is generated according to class model and data base specification. Other project models are formed interactively, i.e. designer, analyst and programmer create IS project models through analyzing models, designed in earlier stages. Therefore, gaps of IS engineering process occur due to the human factor. These gaps mean, that the project model is formed in an interactive way (when the human participates), but not in an algorithmic one. This determines the incompatibility of IS project models and the incoherence of IS designing process, because in IS engineering process human is overloaded. Many mistakes can be avoided when applying formalized (algorithmic) methods of knowledge analysis, control and generating.

In the dissertation the stage of user requirements acquisition, analysis and specification is analyzed. Usually user requirements acquisition process starts from the construction of the Use Case model. Such model is formed without examining the consumer as the main source of knowledge, according to formal or formalized criteria. The analyst performs problem domain knowledge analysis and composes Use Case model. The dissertation offers the solution to this problem by designing and then using enterprise knowledge repository of CASE systems.

Problem domain knowledge (which is examined through formalized criteria) should be stored in the enterprise knowledge repository of CASE tool and should be used to control knowledge of user and analyst also to verify IS project solutions. This repository is used for the generation of IS engineering design stage models too. The composition of enterprise model is regulated by formalized method based specification, which is called enterprise metamodel.

The formalized method, used in this dissertation, was created in Kaunas University of Technology, Department of Information Systems. This method is based on the Control Theory and was used as a background for enterprise metamodel designing process.

The core of scientific problem is creation of enterprise model based method of computerized IS engineering user requirements specification. CASE system, created on the basis of this method, expands enterprise knowledge repository. The expansion intellectualizes the stage of functional user requirements acquisition, analysis and specification. Formalized descriptions of method and engineering tools is depicted below:

- modified work flow model based enterprise modeling method;
- generation algorithms of Use Case models (UCM).

This dissertation presents method description also engineering solutions through integrating model based CASE tool knowledge repository into computerized IS engineering stage of user requirements acquisition, analysis and specification.

The scope of the research involves the following IS engineering stages:

- User requirements acquisition and problem domain analysis on the basis of knowledge base;
- User requirements specification, i.e. generation of user requirements specification on the basis of knowledge base.

Control theory based formalized method of business process modeling, used in the dissertation, defines formalized criteria for both enterprise modeling and the control of user requirements.

The object of the research. Computerized functional user requirements acquisition process, based on problem domain modelling, and user requirements specification, based on enterprise model, which is stored in repository of the CASE system.

The purpose of the research is to create enterprise model design method and enterprise model based method of functional user requirements acquisition, analysis and specification.

During the research the following **tasks** are approached:

- to make an analysis of enterprise modeling standards and enterprise models used in CASE systems in order to single out essential disadvantages of IS engineering user requirements acquisition, analysis and specification also to define the composition of CASE system enterprise knowledge repository;
- to create method of enterprise knowledge acquisition into the CASE system repository, based on work flow models;
- to create the user requirements specification method (algorithms) based on enterprise knowledge repository;
- to implement the prototype of enterprise knowledge acquisition into the CASE system repository method, based on work flow models;
- to create user requirements specification algorithms, generating Use Case models on the basis of enterprise knowledge.

Research methods: structural enterprise modeling methods, object oriented IS modeling (UML).

Scientific novelty of the dissertation.

The dissertation presents enterprise model based computerized functional user requirements acquisition, analysis and specification method, which covers the following aspects:

- work flow model based method of capturing and analyzing computerized problem domain knowledge, using the enterprise metamodel;
- enterprise model based method of specification of functional user requirements models.

This dissertation deals with the formalized enterprise metamodel, which restricts the development of the enterprise model of a particular problem domain. Enterprise metamodel based enterprise model is called formalized enterprise model. The enterprise knowledge repository of CASE system is created on the basis of formalized enterprise metamodel specification.

The created work flow model based method ensures that knowledge acquired to enterprise knowledge repository is sufficient to generate Use Case models of various types. The dissertation presents application of enterprise knowledge repository of CASE systems in generation of Use Case models. The peculiarity of the work is that modified work flow models define business process and business function as qualitatively different enterprise components: the process parallels the material while the function parallels informational ones. Interaction between enterprise process and function is an essential component of enterprise model, because it forms informational feedback loop. Thus metamodel was chosen. In this metamodel theoretically correct controlling process is implemented, which creates the feedback loop between controlled object and controlling function.

Elimination algorithms of process and function logical gaps are created in order to identify and eliminate logical gaps in user requirements identification, analysis and specification stage. In this dissertation Use Case model generating algorithms are developed according to several user criteria (enterprise function, enterprise process, actor and enterprise subgoal). These algorithms generate UCM variations, which are corrected by designer.

Practical significance of the work. The work substantiates enterprise knowledge based engineering stage of functional user requirements, gives engineering tools and algorithms for user requirements acquisition, analysis and specification. The work also aims to complement the composition of CASE system repository with the enterprise knowledge repository. The basic elements of this repository are enterprise metamodel and enterprise model. The advantage of such CASE systems, complemented by enterprise knowledge repository, is that computerized problem domain knowledge (stored in it) is an extra source for creating project IS models. This expands functional capabilities of CASE system, i.e. capabilities of generating and examining project models.

The dissertation suggests new engineering tools (modified work flow models) for problem domain knowledge acquisition to CASE system enterprise knowledge repository: business process work flow model, process work flow model, functional work flow model and work flow model of functional composition.

Publications and Approbation of the Research Results. The author of this dissertation published 16 scientific publications. 1 article is published in ISI (Institute of Scientific Information) indexed journal, 3 articles in the journals, included in the list certified by the department of Science and Studies of Lithuania, 3 articles in the proceedings of international conferencies abroad and 9 articles in the proceedings of Lithuanian conferencies.

Structure and Volume of the Dissertation. The dissertation contains the introduction, 3 chapters, conclusions, lists of the author's publications, list of references and 19 appendixes. The total volume of the dissertation is 174 pages, including 57 tables and 108 pictures. The list of references contains 91 sources. The structure of the work reflects object, goals and tasks.

Introduction defines the state and problems of modern CASE systems. Modern computerized IS engineering and CASE tools involve enterprise models and IS designing models. Enterprise models, which are analyzed in scientific literature and applied in modern CASE systems, do not ensure continuous designing process, a logical relation among models of IS engineering stages. This is because enterprise models allow a lot of freedom of human solutions. The structure of CASE system enterprise knowledge repository is substantiated by formalized enterprise model; its peculiarity is enterprise metamodel, designed on the basis of the Control Theory. It should be noted that the application of enterprise metamodels in IS engineering is not given enough attention in IS engineering literature. There are some internationally certified enterprise models, but the majority of them are applied in business re–engineering and not computerized IS engineering.

In IS engineering, enterprise model is applied as the basic structure of knowledge, which is essential to generate project models. The traditional stage of (IS engineering) user requirements acquisition, analysis and specification is implemented by the customer and system analyst. In this stage, enterprise knowledge repository performs the role of extra information source, which acquires and analyses computerized problem domain knowledge and user requirements.

Chapter One provides the survey of IS engineering trends, the analysis of the role of CASE tool components during IS engineering process, and the theoretical models of IS engineering development life cycles. This chapter deals with the role and development tendencies of computerized problem domain modeling in IS engineering. Traditional and knowledge–based processes of computerized requirements engineering are analyzed, too. The advantages of knowledge–based computerized IS engineering user requirements acquisition, analysis and specification stage, and the disadvantages of traditional computerized IS engineering stage are distinguished. This chapter sums up the major requirements that are imposed on user requirements specification in enterprise modeling, as well as the methods of user requirements acquisition and specification. In order to define the composition of CASE tool enterprise knowledge repository, the comparative analysis of the main enterprise modeling standards (such as ENV 12204, ENV 40003, UEML and WFMC TC00–1003) is performed in terms of composition, function, information, resources and organization. The analysis of the major enterprise modeling methods is performed in aspect of enterprise processes and function interaction. Moreover, the research of the stage of user requirements acquisition, analysis and specification, based on enterprise knowledge repository is done, and the advantages of this stage discussed.

Computerized IS engineering is becoming knowledge–based IS engineering. Today computerized IS engineering methods and tools are going through a new phase of development – they are integrated with enterprise modeling methods and tools. Thus an field of method engineering has been formed to study and create advanced CASE methods. The issue of enterprise knowledge design is topical in the field of CASE methods development. One of the solutions to this problem is MDA (Model Driven Architecture), which is introduced by OMG. IS engineering methods that are being formed aim to integrate the knowledge about problem domain processes, functions and business rules and to apply them so that IS engineering process can be intellectualized. The results of IS engineering and enterprise modeling analysis indicate that IS engineering requirements stimulate the integration of such scientific fields and technologies as methods of IS engineering, enterprise modeling, enterprise re–engineering, decision support systems and others.

After problem domain knowledge (which is necessary for IS engineering) is acquired into enterprise model of CASE system repository, IS engineering project models can be generated interactively. The analyst, the enterprise knowledge repository of CASE system and the designer participate in this process. The basic component of knowledge–based CASE tool is enterprise knowledge repository, which intellectualizes the process of information systems design. Knowledge–based IS engineering is the process, in which the following equivalent partners – knowledge resources participate: the user, the analyst, enterprise knowledge repository of CASE tool and the designer. Information system in traditional computerized IS engineering is created empirically, beginning with user requirements acquisition, analysis and specification. In addition, modern scientific management trends, such as knowledge–based enterprise management, knowledge management and knowledge–based IS engineering, begin using IS engineering tools – CASE systems. In this way some time is saved and solutions are improved in quality – enterprise model contains verified knowledge about problem domain.

Chapter Two presents the principles of CASE tool Enterprise Knowledge Repository formation and application in IS engineering user requirements acquisition, analysis and specification stage. It also presents the composition of knowledge based CASE tool as well as its role in IS engineering life cycle during user functional requirements acquisition, analysis and specification stage. Enterprise Knowledge Repository expands the architecture of modern CASE tools through the performance of the main function of computerized problem domain knowledge storage. Figure 1 demonstrates the architecture of the CASE system, enhanced by the Enterprise Knowledge Repository. The Enterprise Knowledge Repository of the CASE system consists of two parts: the Enterprise Meta–Model (EMM) and the Enterprise Model (EM). The EMM is a generic level model; an EM includes the partial and particular level models in accordance with GERAM. The EMM regulates the formation order of the EM. The EMM defines the composition of computerized problem domain knowledge, which is necessary for creating project models and generating programmed code. The EM of the computerized problem domain is formed by the user and analyst according to EMM constraints.

The Enterprise Knowledge Base of the CASE system is supposed to be the third active source of Enterprise knowledge (together with the Analyst and User) for information systems engineering. In this enhanced environment of information system development the EMM is a source of pre–defined knowledge, and is used to control the process of business domain knowledge acquisition and analysis. It is also used to control the construction of an EM for a particular problem domain. As the main enterprise knowledge structure, the EMM controls IS engineering process so that the possibility of logical errors and gaps can be reduced. For instance, the designer is informed about impossible interaction of certain Functions and Processes as well as impossible participation of separate Organizational Units or Actors in it. However, if the need for these elements is identified, Enterprise Knowledge Repository is updated accordingly.

Knowledge–based IS development supposes that all stages of IS development life cycle are supported by the Knowledge Repository of CASE system. Together with appropriate algorithms, the Knowledge Repository of the CASE system assures consistency among the IS analysis and design models, gives new possibilities for verification and validation of IS development life cycle stages. During the design stage of IS engineering life cycle, it is advisable to generate conceptual and detailed level diagrams from the Enterprise Knowledge Repository of CASE tool for e.g. Entity Relationship diagram, Class Model, etc. It is also advisable to generate a fully functioning programmed code from Knowledge Base in order to intellectualize IS engineering process. Moreover, Enterprise Knowledge Repository of CASE system can be used to simulate and improve business processes in the enterprise. The Enterprise Knowledge Base of the CASE system can be also used to verify business domain knowledge, which acquired by analyst and used to construct a particular Enterprise Model. This is done by verifying constructed Enterprise Model against the predefined knowledge structure of the Enterprise Meta Model.

The architecture of CASE system with Enterprise Knowledge Repository is presented in Figure 1.

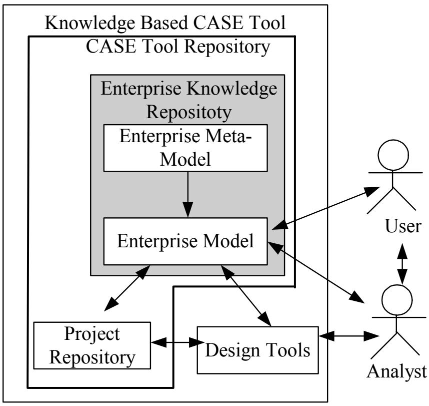

Figure 1. The architecture of CASE system with Enterprise Knowledge Repository

The conceptual scheme of Enterprise Metamodel is shown in Figure 2. The basic feature of the Enterprise Metamodel is the interaction of *Process* and *Function*. A *Process* is a partially ordered set of steps, which can be executed to achieve the desired material end–result. *Process* consumes material resources and produces some material output, i.e. a product. *Processes* are triggered by one or more *Event* occurrences. *Function* is a work flow element, which controls processes. A *Function* is a complex construct. The structure of the *Function* is defined on the basis of the formal definition of management function. At least one *Function* controls each *Process*, transforming material input flow into material output flow. *Process* supplies enterprise *Function* with processing state attributes (*Process_Output*), which are transformed into the input attributes *(IP_Input)* of Data Processing and Decision Making (*IP*) functional component during interpretation. *Interpretation* is a set of rules, intended to transform *Process_Output* information flow into *IP_Input* information flow, which is prepared for processing in *IP* functional component. *Interpretation* is an essential component of *Function* because the format of *Process_Output* may be inconsistent with the certified data format of *IP_Input*. *IP* is a component of *Function*, which performs Informational Processing and Decision Making operations. *IP* functional part transforms *IP_Input* information flow into *IP_Output* information flow. *Realization* is component of *Function*, performing an action contrary to *Interpretation*. *Realization* transforms *IP_Output* information flow into *Process_Input* information flow. The internal structure of *Interpretation*, *IP* and *Realization* components of *Function* is based on *Business Rules*. *Function* accomplishes at least one organizational *Goal* or its subgoal. *Process* and *Function* are performed by an enterprise *Actor*. Not only a human or organizational unit, but also software or device can perform *Function* or *Process*. Material processing is stimulated by an environmentally initiated *Event*. *Environment* initiates *Event* and influences enterprise *Goals*. The conceptual scheme of Enterprise Metamodel is shown in Figure 2.

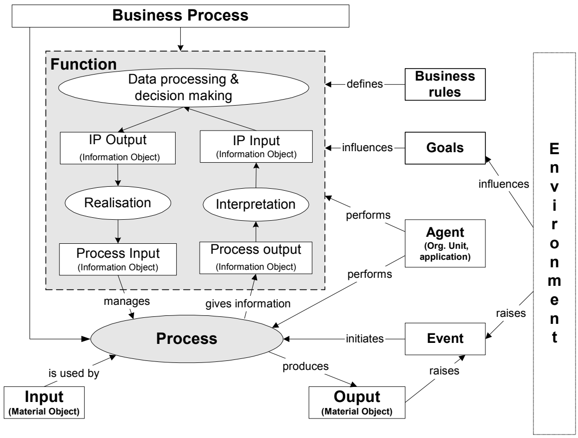

Figure 2. Conceptual scheme of Enterprise Metamodel.

The class model of the Enterprise Metamodel is presented in Figure 3. Its core consists of twenty four classes, the basic of which are *Process*, *Function* and *Actor*. Such classes as *Process*, *Function*, *Actor* and *Goals* may have an internal hierarchical structure, which is demonstrated by aggregation relations. The class *Process* according to aggregation relationship is related to the class *Material_Flow,* while class *Material_flow* is related to such classes as *Material_Input_Flow* and *Material_Output_Flow* according to generalization relationship. Class *Process,* according to association relationships is related to the classes of *Function*, *Actor* and *Event*. According to aggregation relationship the class *Function* is related to classes *Information_Flow, Information_Activity, Interpretation, Realization, Data processing and Decision making (IP)*. All these relationships define the internal structure of *Function*. The class *Information_Flow* according to generalization relationships is related to classes *Process_Output*, *IP_Output* also *Process_Input*, while class *Information_Activity* in the same relationships is related to classes *Interpretation*, *IP* as well as *Realization*. The class *Function* according to association relationship is related to classes *Process, Actor, Goals* and *Business Rules*. According to generalization relationship class *Business rules* (BR) is related to classes *BR_Interpretation, BR_IP* also *BR_Realization*. The class *Actor* is related to classes *Function_Actor* and *Process_Actor* considering the same relationship.

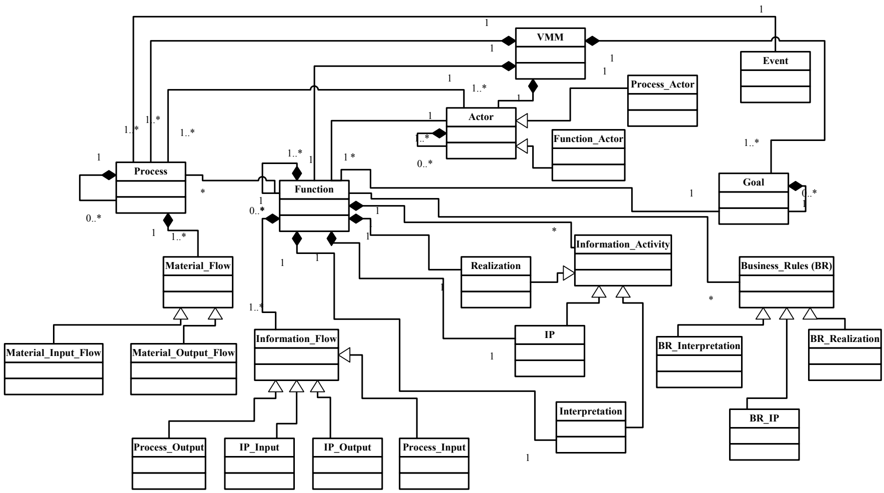

Figure 3. Class model of Enterprise Metamodel.

Problem domain knowledge acquisition

The acquisition of user requirements is the initial stage of traditional IS development life cycle, together with enterprise modelling. Most of user requirements acquisition techniques are based on empirical information acquired provided by the user (business domain expert) and systemized by the analyst. Therefore, the user and the analyst are two sources of information in traditional IS engineering. Problems occur when empirically acquired information (requirements) has to be verified and validated. This chapter deals with the major principles of a knowledge–based approach to IS engineering and the Enterprise Knowledge Repository of CASE system (containing the EMM and particular Enterprise model) is considered to be the third source of information for IS engineering – both for user requirements analysis and specification and for other IS development life cycle stages.

In the stage of problem domain knowledge acquisition 6 types of modified work flow models are created: Work Flow Model of Business Processes (VP_WFM), Work Flow Model of Processes (P_WFM), Work Flow Model of Functions (F_WFM), Work Flow Model of Processes without Gaps, Work Flow Model of Functions without Gaps, Work Flow Model of Functional Composition (FS_WFM).

In order to create such models and transform knowledge into the enterprise model, algorithms of four types are developed: the algorithm separating VP_WFM into P_WFM and F_WFM, the algorithm which identifies and eliminates logical gaps in P_WFM, the algorithm which identifies and eliminates logical gaps in F_WFM and the algorithm which determines the composition of a particular function according to the internal structure of Enterprise Metamodel. The stage of work flow model based computerized problem domain knowledge acquisition and analysis is given in Figure 4.

Problem domain knowledge, acquired in VP_WFM, is transformed into P_WFM and F_WFM when separation algorithm is performed. Yet, in the transformation process logical gaps may occur. *A logical gap* is a semantic discontinuity between the elements of the problem domain model (for instance, workflow model). Logical gaps in the P_WFM and F_WFM models are identified by the algorithms of the P_WFM and F_WFM analysis and eliminated by the analyst. The application of these algorithms requires an additional analysis of the problem domain. Logical gaps can be eliminated in two ways:

- New elements of the P_WFM (*Material Flow, Process, Actor*) and F_WFM (*Information flow, Activity, Actor*) can be added by the Analyst as a result of additional analysis of the problem domain, performed by the User and Analyst;
- Some elements of the P_WFM (*MaterialFlow, Process, Actor*) and F_WFM (*Information flow, Activity, Actor*) can be excluded by the Analyst during the semantic analysis of the workflow models, performed by the User and Analyst.

The result of logical gaps elimination algorithms are P_WFM and F_WFM without logical gaps. In such eliminating process VP_WFM is also updated with knowledge about lacking processes, activities, information or material flows of a particular problem domain. This process is called the first quality assuring cycle of computerized problem domain knowledge.

The algorithm defining functional composition is performed at the next step of the stage of work flow model based computerized problem domain knowledge acquisition and analysis. During this process, completeness of functional composition, which controls each process, is verified. (i.e. it is verified weather F_WFM functional elements – activities, controlling each process, are specified). The lacking activities are identified on the basis of enterprise metamodel composition. The process of functional composition algorithm performance indicates activities, which exist in the enterprise problem domain, but are not specified in F_WFM. Information flows, which relate these activities, are also indicated in this process. Material processes, information activities, material and information flows (which are indicated during performance of functional composition algorithm) complement VP_WFM by new elements. This process is called the second quality assuring cycle of problem domain knowledge acquisition process. The result of functional composition defining algorithm is FS_WFM. This model specifies the internal composition of particular material process controlling function, i.e. F_WFM model activities (which are attributed to *Interpretation*, *Information Processing and Decision Making* and *Realization*) and their relating information flows.

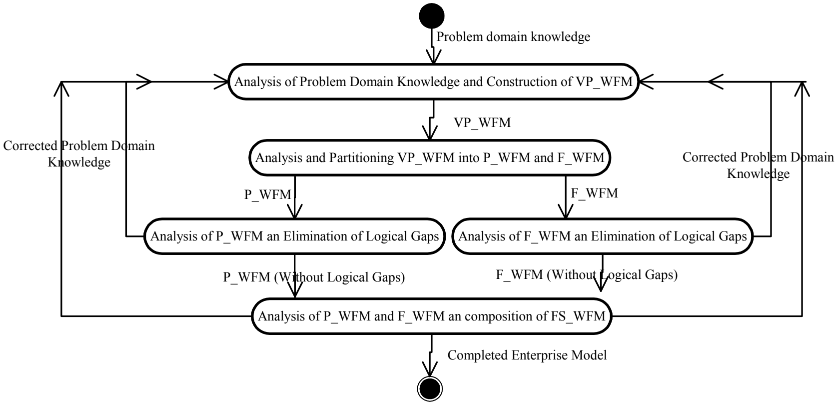

Figure 4. The stage of work flow model based computerized problem domain knowledge acquisition and analysis.

Work Flow Model of Business Processes

Primary knowledge about computerized problem domain is acquired to VP_WFM. The user and analyst give knowledge about computerized problem domain when designing VP_WFM. VP_WFM is used to acquire knowledge about material processes, enterprise activities, informational and material flows and actors of a particular problem domain. Work Flow Model of Business Processes is designed on the basis of composition of traditional work flow model. The main components of traditional work flow model (Provision "Work Bench" etc.) are *Actors*, *Activities* and *Flows*. In graphical notation activities and flows are signed by symbols without reference what nature (material or informational one) business process and flow belong to, i.e. the signing element is the same of informational and material flow as well as informational activity and material process. In order to make the process of problem domain knowledge acquisition more effective it is advisable to modify traditional work flow model by establishing flows of two types: material and informational. The modified work flow model is called Work Flow Model of Business Processes (VP_WFM).

VP_WFM is applied to problem domain acquisition process because it is sufficient for capturing knowledge about business processes, actors, material and information flows. It is simple and easily mastered too. Metamodel of Work Flow Model of Business Processes is presented in Figure 5.

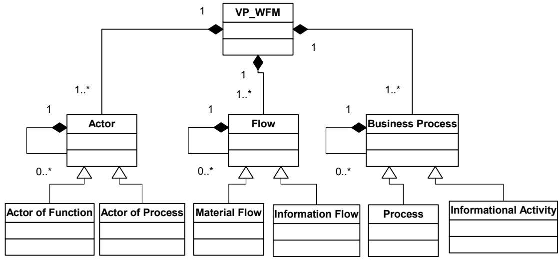

Figure 5. Metamodel of Work Flow Model of Business Processes

Two types of VP_WFM flows are necessary for VP_WFM separation into P_WFM and F_WFM during the next stage of problem domain knowledge acquisition and analysis. Each business process of VP_WFM, except initial and final ones, has material and (or) informational input and output. Business process can be of either material or informational nature. Business process which is related to material flows is defined as business process of material nature, while business process that is related to informational flows is defined as business process of informational nature. Business processes of material nature and their material input or (and) output flows, are specified as material processes with these flows in P_WFM when separation algorithm is performed. Business processes of informational nature and their informational input or (and) output flows, are specified as informational activities with these flows in F_WFM when separation algorithm is performed.

Business process of VP_WFM is defined as the sequence of organizational actions, which transform inputs into outputs. Material flow is a material input and (or) output of business process, supplying material resources necessary to perform the process. Material input (output) of business process is not a mandatory element of each business process. Information flow is informational input and (or) output of business process, intended to control it. VP_WFM actors are human, group of humans or organizational unit, which perform business process and are responsible for its successful performance. The prototype of VP_WFM modeling tool is realized in MS "VISIO 2000" CASE tool and MS "ACCESS 2000" data base management system. VP_WFM is formed in two main steps when applying the suggested prototype. During the first step the diagram of VP_WFM is charted by graphical editor, realized in MS "VISIO 2000" environment, and to its component parts (business process, flow and actor) are given attributes. During the second step VP_WFM model is exported to MS "ACCESS 2000" data base.

Work Flow Model of Processes

Work Flow Model of Processes (P_WFM) specifies material processes of organization (they are singled out of VP_WFM, and have material inputs and (or) outputs) and actors, who implement them. A *Process* is a partially ordered set of steps, which can be executed to achieve some desired material end–result. A *Process* consumes material resources (it is an input of the process) and produces some material output – production. The components of the process are subprocesses, tasks, operations. Definitions of material flow and actor of P_WFM are analogical to VP_WFM. The metamodel of P_WFM is presented in Figure 6.

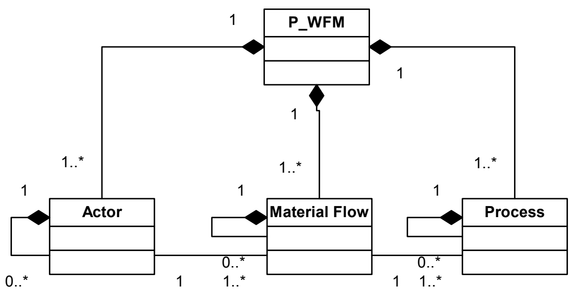

Figure 6. Metamodel of Work Flow Model of Process.

Work Flow Model of Functions

Informational activities, information flows and actors are the components of Work Flow Model of Functions (F_WFM). Informational activity is function, or its component, which processes information flows when changing information input into information output. Each material process is controlled by at least one function, which consists of informational activities and information flows, linking that activities. Material process is fully controlled by business function, while activity controls this process partly. Definitions of information flow and actor in F_WFM are analogical to VP_WFM. F_WFM metamodel is presented in Figure 7.

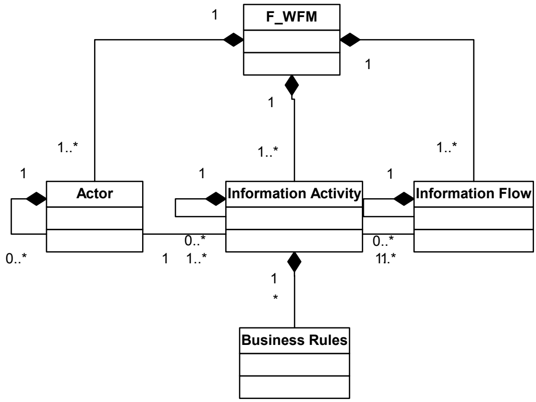

Figure 7. Metamodel of Work Flow Model of Functions.

Work Flow Model of Functional Composition

The result of functional composition verification algorithm is Work Flow Model of Functional Composition (FS_WFM). Elements of F_WFM are specified in FS_WFM as components of the functional composition, defined in enterprise metamodel. FS_WFM specifies only one function, which controls one or more processes, specified in P_WFM. In accordance with the internal structure of function which is defined by enterprise metamodel, there are three types of F_WFM activities: Information activity of interpretation, information activity of processing and decision making (IP), Information activity of realization. Each F_WFM activity can correspond to one of the above mentioned component parts of functions. Algorithm, which defines functional composition, determines what part of function activities belong to and what material process do they control in F_WFM. Each activity of F_WFM, specified in FS_WFM, can be analogical component (Interpretation, IP or Realization) of several FS_WFM. FS_WFM metamodel is presented in Figure 8.

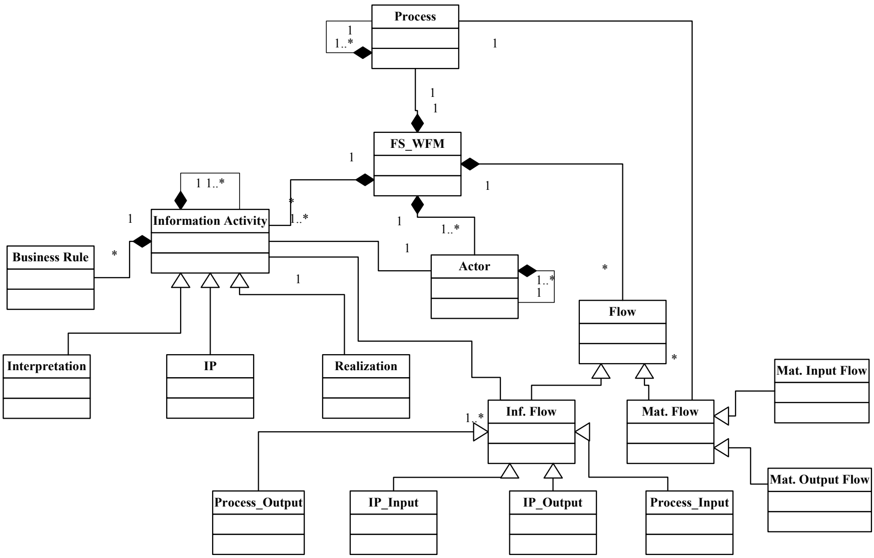

Figure 8. Metamodel of Work Flow Model of Functional Composition (FS_WFM).

Comparison of traditional and modified work flow models in composition aspect is presented in Table 1

Table 1. Comparison of traditional and modified work flow models in composition aspect

	Traditional WFM	VP_ WFM	P_ WFM	F_ WFM	P_WFM (without gaps)	F_WFM (without gaps)	FS_WF M
Business Process	+ (not detailed)	+	–	–	–	–	–
Activity		–	–	+	–	+	+
Process		–	+	–	+	–	+
Material Flow	+ (not detailed)	+	+	–	+	–	+
Informational Flow		+	–	+	–	+	+
Actor	+	+	+	+	+	+	+
Activity type	–	–	–	–	–	–	+
Possibility of Logical Gaps	+	+	+	+	–	–	–

Theoretical Substantiation of Separation VP_WFM

The initial verification stage of problem domain knowledge completeness is a process, which separates VP_WFM into P_WFM and F_WFM. When VP_WFM is separated into P_WFM and F_WFM, wrong specified problem domain knowledge (informational gaps) is identified. This process of separation is necessary, because after eliminating algorithm is performed, the specified initial knowledge is used as functional composition defining algorithm input. During VP_WFM separation into P_WFM and F_WFM the main three rules are applied. First rule – if business process input and (or) output (specified in VP_WFM) are information flows, this business process will be specified as informational activity with input and (or) output flows in F_WFM. Second rule – if business process input and (or) output (specified in VP_WFM) are material flows, this business process will be specified as process with material input and (or) output flows in P_WFM. Third rule – business process actor is specified as an actor of F_WFM activity or P_WFM process, which was singled out of business process VP_WFM, performed by him.

Figure 9 gives an example of VP_WFM_p components, depicted in P_WFM_p and F_WFM_p diagrams. VP_WFM_p specifies business processes VP1, VP2 and VP3, information flows I1 and I2 and material flow M1. Using the first rule of VP_WFM_p separation, business process VP2 and information input I1 are specified in F_WFM_p as activity F2, which has information input I1. Business process VP2 actor V1 according to the third separation rule is depicted as actor V1 in F_WFM_p. According the second separation rule, business process VP3 and material input M1 are depicted as material process P2 with material input M1 in P_WFM_p. Business process VP3 and its information input I2 are depicted as activity F3 with information input I2. Business process VP1 is related with information outputs I1 and I2, also material output M1. According to the first VP_WFM_p separation rule, VP1 is depicted as activity F1 with information outputs I1 and I2 as well as according to the second rule, VP1 is depicted as material process P1 related with material output M1.

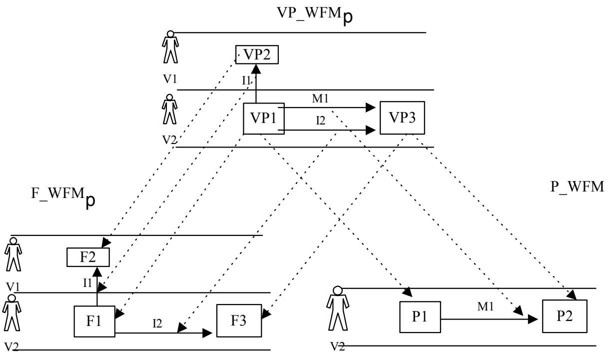

Figure 9. Example of VP_WFM_p separation into P_WFM_p and F_WFM_p.

Each process, except the initial and final ones, must have material input and output flows. Analogical requirement obtains regarding activity, simply its input and output flows are informational ones. If this condition is not complied with, F_WFM or P_WFM has logical gaps, that are eliminated when process and function gaps eliminating algorithms are performed. Algorithm of VP_WFM separation into F_WFM and P_WFM is realized by MS “VISIO 2000” CASE tool and MS “ACCESS 2000” data base management system.

Elimination Algorithm of Process Work Flow Model Gaps.

It is likely that on separating VP_WFM into F_WFM and P_WFM logical gaps may be identified in newly created F_WFM and P_WFM. A logical gap is a semantic discontinuity among the elements of the workflow model. The logical gaps appear when problem domain knowledge is acquired incompletely. On purpose to eliminate gaps of P_WFM, detecting and eliminating algorithm is applied. Without reference to elimination method, P_WFM is complemented by non–existing, but wrongly or hardly specified knowledge (process, material flow and actor). Logical gaps of P_WFM are identified during the analysis of input and output flows of each material process. A logical gap in the P_WFM and F_WFM is identified if some Process or Activity is not related to input or output flow. Except the first and the last processes of the workflow model each Process of the P_WFM must be related to at least one input material flow and one output material flow, in the same as each Activity of F_WFM must be related to at least one input information flow and one output information flow. On purpose to eliminate logical gaps of P_WFM, the prototype of informational system, eliminating P_WFM gaps, is used: it was created by MS "VISIO 2000" CASE tool and MS "ACCESS 2000" data base management system.

F_WFM logical gaps elimination algorithm is analogical to P_WFM logical gaps elimination algorithm. The main difference is that all actions is performed with F_WFM activities and informational flows, but not with P_WFM processes and material flows.

Table 2 presents the FS_WFM components of function, which are defined according to activities input and output flows existing in F_WFM. According to the types of informational input and output flows, three types of the following activities can be distinguished: *Interpretation*, *IP* and *Realization*.

Table 2. Possible combinations of FS_WFM activities input and output.

Type of Activity Output / **Type of Activity Input**	*Process Output*	*IP Input*	*IP Output*	*Process Input*
Process Output	Impossible	*Interpretation*	*Interpretation* and *IP*	*Interpretation IP* and *Realization*
IP Input	Impossible	Impossible	*IP*	*IP* and *Realization*
IP Output	Impossible	Impossible	Impossible	*Realization*
Process Input	Impossible	Impossible	Impossible	Impossible

If input and output of FS_WFM activity are information flow "*Process Output*", impossible type of activity is identified. Activities of FS_WFM, according to composition of enterprise metamodel, cannot have informational input and output flows of the same type. Activities, which have information input and output flows *("Process Output",* "*IP Input*", "*IP Output*", "*Process Input*") of analogical type, can exist neither. If activity input is "*Process Output*" and output is "*IP Input*", the activity will be component of function called *Interpretation*. *Interpretation* is set of rules, intended to transform information flow "*Process Output*" into "*IP Input*", which is prepared for *IP* processing. *Interpretation* is a necessary component of function, because "*Process Output*" information flow can mismatch data format, determined for functional IP element input "*IP Input*". If activity input is "*IP Input*" and output is "*IP Output*", the activity is *IP* component of function. *IP* is functional component, which is mainly intended to control process of information processing and decision making. If activity input is "*IP Output*" and output is "*Process Input*", the activity is part of function called *Realization*. *Realization* is functional part, performing process, which is contrary to interpretation. *Realization* transforms "*IP Output*" data (processed in *IP* stage) into "*Process Input*" format (suitable to direct process control).

There are some cases when F_WFM activities defines several component parts of function, according to activity input and output flows. If activity input is "*Process Output*" and output is "*IP Output*", the activity will have such functional components as

IP and *Interpretation* as well as information flow "*IP Input*" (which links *IP* and *Interpretation*). If activity input is "*Process Output*" and output is "*Process Input*", activity will consist not only of *Interpretation*, *IP* and *Realization* but also "*IP Input*" (which link *Interpretation* and *IP*) and "*IP Output*" (which link *IP* and *Realization*). Such composition indicates that this activity is function.

Activity input "*IP Input*" indicates two possible types of outputs: "*IP Output*" and "*Process Input*", while enterprise output "*Process Input*" indicates activities *IP* and *Realization* as well as information flow "*IP Output*" (which links *IP* and *Realization*). Activity input "*Process Input*" and output "*Process Output*" signal an error in F_WFM, thus such type of activity is impossible.

Chapter three. Here principles and algorithms of enterprise knowledge based Use Case model generation are defined. Also the prototype of IS, which is applied in user requirements acquisition, analysis and specification stage, is presented.

IS engineering life cycle begins at problem domain analysis, user requirements analysis and specification. In problem domain analysis of object oriented IS engineering, activity diagram and Use Case model can be applied. Authors translate the term "Use Case Model" into Lithuanian (as model of engineering tasks) rather differently, therefore this term will be called UCM (UML 1.4, 2000). Object oriented IS engineering, in many CASE tools, begins at UCM. Usually, UCM develop system analyst, who analyzes problem domain as well as functional and non–functional user requirements, intended to create informational system. In this way, UCM becomes a basic model for problem domain knowledge analysis and user requirements specification.

UML tools are insufficient for enterprise modeling, intended to develop process of information systems engineering. The information, acquired during problem domain analysis process, should be arranged like knowledge structure, which allows to generate IS project models (UCM as user requirements specification component, Class Model etc.). Such knowledge structure is called Enterprise Model and is stored in repository of CASE system.

Formalized Description of Use Case Model

There are different techniques of user requirements acquisition and specification: models of UML, user requirements specification templates (Volere) and other (Loucopoulos, 1995). The *Use Case* model (UCM) is a popular UML model aimed for user requirements specification. *Use Case* models are used to show the relationships between the *Actors* that use the system and the *Use Cases* they use. A *Use Case* is a procedure by which external Actors can use the system. Taken together the *Use Cases* define the full functionality of the system from the user point of view and can be used as an initial point of the process of system development. Let us discuss the UCM generation principles on the base of the Enterprise model. The Enterprise model enables the knowledge–based generation of the UCM.

The *Use Case* meta–model comprises the following constructs: *Actor*, *Use Case* and *Relation* (there are three types of relations: *Association*, *Include* and *Extends*). An A*ctor* defines a coherent set of roles that users of an entity can play while interacting with the entity. The U*se Case* construct is used to define the behaviour of a system or other semantic entity without revealing the entity's internal structure. Each *Use Case* specifies a sequence of actions, including variants that the entity can perform, while interacting with actors of the entity. The construct *Association* can be refined as an *Information Flow* between the constructs *Actor* and *Use Case*. *Association* states that an instance of the *Use Case* and a user playing one of the roles of the *Actor* communicate. An *Include* relationship defines that a *Use Case* contains the behaviour defined in another *Use case*. An *Extend* relationship defines that instances of a *Use Case* may be augmented with some additional behaviour defined in an extending *Use Case*. Detailed UCM composition is specified in UML specification version 1.4 (UML 1.4, 2000). UCM metamodel, presented in this specification version, is depicted in Figure 10.

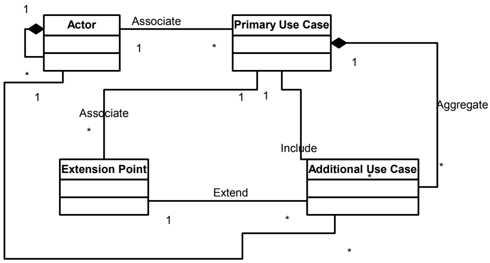

Figure 10. UCM metamodel.

Formalized Description of Enterprise Metamodel

Formalized description of enterprise and UCM metamodels is necessary to describe UCM generation algorithm. Formalized enterprise metamodel is described on the basis of abstract algebra (Malcev A., 1970).

M=<K, R>, M – stands for enterprise metamodel, K–set of classes, R–set of relations. K={K1, K2,…, K21}, R={r1, r2, r3}. Thus enterprise model M1 is described as follows: M1=<{K1, K2,...,K21}, {r1, r2, r3}>, where K1– class *Process*, K2– class *Function*, K3– class *Actor*, K4– class *Event*, K5– class *Goal*, K6– class *Material Flow*, K7– class *Input Material Flow*, K8– class *Output Material Flow*, K9– class *Information Flow*, K10– class *Interpretation*, K11– class *Information Processing and Decission Making* (IP), K12– class *Realization*, K13– class *Information Activity*, K14– class *Business Rule* (*BR*), K15– class *Interpretation Business Rules*, K16– class *Information Processing and Decission Making Business Rule (IP BR)*, K17– class *Realization Business Rules*, K18– class *Process Output*, K19– class *IP Input* , K20– class *IP Output*, K21– class *Process Input*, r1– *Aggregation* relation, r2– *Generalization* relation, r3– *Association*. The conceptual scheme of enterprise metamodel is shown in Figure 11.

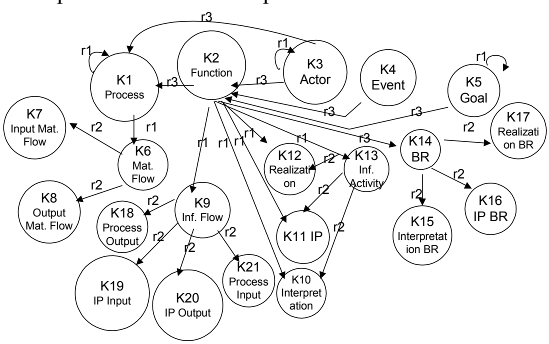

Figure 11. Conceptual scheme of enterprise metamodel.

Interrelations among components of enterprise metamodel are described below:

1. the classes *Process* (K1), *Actor* (K3) and *Goals* (K5) has internal hierarchical composition: (K1)r1(K1), (K3)r1(K3), (K5)r1(K5;

2. the class *Process* (K1) is related to the class *Material Flow* (K6) according to *Aggregation* relation (r1): (K1)r1(K6);

3. the class *Material Flow* (K6) is related to classes *Input Material Flow* (K7) and *Output Material Flow* (K8) according to *Generalization* relation (r2): (K6)r2(K7), (K6)r2(K8)

4. the class *Function* (K2) is related to classes *Information Flow* (K9), *Interpretation* (K10), *IP* (K11), *Realization* (K12) and *Information Activity* (K13) according to *Aggregation* relations (r1): (K2)r1(K9), (K2)r1(K10), (K2)r1(K11), (K2)r1(K12), (K2)r1(K13);

5. the class *Information Activity* (K13), is related to classes *Interpretation* (K10), *IP* (K11) and *Realization* (K12) according to *Generalization* relation (r2): (K13)r2(K10), (K13)r2(K11), (K13)r2(K12);

6. the class *Business Rules* (K14) is related to classes *Interpretation BR* (K15), *IP BR* (K16) and *Realization BR* (K17) according to *Generalization* relation (r2): (K14)r2(K15), (K14)r2(K16), (K14)r2(K17);

7. the class *Actor* (K3) is related to class *Process* (K1) according to *Association* relation (r3): (K3)r3(K1);

8. the class *Actor* (K3) is related to class *Function* (K2) according to *Association* relation (r3): (K3)r3(K2);

9. the class *Goal* (K5) is related to class *Function* (K2) according to *Association* relation (r3): (K5)r3(K2);

10. the class *Function* (K2) is related to class *Business Rules* (K14) according to *Association* relation (r3): (K2)r3(K14).

Principles of Enterprise Model Based UCM Generation

Traditionally, UCM is designed to computerize a particular task (information processing activity). Using identifiers of enterprise model elements, it is possible to single out knowledge, related to the task, and depict them according to UCM designing rules.

Notionally, UCM can be generated according to any enterprise model class (*Process*, *Function*, *Actor*, *Goal* etc.). During UCM generation for class *Process*, material processes (existing in problem domain) and their actors are specified in UCM. This type of UCM is called UCM of processes. The purpose of UCM of function is to specify the composition of function (its components) and actors. The components of Use Case model of Function are *Function*, *Information Activities*, *Information Flows* and *Actors*. UCM, generated for the class *Actor*, specifies actor's material processes, functions and informational activities. UCM generated for class *Goal* specifies *Functions*, related to organizational goals, and their components (informational activities). UCM, generated for such classes as *Material flow*, *Information flow*, *Information Activity* and *Business Rule*, are possible, but are not analyzed in detail in this dissertation. The most frequent variants of enterprise model based UCM generation are shown in table 3.

Table 3. Variants of enterprise model based UCM generation.

Types of UCM	The Enterprise Model elements, specified in UCM				
	Process	Function	Inf. Activity	Actor	Goal
UCM of Process	+	–	–	+	–
UCM of Function	–	+	+	+	–
UCM of Actor	+	+	+	+	–
UCM of Goal	–	+	–	+	+
UCM of Inf. Activity	–	+	+	+	–

Possibilities of enterprise model based UCM generation is shown in figure 12.

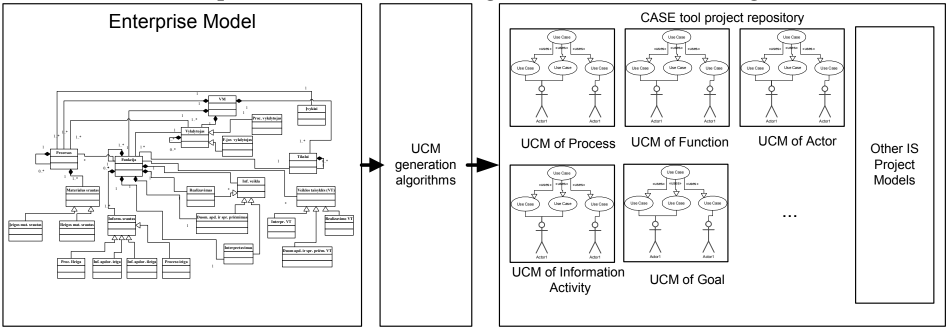

Figure 12. Possibilities of enterprise model based UCM generation.

UCM Generation for Particular Function of Enterprise Model

UCM, intended to particular function of Enterprise Metamodel, is called functional UCM. UCM metamodel of function consists of the classes *Actor*, primary Use Case *"PA (Function)*", additional Use Case *"PA (Activity)*" and "*Extension point*" as well as links of three types "*Associate*", "*Include*" and "*Generalize*".

Algebraic functional UCM F2 is described as system F2: F2=<{K22, K23,K24,K25}, {r1, r3} F2– stands for UCM of *Function*, K22– class *Actor* , K23– class *PA(Function)*, K24– class *Extension point*, K25– class *PA (Activity)*, r1– *Aggregation* relation, r3– *Association*. Relation *Generalization* is not used for generation process of UCM of *Function*. The relations among functional UCM F2 components are described as follows:

1. the class *Actor* has internal hierarchical composition: (K22)r1(K22);
2. the class *Actor* (K22) is related to classes *PA (Function)* (K23) and *PA (Activity)* (K25) according to *Association* relation (r3): (K22)r3(K23), (K22)r3(K25);
3. the class *PA (Function)* (K23) is related to class *Extension point* (K24) according to *Association* relation (r3): (K23)r3(K24);
4. the class *PA (Function)* (K23) is related to class *PA (Activity)* (K25) according to *Aggregation* relation (r1): (K23)r1(K25). This relation specifies *Information Activities* as component parts of the *Function*;
5. the class *PA (Function)* is related to class *PA (Activity)* (K25) according to *Association* relation (r3): (K23)r3(K25);

6. the class *Extension point* (K24) is related to class *PA (Activity)* (K25) according to *Association* relation (r3). (K24)r3(K25). Graphical scheme of functional UCM metamodel is presented in Figure 13.

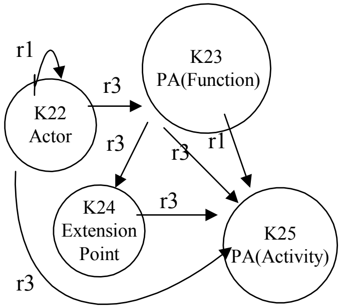

Figure 13. Graphical scheme of functional UCM metamodel.

Relation Between Enterprise Model and UCM of Function

Logical relation between enterprise model and UCM of function is determined on the basis of its class models analysis. Such enterprise model classes as *Function*, *Actor*, *Informational activity* and *Business rules* are necessary to generate UCM model of function. During the generation of UCM model of function, it is reflected into classes *PA (Function)*, *Actor*, *PA (Activity)* and *Extension point*. This can be formally defined by reflections of set (Alekna P. 1996) φ_1: K2→K23; φ_2:K3→K22; φ_3:K13→K25; φ_4:K14→K24. Names of sets correspond to class names.

Table 4. Reflections of enterprise model M1 elements to elements of UCM model F2 of function.

Components of EM			**Components of UCM of Function**		
The name of class	**Component of system M1**	**Reflection (→)**	**The name of class**	**Component of system F2**	**Formalized description**
Function	K2	φ_1	*PA (Function)*	K23	φ_1: K2→K23
Actor	K3	φ_2	*Actor*	K22	φ_2:K3→K22
Inf. Activity	K13	φ_3	*PA (Activity)*	K25	φ_3:K13→K25
Business Rules	K14	φ_4	*Extension point*	K24	φ_4:K14→K24

The reflections of Enterprise Model M1 elements to UCM F2 is shown in figure 14

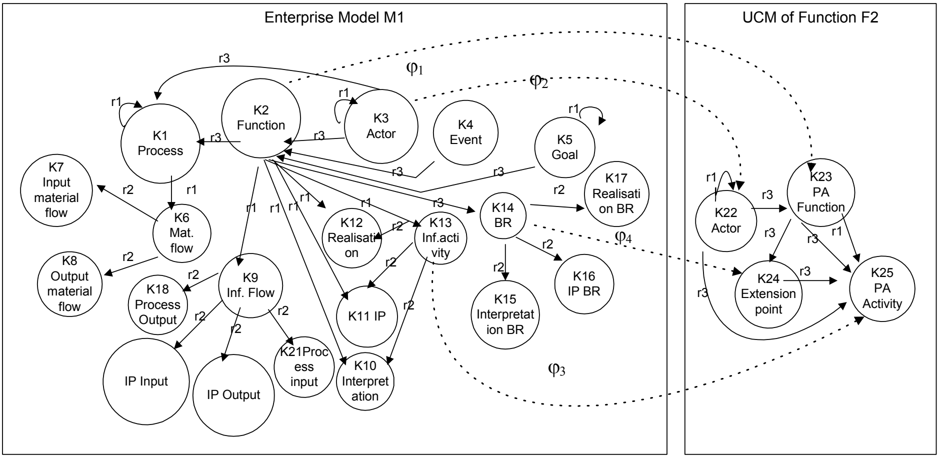

Figure 14. The reflections of Enterprise Model M1 elements to UCM F2.

Examples of UCM F2, generated according to these rules, is presented in Figure 15.

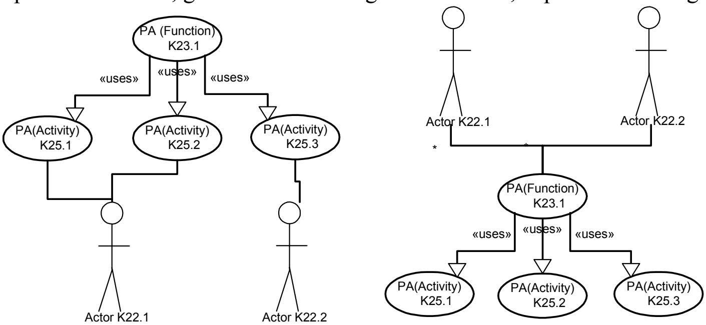

Figure 15. Examples of generated functional UCM.

UCM Generation for Process

The UCM, generated for particular *Process* of enterprise model, is called UCM of Process. The metamodel of UCM of Process consists of the classes *Actor*, the main Use Case *"PA Process"*, an extra Use Case "*PA Subprocess"* and *"Extension point"* as well as relations *Associate* and *Aggregate*. Process UCM can be generated for material process only.

Reflections of elements of enterprise model M1 to UCM P2 are presented in table 5.

Table 5. Reflections of elements of enterprise model M1 to UCM P2 elements.

Components of EM			**Components of UCM of Process**		
The name of class	**Component of system M1**	**Reflection (→)**	**The name of class**	**Component of system P2**	**Formalized description**
Process	K1	φ_5	*PA Process*	K27	φ_5: K1→K27
Actor	K3	φ_6	*Actor*	K26	φ_6:K3→K26
Business Rules	K14	φ_7	*Extension Point*	K28	φ_7:K14→K28
Process	K1	φ_8	*PA Subprocess*	K29	φ_8:K1→K29

Examples of UCM P2 is shown in Figure 16.

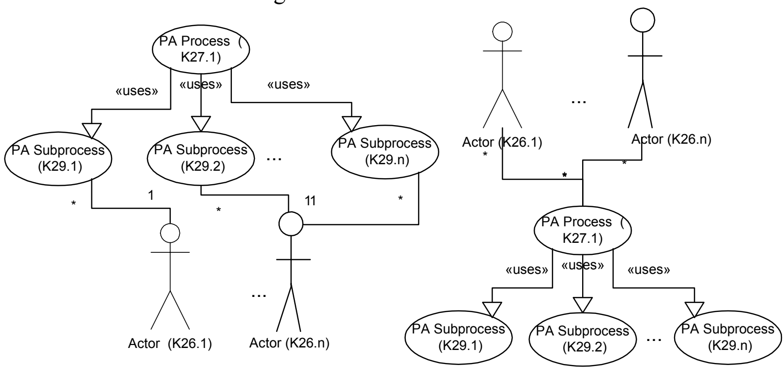

Figure 16. Examples of UCM P2

UCM Generation for an Actor

UCM generated for particular *Actor*, who is specified in enterprise model, is called UCM of *Actor*. UCM of *Actor* metamodel consists of the classes *Actor*, the main Use Case *PA Function (PA Process)*, an extra Use Case *PA Activity (PA Subprocess)* and *Extension Point* as well as relations *Associate* and *Aggregate*. UCM of *Actor* can be generated for each actor, the knowledge about whom is stored in enterprise model repository of CASE tool.

Table 6. Reflections of elements of enterprise model *M1* to UCM V2.

Components of EM			**Components of UCM of Actors**		
The name of class	**Component of system M1**	**Reflection (→)**	**The name of class**	**Component of system V2**	**Formalized description**
Function	K2	φ_9	*PA Function*	K31	φ_9: K2→K31
Actor	K3	φ_{10}	*Actor*	K30	φ_{10}:K3→K30
Business Rules	K14	φ_{11}	*Extension Point*	K32	φ_{11}:K14→K32
Inf. Activity	K13	φ_{12}	*PA Activity*	K33	φ_{12}:K13→K33
Process	K1	φ_{13}	*PA Process*	K31	φ_{13}:K1→K31
Process	K1	φ_{14}	*PA_Subprocess*	K33	φ_{14}:K1→K33

Examples of the types (from I to VI) of UCM V2 are given in Figure 17.

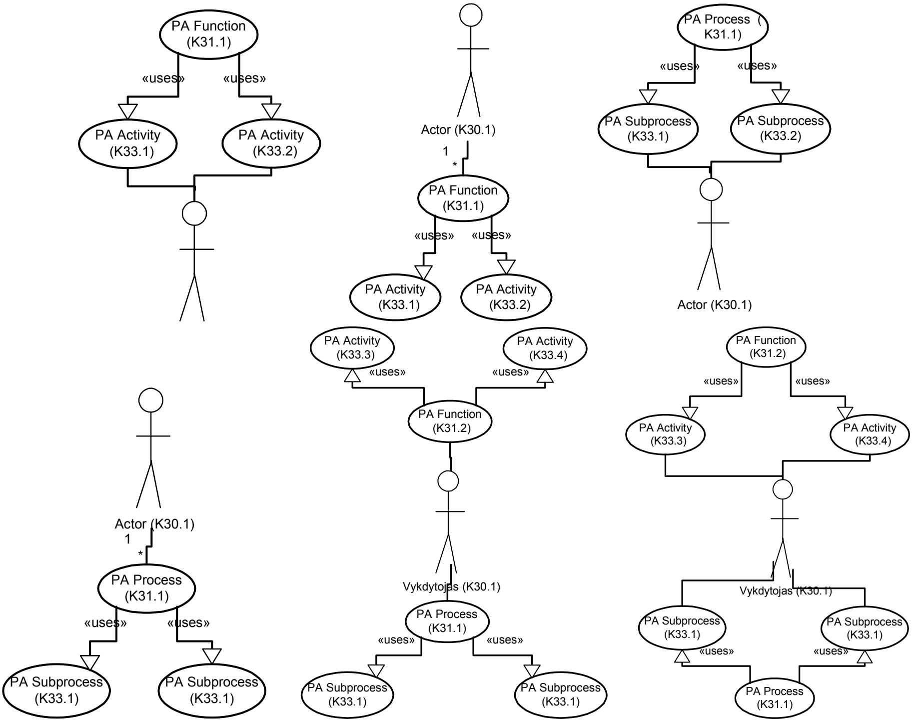

Figure 17. Examples of the types of UCM V2.

UCM of Goal Metamodel

UCM, generated for particular *Goal*, which is specified in enterprise model, is called UCM of goal. The metamodel of UCM of *Goal* consists of the classes *Actor*, the main Use Case *PA Goal,* an extra Use Case *PA Function* and *Extension point* as well as relations *Associate* and *Aggregate* UCM of *Goal* can be generated for each goal, the knowledge about which is stored in enterprise model repository.

Table 7. Reflections of elements of enterprise model M1 to UCM T2 elements.

Components of EM			**Components of UCM of Goals**		
The name of class	**Component of system M1**	**Reflection (→)**	**The name of class**	**Component of system T2**	**Formalized description**
Function	K2	φ_{15}	*PA Function*	K37	φ_{15}: K2→K37
Actor	K3	φ_{16}	*Actor*	K34	φ_{16}:K3→K34
Goal	K5	φ_{17}	*PA Goal*	K35	φ_{17}:K5→K35
Business Rule	K14	φ_{18}	*Extension Point*	K36	φ_{18}:K14→K36

Example of UCM T2 is given in Figure 18.

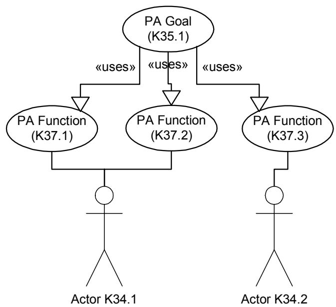

Figure 18. Example of UCM T2

Activity UCM Metamodel

UCM, generated for particular *Information Activity* , which is specified in enterprise model, is called *Activity* UCM. *Activity* UCM metamodel consists of the classes *Actor*, the main Use Case *PA Activity*, an extra Use Case *PA Function* and *Extension Point* as well as relations *Associate* and *Aggregate.* Activity UCM can be generated for each *Information Activity*, about which knowledge is stored in enterprise model repository.

Table 8. Reflection of M1 elements to UCM V3 elements.

Components of EM			**Components of UCM of Activity**		
The name of class	**Component of system M1**	**Reflection (→)**	**The name of class**	**Component of system V3**	**Formalized description**
Function	K2	φ_{19}	*PA Function*	K41	φ_{19}: K2→K41
Actor	K3	φ_{20}	*Actor*	K38	φ_{20}:K3→K38
Business Rule	K14	φ_{21}	*Extension Point*	K40	φ_{21}:K14→K40
Information Activity	K13	φ_{22}	*PA Activity*	K39	φ_{22}:K13→K39

Example of UCM V3 is given in Figure 19.

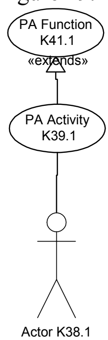

Figure 19. Example of UCM V3

Conclusions

1. After a complete analysis of enterprise modeling standards, CASE methods and tools, used in IS engineering, the following disadvantages of IS engineering user requirements acquisition, analysis and specification stage is defined:

1.1 The content of enterprise models, presently used in CASE systems, is not verified according to formalized criteria. Although enterprise models are created on the basis of various notations (such as data flow diagrams, work flow models etc.), their composition is not verified by CASE systems with respect to particular enterprise domain characteristics.

1.2 Traditionally user requirements specification process is performed by the analyst and user, who use their experience and knowledge. The result is empirical, because user requirements specification is not verified according to formalized criteria.

1.3 One of the reasons why the composition of enterprise models (created in this way) is incomplete for the improvement of CASE methods are insufficient features of both empirical Enterprise Models and user requirement specifications.

2. Composition of enterprise knowledge repository of CASE tool which is used to acquire, analyze and specify user requirements also generate Use Case models is based on enterprise meta– model. This repository is another source of knowledge (besides analyst and user), which gives formally checked information about problem domain. The work contains prototype of enterprise knowledge repository. The prototype examines enterprise knowledge repository composition, whether it is sufficient for user requirements acquisition, analysis and specification also Use Case model generation.

3. The created Enterprise Model Based CASE system repository composition method is based on the following four types modified work flow models:

1.1 Work Flow Model of Business Processes (VP_WFM), designed to specify problem domain processes, material and informational flows and actors;

1.2 Work Flow Model of Material Processes (P_WFM), designed to specify problem domain material processes, material flows and actors;

1.3 Work Flow Model of Business Functions (F_WFM), designed to specify problem domain activities, informational flows and actors;

1.4 Work Flow Model of Functional Composition (FS_WFM) designed to specify the composition of business function.

The traditional composition of work flow model had to be modified in order to develop this method. Processes and enterprise functions were distinguished as things of qualitatively different nature in terms of control theory. Enterprise process models the material processes of enterprise, while the function models the informational one.

4. The interaction between enterprise process and function is a mandatory component of enterprise model, because during the interaction the feedback loop, necessary in process control, is formed. Thus metamodel, which contains theoretically correct control process (it creates feedback loop between controlled object and controlling function) was chosen.

5. The peculiarity of enterprise model based computerized specification method of user functional requirements is that computerized problem domain knowledge is verified according to the enterprise metamodel composition. The enterprise metamodel is used as source and criterion of enterprise knowledge necessary to IS Engineering. This determines the possibility of the user requirements specification quality control performed by CASE system.

5.1 The problem domain analysis in this dissertation is performed through the distinction of qualitatively different enterprise functions and enterprise processes. While analyzing the interaction of enterprise functions and processes from the informational aspect, the opportunity to verify the functional composition of the enterprise model according to formalized criteria occurs. Here two things are distinguished: material processes in computerized enterprise domain and component parts–activities of enterprise functions, which control these processes.

5.2 The dissertation presents the business process analysis method, which identifies and eliminates logical gaps. Such gaps occur when the user gives incomplete information about the material processes and material input and output flows in the computerized problem domain.

5.3 The dissertation also presents the functional enterprise analysis method of problem domain, through which logical gaps are identified and eliminated. Such gaps occur when the user gives incomplete information about the existing informational activities and informational input and output flows in computerized problem domain. The analysis of the problem domain functional composition determines the components of each function according to the enterprise metamodel. If the user information based functional model differs from functional composition determined in the enterprise metamodel, CASE system identifies a logical gap, i.e. the lacking functional component. Thus user functional requirements specification method was developed in this work. This method ensures the user functional requirements specification, which is verified according to enterprise knowledge, acquired on the basis of formalized composition (enterprise model).

6. The modified, work flow model based prototype of CASE system enterprise knowledge repository is programmably implemented. It prove the possibility to implement created method to stages of computerized information system engineering process.

7. User requirements specification algorithms were created. On the basis of enterprise knowledge, they generate Use Case models for particular element of enterprise model: enterprise function, enterprise process, actor, goal and activity. It expands the functionality of traditional CASE tool, because interactive process of Use Case model generation is performed. This process is performed according to initial user conditions.

8. The work presents essential solutions to knowledge based computerized specification method of user functional requirements. Depicted algorithms (of the main methodological steps) and prototypes of programming realization indicate that such knowledge based method renders qualitative advantages if compared to traditional methods of IS design stage. The advantages are:

8.1 During Enterprise Model designing step, enterprise meta– model based formalized control of problem domain knowledge is performed.

8.2 Enterprise model analysis is performed from the aspect of control. The quality assurance loops of two types are introduced. The first one is used to eliminate logical gaps in P_WFM and F_WFM and apply these changes to VP_WFM. The second one is used to verify the composition of each function according Enterprise metamodel and apply these changes to P_WFM, F_WFM and VP_WFM.

8.3 In above cases, CASE enterprise knowledge repository is an active "participant" of knowledge based IS engineering process. It controls method and is an extra source of knowledge besides user and analyst. In traditional IS engineering only user and analyst stand for the source of knowledge. This is the qualitative difference in IS engineering process.

8.4 On the basis of the method (created in this work), interactive user requirements (Use Case models) generating algorithms were created. They control user requirements specification process. Such opportunity (to control generation process) is ensured by CASE tool enterprise knowledge repository.

8.5 Knowledge (stored in the repository) becomes a criteria, which controls actions of user and analyst, i.e. it reduces human factor influence in the user requirements acquisition, analysis and specification stage of IS engineering.

List of Publications on the Theme of Dissertation

Articles in ISI (Institute of Scientific Information) indexed journal

1. Gudas S., Skersys T., Lopata A. Framework for knowledge–based IS engineering, LNCS 3261, Proceedings of Third International Conference "Advances in Information Systems" (ADVIS 2004), Springer Berlin p. 512– 522. ISSN 0302–9743, ISBN 3–540–23478–0 .

Articles in the Lithuanian Journals Approved by the Department of Science and Studies

1.Lopata A., Gudas S. Žiniomis grindžiama informacijos sistemų inžinerija. // Informacijos mokslai T30, Vilnius. Vilniaus universiteto leidykla, 2004, p. 90– 98. ISSN 1392- 0561 (in Lithuanian).

2.Lopata A., Gudas S. Vartotojo poreikių modelio generavimas veiklos modelio pagrindu. // Informacijos mokslai T26, Vilnius: Vilniaus universiteto leidykla, 2003, p. 134–140. ISSN 1392- 0561 (in Lithuanian).

3.Lopata A., Gudas S. Informacijos išteklių identifikavimas, veiklos modelio pagrindu. // Informacijos mokslai, T19, Vilnius: Vilniaus universiteto leidykla, 2001, p.43–50. ISSN 1392- 0561 (in Lithuanian).

Articles in Other Lithuanian Journals

1. Lopata A., Gudas S. Žiniomis grindžiamos IS inžinerijos kilmė // Informacinės Technologijos verslui 2004: konferencijos pranešimų medžiaga. Kaunas: Technologija 2004 p. 120– 123, ISBN 9955–09–649–7. (in Lithuanian).

2. Budrevičienė S., Lopata A., Gudas S. Veiklos žinių kaupimo posistemis // Informacinės technologijos 2004: konferencijos pranešimų medžiaga, Kaunas: Technologija 2004 p.581– 584, ISBN 9955–09–588–1. (in Lithuanian).

3. Lopata A., Gudas S Modifikuotų darbų sekų modelių taikymas dalykinės srities žinioms specifikuoti // Informacinės technologijos 2003: konferencijos pranešimų medžiaga, Kaunas: Technologija, 2003 p.36–40. (in Lithuanian).

4. Lopata A., Gudas S. Veiklos funkcijų darbų sekų modelis // Informacinės technologijos verslui 2003: konferencijos pranešimų medžiaga. Kaunas 2003 p.45–48(in Lithuanian).

5. Lopata A. Veiklos modelių sudėties analizė. // Informacinės Technologijos 2002: konferencijos pranešimų medžiaga. Kaunas: Technologija 2002 p.377–381(in Lithuanian).

6. Gudas S., Skersys T., Lopata A. Domain Knowledge Integration For Information Systems Engineering // Informacinės technologijos verslui 2002: konferencijos pranešimų medžiaga. Kaunas: Technologija 2002. p.56–59.

7. Lopata A., Gudas S. Informacinių išteklių identifikavimo modelis // Informacinės technologijos verslui 2001” Kaunas Technologija 2001. p. 56–59. (in Lithuanian).

8. Lopata A., Gudas S. Veiklos informacinių išteklių kompiuterizavimo metodų analizė. // Informacinės technologijos 2001: konferencijos pranešimų medžiaga. Kaunas: Technologija 2001. p. 361– 364. (in Lithuanian).

9. Lopata A. Organizacijos informacinių išteklių specifikavimo modelis. // Informacinės technologijos, 4–osios magistrantų ir doktorantų konferencijos pranešimų medžiaga. Kaunas Technologija 1999m. p. 94–97. (in Lithuanian).

Articles in Other International Journals

1. Lopata A. Analysis of Enterprise Modeling Constructs on the Basis of International Standards. // Business Operation and its Legal Environment: processes, tendencies, results: Conference Proceedings. Riga, Latvia 2002 ISBN 9984–609–93–6. p.174–178.

2. Lopata A., Gudas S. Framework for Identification of Information Resources // New Trends of the Development of Industry: Conference Proceedings, Brno, p. 6–7 Czech Republic 2002 ISBN 80–214–2102–9 (Compact Disc).

3. Lopata A. Specification Model of Enterprise Information Resources. Transformation of Economic and Social Relations: Processes, Tendencies and Results: Conference Proceedings. Riga, Latvia 2001.

Information about the Author of the Dissertation

Audrius Lopata was born in 1976. 04. 10

1993– Finished Kaunas 7th Secondary Scool

1993– 1997: Studies at Vilnius University, Kaunas Faculty of Humanities– Bachelor of Information Management.

1997– 1999: Studies at Vilnius University, Kaunas Faculty of Humanities– Master of Information Management.

2000– 2004: PhD Studies in Informatics Engineering Kaunas University of Technology, Faculty of Informatics.

Reziumė

Mokslinės problemos esmė: Kompiuterizuotos IS inžinerijos funkcinių vartotojo reikalavimų specifikavimo metodo, grindžiamo veiklos modeliu sukūrimas. Šio metodo pagrindu sukurta CASE sistema išplečiama veiklos žinių saugykla. Tai intelektualizuoja funkcinių vartotojo reikalavimų surinkimo, analizės ir specifikavimo etapą. Pateikiami formalizuoti metodo ir inžinierinių priemonių aprašymai:

- veiklos modeliavimo būdas modifikuotų darbų sekų modelių pagrindu;
- vartotojo reikalavimų modelių (UCM) generavimo algoritmai.

Pateikiamas metodo aprašymas ir inžineriniai sprendimai, kaip veiklos modelio pagrindu sudaryta CASE įrankio veiklos žinių saugykla integruojama kompiuterizuotos IS inžinerijos vartotojo reikalavimų surinkimo, analizės ir specifikavimo etape.

Tyrimo apribojimai: Tyrimas apima šiuos IS inžinerijos etapus:

- vartotojo reikalavimų surinkimas ir analizė veiklos žinių bazės pagrindu;
- vartotojo reikalavimų specifikavimas, t.y. reikalavimų specifikacijos generavimas veiklos žinių bazės pagrindu.

Darbe panaudotas formalizuotas veiklos valdymo informacinių procesų metodas, teikiantis kriterijus veiklos modeliavimui ir vartotojo pateikiamos informacijos kontrolei.

Disertacijos tyrimo objektas. Kompiuterizuotas vartotojo funkcinių reikalavimų surinkimo procesas, grindžiamas dalykinės srities modeliavimu, ir vartotojo reikalavimų specifikavimas CASE sistemoje saugomo veiklos modelio pagrindu.

Tyrimo tikslas. Sukurti veiklos modelio sudarymo metodą ir veiklos modeliu grindžiamą funkcinių vartotojo reikalavimų surinkimo, analizės ir specifikavimo metodą.

Darbe sprendžiami tokie **uždaviniai:**

- atlikti veiklos modeliavimo standartų ir CASE sistemose naudojamų veiklos modelių analizę, siekiant išskirti esminius IS inžinerijos vartotojo reikalavimų surinkimo, analizės ir specifikavimo etapo trūkumus ir pagrįsti bei nustatyti CASE sistemos veiklos žinių saugyklos sudėtį, būtiną ir pakankamą vartotojo reikalavimams surinkti, analizuoti, specifikuoti ir vartotojo reikalavimų modeliams generuoti;
- sukurti CASE sistemos veiklos žinių saugyklos užpildymo metodą, grindžiamą darbų sekų modeliųpagrindu ;
- sukurti funkcinių vartotojo reikalavimų specifikavimo veiklos žinių saugykloje sukauptų žinių pagrindu metodą (algoritmus).
- programiškai realizuoti modifikuotais darbų sekų modeliais grindžiamą CASE sistemos veiklos žinių saugyklos užpildymo metodo prototipą;
- sukurti vartotojo reikalavimų specifikavimo algoritmus veiklos žinių pagrindu generuojančius vartotojo reikalavimų modelius (angl. Use Case model).

Tyrimo metodai: struktūriniai veiklos modeliavimo metodai, objektiškai orientuotas IS modeliavimas (UML).

Darbo mokslinis naujumas.

Darbe pateiktas veiklos modeliu grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų surinkimo, analizės ir specifikavimo metodas, kuris apima:

- kompiuterizuojamos dalykinės srities žinių surinkimo ir analizės būdą (darbų sekų modelių pagrindu), panaudojant veiklos metamodelį;
- IS vartotojo funkcinių reikalavimų specifikavimo modelių sudarymo veiklos modelio pagrindu būdą.

Disertacijoje pagrindžiamas formalizuoto veiklos metamodelio būtinumas. Darbe panaudotas formalizuotas veiklos metamodelis, kuris apriboja konkrečios dalykinės srities veiklos modelio formavimą. Metamodelio pagrindu sudarytas veiklos modelis yra formalizuotas veiklos modelis. Darbe aprašytas formalizuoto veiklos metamodelio specifikacijos pagrindu grindžiamos CASE sistemos veiklos žinių saugyklos sudarymo būdas.

Sukurtas modifikuotais darbų sekų modeliais grindžiamas CASE sistemos veiklos žinių saugyklos užpildymo metodas, užtikrinantis veiklos žinių, pakankamų vartotojo reikalavimų modeliams generuoti, surinkimą. Darbe pateiktas CASE sistemos veiklos žinių saugyklos taikymas vartotojo reikalavimų modeliams generuoti. Darbo ypatumas yra tai, kad veiklos modelis kuriamas apibrėžiant veiklos procesą ir veiklos funkciją kaip kokybiškai skirtingos prigimties veiklas. Veiklos procesas atitinka materialią (gamybos) veiklą, o veiklos funkcija atitinka informacinę (valdymo) veiklą. Veiklos proceso ir veiklos funkcijos sąveika yra būtinas veiklos modelio komponentas, nes tokiu atveju suformuojamas informacinis grįžtamasis ryšys, būtinas proceso valdymui. Todėl ir buvo pasirinktas toks veiklos metamodelis, kuriame realizuotas teoriškai korektiškas valdymo procesas, sukuriantis grįžtamąjį ryšį tarp valdomojo objekto ir valdančiosios funkcijos.

IS inžinerijos vartotojo reikalavimų surinkimo, analizės ir specifikavimo etapo loginiams trūkiams identifikuoti ir juos šalinti sukurti procesų ir funkcijų trūkio taškų šalinimo algoritmai patobulina projekto kokybės kontrolės procesą.

Darbe sukurti vartotojo reikalavimų modelių generavimo algoritmai (angl. *Use Case model)* pagal kelis vartotojo parinktus kriterijus (veiklos funkciją, veiklos procesą, vykdytoją, veiklos potikslį) generuojantys UCM variantus, kuriuos koreguoja projektuotojas.

Praktinė darbo svarba: Darbas pagrindžia veiklos žiniomis grindžiamą funkcinių vartotojo reikalavimų inžinerijos etapą, pateikia inžinerines priemones ir algoritmus vartotojo reikalavimams surinkti, analizuoti ir specifikuoti. Darbe pasiūlyta papildyti kompiuterizuotų IS kūrimo (CASE) įrankių saugyklos sudėtį veiklos žinių saugykla. Tokios CASE sistemos, papildytos veiklos žinių saugykla, privalumas tas, kad joje saugomos kompiuterizuojamos dalykinės srities žinios yra papildomas šaltinis sukurti projektinius IS modelius. Tai išplečia CASE sistemos funkcionalumą, t.y projektinių modelių generavimo ir testavimo galimybes.

Darbe pasiūlytos naujos inžinerinės priemonės dalykinės srities žinioms į CASE įrankio veiklos žinių saugyklą surinkti– modifikuoti darbų sekų modeliai:

- veiklos procesų darbų sekų modelis;
- procesų darbų sekų modelis;
- funkcijų darbų sekų modelis;
- funkcijos sudėties darbų sekų modelis.

Rezultatų aprobavimas. . Mokslinio darbo rezultatai paskelbti 16 mokslinių publikacijų, iš kurių 1 publikacija atspausdinta leidinyje, įrašytame į Mokslinės informacijos instituto (ISI) leidinių sąrašą, 3 publikacijos atspausdintos leidiniuose įrašytuose į Mokslo ir studijų departamento patvirtintą leidinių sąrašą, 3 publikacijos kituose recenzuojamuose tarptautiniuose ir užsienio leidiniuose ir 9 publikacijos respublikinių konferencijų pranešimų medžiagose.

Darbo struktūra ir apimtis. Disertaciją sudaro įvadas, 3 dalys, išvados, literatūros sąrašas, publikacijų sąrašas ir 19 priedų. Pagrindinė darbo medžiaga aprašyta 174 puslapiuose, įskaitant 57 lenteles, 108 paveikslų. Panaudotos literatūros sąrašą sudaro 91 šaltiniai. Darbo struktūrą nusako tyrimo objektas, tikslas bei iškelti uždaviniai. Autoriaus publikacijų sąrašas pateiktas disertacijos pabaigoje.

UDK 004.415.2(043)
SL 344. 2004 -12- 24 1 leidyb.apsk.l. Tiražas 70 egz. Užsakymas 474.
Išleido leidykla „Technologija“, K. Donelaičio g. 73, 44029 Kaunas.
Spausdino leidyklos “Technologija” spaustuvė, Studentų g. 54, 51424 Kaunas.